

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
Department "Institut für Informatik"
Lehr- und Forschungseinheit Medieninformatik
Prof. Dr. Heinrich Hußmann

Masterarbeit

On-Site 3D Augmented Reality Tourist Guide

Simon Pfaab
S.Pfaab@campus.lmu.de

Bearbeitungszeitraum: 01.06.2015 bis 01.02.2016
Betreuer: Dr.-Ing. Wolfgang Höhl
Externer Betreuer: Matthias Greiner
Verantw. Hochschullehrer: Prof. Dr. Heinrich Hußmann

Zusammenfassung

Im Rahmen dieser Arbeit wird eine Augmented Reality-Anwendung für Smartphones und Tablets erstellt, mit deren Hilfe man ein historisches, bereits zerstörtes Gebäude als 3D-Modell im städtischen Raum München erlebt. Aufbauend auf bereits bestehenden Augmented Reality-Applikationen wie beispielsweise *Zeitfenster*, *Chronovisor* oder *TimeTraveler* soll die historische Rekonstruktion verbessert und erweitert werden.

Voranehend werden Herausforderungen von Augmented Reality-Applikationen und Tourist Guides im Allgemeinen betrachtet und analysiert, um anschließend eine möglichst optimale Lösung für das gegebene Szenario zu erarbeiten. Anhand einer markerlosen Bilderkennung soll eine exakte dreidimensionale Umgebungserkennung erfolgen, die das genaue Einpassen des virtuellen Gebäudemodells in den städtischen Raum erlaubt. Hierbei liegt der Fokus neben der Auswahl des richtigen Trackingverfahrens auch auf der Positionierung des Modells mit Berücksichtigung der wechselnden Lichtverhältnisse zu unterschiedlichen Tages- und Jahreszeiten und der Erfassung von Verdeckung und Transparenz.

Für die Umsetzung der Ziele werden in Kooperation mit der *Metaio GmbH* die dort entstandenen Anwendungen *Metaio Toolbox*, *Metaio Creator* und *Junaio AR Browser* verwendet. Schon früh werden Nutzer in die Entwicklung integriert und per Anwendungsstudien soll gewährleistet werden, dass als Ergebnis dieser Arbeit ein interaktiver und vor allem benutzerfreundlicher On-Site 3D Augmented Reality Tourist Guide entsteht.

Abstract

In this thesis, an augmented reality application for smartphones and tablets will be developed. With this app, the user can experience a historic and already destroyed building in the city of Munich in form of a 3D-model. Based on already existing augmented reality applications like *Zeitfenster*, *Chronovisor* or *TimeTraveler*, the methods of historic reconstruction shall be improved and extended.

Prior to the development, challenges of augmented reality applications and also tourist guides in general will be verified and analysed, so that afterwards the best solution for the given scenario can be found. Based on markerless image tracking a three-dimensional environment recognition shall take place, which allows the exact positioning of the virtual model of the building in the city environment. Doing this, besides on the selection of the right tracking method, the positioning while considering the changing lighting conditions at different times of day and season, and occlusion and opacity will be the focus of this work.

These goals will be realised in cooperation with the *Metaio GmbH* using their applications *Metaio Toolbox*, *Metaio Creator* and *Junaio AR Browser*. Early in the development stage, users will be involved and by making usability evaluations an interactive and user-friendly on-site 3D augmented reality tourist guide will be realised.

Aufgabenstellung

Diese Arbeit beschäftigt sich mit der Funktionsweise von markerlosen Trackingsystemen in Augmented Reality Anwendungen für die historische Rekonstruktion von Gebäuden. Die generelle Funktionsweise von markerlosen Trackingsystemen wird erfasst, untersucht und methodisch verbessert.

Zur Zeit gibt es folgende Problemkreise von Augmented Reality bei Anwendungen im Aussenraum: (A) Die performante, markerlose und exakte dreidimensionale Umgebungserkennung (Tracking) durch virtuelle Kantenmodelle (B) die Berücksichtigung wechselnder Lichtverhältnisse zu unterschiedlichen Tages- und Jahreszeiten im Aussenraum und (C) das genaue dreidimensionale Einpassen virtueller Geometrie in eine vorhandene städtebauliche Situation unter Berücksichtigung von Verdeckung (Occlusion) und Transparenz.

Ähnliche Ansätze verfolgen die Anwendungen Zeitfenster (Burkert et al.), Chronovisor (Gontz et al.), TimeTraveller (Metaio GmbH.) und die Arbeit On-Site Augmented Collaborative Architecture Visualization (Schattel et al.). Es handelt sich dabei um vorhandene Augmented Reality Lösungen für die historische Rekonstruktion von Gebäuden.

Als Basis für die Anwendung wird der Junaio AR Browser, Metaio Toolbox und 3D Maps gewählt. Die Anwendung soll es auch dem Laien ermöglichen, virtuelle oder historische Gebäudemodelle im städtischen Raum über AR-Technologie zu erleben.

Ergebnis dieser Arbeit ist ein Prototyp eines interaktiven, benutzerfreundlichen On-Site 3D Augmented Reality Tourist Guides für ein vorgegebenes historisches Gebäude in München, lauffähig auf mobilen Endgeräten auf den Plattformen iOS und Android.

Wichtiger Bestandteil der Arbeit ist die Usability Evaluation als iterative, benutzerzentrierte Entwicklung mit Benutzerbeteiligung schon in frühen Phasen der Anwendungsentwicklung.

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, 01. Februar 2016

.....

Sperrvermerk

Die Technische Universität (TU) Darmstadt, Fachgebiet Digitales Gestalten, stellt Herrn Simon Pfaab für die Anfertigung dieser Masterarbeit am Lehrstuhl für Medieninformatik der Ludwig Maximilians-Universität (LMU) München die digitalen Daten des 3D-Modells der Hauptsynagoge München kostenfrei zur Verfügung. Eine weitere kommerzielle Verwertung dieses 3D-Modells ist nicht vorgesehen. Bei jedweder Veröffentlichung ist eine Genehmigung des Fachgebiets Digitales Gestalten, durch Dr.-Ing. Marc Grellert zu erfragen. Das Logo der TU Darmstadt wird integriert.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Nutzen eines On-Site 3D Augmented Reality Tourist Guides	2
1.2	Überblick über die Kapitel	2
2	Aktueller Forschungsstand	5
2.1	Augmented Reality-Systeme	5
2.1.1	Augmented Reality	5
2.1.2	Tracking	6
2.1.3	Sensorbasiertes Tracking	7
2.1.4	Bildbasiertes Tracking	8
2.1.5	Zusammenfassung	9
2.2	Tourist Guides	9
2.2.1	Anforderungen	10
2.2.2	Anwendungsgebiete	10
2.2.3	Nutzergruppe	10
2.2.4	Zusammenfassung	11
2.3	Augmented Reality Tourist Guides	11
2.3.1	Chronovisor	11
2.3.2	Landauer Walk	12
2.3.3	Zeitfenster	13
2.3.4	TimeTraveler	14
2.3.5	Zusammenfassung	15
3	Konzeption des Prototypen	17
3.1	Zielsetzung	17
3.2	Anforderungen	18
3.3	Entwicklungsumgebung	18
3.3.1	Native Entwicklung	19
3.3.2	AREL	19
3.3.3	Metaio Creator	20
3.4	Anwendungen	21
3.4.1	<i>Junaio Browser</i>	21
3.4.2	Beispiel	22
3.5	Zusammenfassung	23
4	Entwicklung des Prototypen	25
4.1	Benutzeroberfläche	25
4.1.1	Evaluierung	27
4.1.2	Ergebnis der Evaluierung	27
4.2	Modellanpassung	29
4.3	Markerloses Tracking	30
4.4	Einpassen der Geometrie	32
4.4.1	Berücksichtigung der Lichtverhältnisse	33
4.4.2	Verdeckung und Transparenz	33
4.5	Offline-Tour	34
4.6	Navigation zum Ziel	35
4.7	Hintergrundinformationen	36

5	Ergebnis der Entwicklung	37
5.1	Finaler Prototyp	37
5.2	Evaluierung finaler Prototyp	38
5.3	Ergebnis der Evaluierung	39
6	Fazit	43
6.1	Zusammenfassung	43
6.2	Verbesserungsmöglichkeiten	43
6.3	Anwendungen und Chancen	44

Abbildungsverzeichnis

1.1	Die Datenbrillen <i>Google Glass</i> (links) und <i>Microsoft Hololens</i> . [11, 17]	1
2.1	Realitäts-Virtualitäts-Kontinuum nach Milgram and Colquhoun [28].	5
2.2	Die Sensoren eines Smartphones, die beim sensorbasierten Tracking zum Einsatz kommen und ihre Funktionen. [36]	7
2.3	Bildbasiertes Tracking mit Hilfe von Kantenerkennung. [29]	9
2.4	Beispielszene aus der Applikation <i>Chronovisor</i> von <i>Colorbitor</i> [8].	12
2.5	Beispielszene aus der Applikation <i>LandauerWalk</i> [35].	13
2.6	Beispielszene aus der Applikation <i>TimeTraveler</i> von <i>Metaio</i> [1].	14
3.1	Städtische Umgebung rund um den Oberpollinger in München.	17
3.2	Überblick über die Funktionsweise von AREL [29].	19
3.3	Auswahl der Trackingverfahren im <i>Metaio Creator</i> [29].	20
3.4	Überblick über die Funktionsweise der Junaio-Channels [29].	21
3.5	Beispielapplikation im <i>Junaio Browser</i> von <i>Metaio</i> [29].	22
4.1	Menüstruktur der Anwendung <i>LandauerWalk</i>	25
4.2	Unterschiedliche Versionen der Benutzeroberfläche.	26
4.3	Ergebnisse der ersten Evaluierung bezüglich Benutzeroberfläche.	27
4.4	Ergebnisse der ersten Evaluierung bezüglich Benutzeroberfläche.	28
4.5	Unterschiedliche Grünfarbtöne für die Buttons der Anwendung.	29
4.6	Modell der <i>Alten Hauptsynagoge München</i> der <i>TU Darmstadt</i> . [24]	30
4.7	Fehlerhafte und nutzbare 3D-Map des Oberpollingers.	31
4.8	Maßstabsgetreues 3D CAD Modell der Fassade des Oberpollingers in München.	32
4.9	Zwei unterschiedlich texturierte Gebäudemodelle, jeweils eins für tagsüber (links) und nachts (rechts).	33
4.10	Einbettung eines Verdeckungsmodells, damit die Hauptsynagoge hinter dem Künstlerhaus positioniert erscheint.	34
4.11	Die Offline-Tour des Prototypen <i>HistoryTraveller</i>	35
5.1	Finales Hauptmenü der Anwendung <i>HistoryTraveller</i> mit anschließendem Menü.	37
5.2	Live-Tour im finalen Prototypen. Links in der Naherfahrung und rechts in der Fernerfahrung.	38
5.3	Schnelligkeit bei der Auffindung der Funktionen des finalen Prototyps.	40
5.4	Bewertung der Trackingverfahren bei der Nah- und Fernerfahrung des finalen Prototyps des <i>HistoryTravellers</i>	41

1 Einleitung

Aktuelle Entwicklungen zeigen, dass im Bereich der Umsetzung von Augmented Reality-Applikationen unzählige Möglichkeiten zur Entfaltung vorhanden sind. Die Tatsache, dass dem Nutzer Objekte und Zusatzinformationen bei Augmented Reality-Anwendungen direkt in das Sichtfeld projiziert und zur Veranschaulichung gezeigt werden können, machen sich viele Entwickler für kreative Ideen zunutze. Aufgrund dieser Möglichkeiten entstand die Idee, eine Applikation (App) für die Tourismusbranche zu entwickeln. Diese sollte dem Nutzer eine Erfahrung von historischen Informationen in Form von multimedialen Inhalten wie Text, Bildern und Videos in Augmented Reality (AR) bieten. Die anfänglichen Ziele waren es, einen Stadtführer mit diesen Funktionen zur Verfügung zu stellen und mit dessen Hilfe man die Vergangenheit heutzutage wieder neu erleben kann.

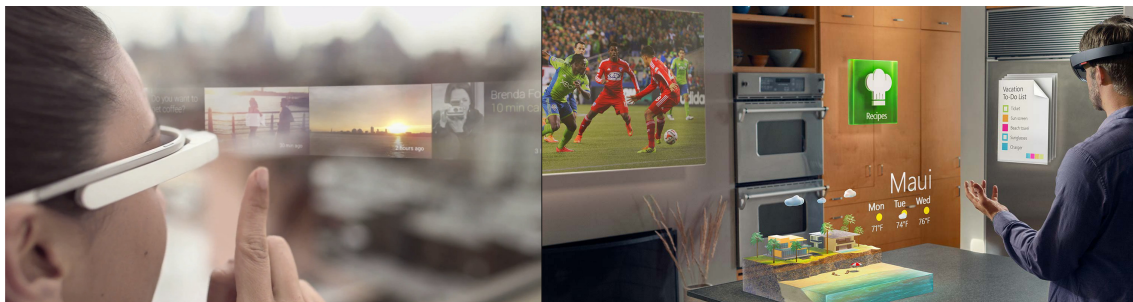


Abbildung 1.1: Die Datenbrillen *Google Glass* (links) und *Microsoft Hololens*. [11, 17]

Dass diese Ziele keine Fiktion sind, sondern Gegenwartsbezug besitzen, zeigt sich daran, dass beispielsweise das US-amerikanische Unternehmen *Google Inc.*, mittlerweile *Alphabet Inc.*, am 4. April 2012 seine neueste Entwicklung aus der sogenannten *Google-X-Sparte* vorstellte, die *Google Glass* (siehe Abbildung 1.1, links). Auch die *Microsoft Corporation* präsentierte Anfang 2015 der Öffentlichkeit mit der *Microsoft Hololens* eine Augmented Reality-Brille (siehe Abbildung 1.1, rechts). Wie in Abbildung 1.1 zu sehen, handelt es sich bei beiden Produkten jeweils um Datenbrillen, die dem Träger die Funktionen eines Smartphones oder Tablets bieten sollen und dabei die Informationen direkt in das Sichtfeld des Nutzers projizieren. Das Display der *Google Glass* beispielsweise besteht aus einem Prisma und befindet sich oberhalb des normalen Sichtfeldes, wohingegen man bei der *Hololens* ein Headset mit zwei transparenten Gläsern trägt und als Display nutzt. Somit entsteht die Möglichkeit, die aufgenommene Realität mit Hilfe der Displays zu erweitern und für Funktionen diverser Apps zu nutzen. Beispielsweise kann man einen realen Text in einer Fremdsprache durch die Brille lesen und von ihr übersetzen lassen. Zur verständnisvolleren Rezeption zeigt die Brille den übersetzten Text anschließend anstelle des unverständlichen Textes an, beziehungsweise überlagert diesen. Dadurch werden dem Nutzer anstatt des unverständlichen Textes lesbare Wörter angezeigt. [11, 17]

In den unterschiedlichsten Bereichen wie beispielsweise im medizinischen Sektor [4], im Produktionssektor zur Hilfe bei der Herstellung oder Reparatur von Produkten [6, 33] oder auch im Unterhaltungssektor in Form von Spielen wird die Augmented Reality-Technologie heutzutage auf innovative Art und Weise genutzt. Gut zu sehen sind die Möglichkeiten im ersten Vorstellungsvideo der Hololens [11]. Zusätzlich beschleunigen Erfindungen wie die Datenbrillen oder auch neue Smartphones mit besserer Rechenleistung die Verbreitung von Augmented Reality-Apps. Gleichzeitig nutzen weitere Branchen die Verbreitung dieser Technologie, um selbst nützliche Funktionen und Anwendungen zur Verfügung zu stellen.

Erste Versuche in der Tourismusbranche entstanden mit Applikationen wie etwa dem *Zeitfenster*, dem *Chronovisor* oder dem *TimeTraveler*. Alle drei Apps stellen Tourist Guides dar, die Augmented Reality für die Darstellung von virtuellen Informationen in der realen Umgebung nut-

zen. Diese Idee wurde aufgegriffen und durch die Verwendung von 3D-Modellen im 3D-Raum soll dem Nutzer ein möglichst realitätsnahes und realistisches Erlebnis geboten werden. Bei der Suche nach einem geeigneten Szenario boten sich mehrere im 2. Weltkrieg zerstörte Gebäude in München an, die mit Hilfe der Augmentierung wieder erlebbar gemacht werden könnten. Im Rahmen dieser Arbeit wird die *Alte Münchner Hauptsynagoge* verwendet, die bis 1938 in München zwischen dem Karlsplatz und dem Lenbachplatz ihren Platz hatte. Die Synagoge zeichnete sich durch die interessante Bauweise im neoromanischen Stil aus und galt lange Zeit als eine der schönsten und bedeutendsten Sakralbauwerke im deutschsprachigen Raum [24]. Aufgrund der besonderen Erscheinung und dem geschichtlichen Hintergrund ist das ehemalige Gebetshaus eine für Touristen potentiell interessante Sehenswürdigkeit, die wegen der Zerstörung durch die Nationalsozialisten nicht mehr besteht. Mit Hilfe der App und der Smartphone- beziehungsweise Tablet-Kamera soll es am Ende der Arbeit also möglich sein, das Gebäude vor Ort als 3D-Modell anstelle des sich dort jetzt befindlichen Oberpollinger-Gebäudes zu sehen.

1.1 Nutzen eines On-Site 3D Augmented Reality Tourist Guides

Aus den obigen Anführungen ergibt sich, dass als Ergebnis dieser Arbeit der Prototyp eines On-Site 3D Augmented Reality Tourist Guide entstehen soll. Dieser soll Augmented Reality-Funktionen mit den Tätigkeiten eines üblichen Tourist Guides vereinen. Ergo soll der Nutzer der App mit Hilfe von Navigationssystemen an die wichtigen Orte geführt werden. Neben einer digitalen Karte hilft somit auch die Navigationsfunktion für die schnelle Findung der schnellsten Route und die sichere Navigation an den Zielort. Als Routenplaner oder Reiseleiter dient der Tourist Guide somit als Stadtführer der eigenen Stadttour.

Vor Ort soll der Nutzer mit Hilfe der Augmented Reality-Funktion der App eine digitale Rekonstruktion erhalten, wodurch eine Erfahrung des historischen Gebäudemodells im heutigen städtischen Raum dem Nutzer zeigen soll, wie das Gebäude vor Ort zu seiner Zeit aussah. Durch die Einbettung eines 3D-Modells in den städtischen Raum hebt sich die Anwendung von herkömmlichen Tourist Guides ab. Die Vor-Ort-Erfahrung macht das Erlebnis authentischer und soll dem Nutzer die Illusion bieten, durch die Immersion in die Vergangenheit und direkt vor das damalige Gebäude versetzt zu werden. Neben der visuellen Erfahrung des bereits zerstörten Gebäudes soll der Tourist Guide außerdem als Informationsquelle genutzt werden. Mit Hilfe von Texteinblendungen und Bildern erfährt der Nutzer alle grundlegenden Hintergrundinformationen über die ehemalige Synagoge. Interessierte Nutzer sollen Hintergründe wie die Baugeschichte oder Besonderheiten der Architektur erleben. Somit dient der On-Site 3D Augmented Reality Tourist Guide wie auch der klassische Reiseführer zur Wissenserweiterung über die Sehenswürdigkeit und zusätzlich zur Vor-Ort-Erfahrung des historischen Gebäudes an seinem ehemaligen Standort.

1.2 Überblick über die Kapitel

Diese Arbeit stellt die Entwicklungsschritte eines prototypischen On-Site 3D Augmented Reality Tourist Guides dar und beginnt mit der Darlegung des aktuellen Forschungsstandes. Als Erstes wird der Begriff Augmented Reality erläutert, wie die dafür nötige Umgebungserkennung (Tracking) funktioniert und welche Vor- und Nachteile die unterschiedlichen Verfahren besitzen. Anschließend befasst sich die Arbeit mit den Anforderungen und Anwendungsgebieten eines Tourist Guides um darzulegen, auf was bei einer Tourist Guide-Anwendung geachtet werden muss. Weiterhin wird anhand bestehender Anwendungen und Arbeiten der derzeitige Forschungsstand bei Augmented Reality Tourist Guides gezeigt und gleichzeitig dargestellt, welche Arbeiten als Inspiration für den Prototyp dieser Arbeit dienen. Auf Problemfelder der bestehenden Anwendungen wird dabei eingegangen und zugleich versucht, diese im entstehenden Prototyp zu verhindern, beziehungsweise entsprechende Lösungen dafür zu finden.

Kapitel 3 stellt die Ausarbeitung des Konzepts mit den definierten Zielen und den daraus hergeleiteten Anforderungen des Ergebnisses dar. Die von *Metaio* zur Verfügung gestellten Produkte werden in diesem Schritt vorgestellt, um daraus ein grundsätzliches Verständnis für die Möglichkeiten zur Umsetzung des Prototyps zu erhalten. Zusätzlich wird auf die daraus resultierende Anwendungsarchitektur eingegangen, die die Basis für die Applikation bilden wird.

In Kapitel 4 findet die Dokumentation der einzelnen Produktions- und Implementierungsschritte statt. Hier werden auftretende Probleme angeführt und welche Lösungen dafür gefunden wurden. Da der Prototyp benutzerzentriert entwickelt wird, werden in Kapitel 4.1.1 und 5.2 die Anwendungsstudien ausgewertet, die schon während der Entwicklung durchgeführt wurden. Die zwei Stadien schließen zum einen die Entwicklung der Benutzeroberfläche und zum anderen die Fertigstellung des Prototyps ab. Diese Stadien des Prototyps werden hier evaluiert und zeigen dem Leser, welche Entscheidungsfaktoren zum letztendlichen Ergebnis der Arbeit geführt haben.

Anschließend wird das Ergebnis der Produktion in Kapitel 5 präsentiert. Alle Funktionen der finalen Anwendung werden hier nochmals im Detail erläutert. Die bereits erwähnte Evaluierung wird ausgewertet und die Ergebnisse präsentiert.

Als Abschluss der Arbeit werden in Kapitel 6 nach einer Zusammenfassung Verbesserungsmöglichkeiten und weitere mögliche Schritte erörtert. Dem Leser soll gezeigt werden, wohin die Weiterentwicklung eines On-Site 3D Augmented Reality Tourist Guides noch führen kann und wo sich Anschlusspunkte für weitere Arbeiten und Untersuchungen bieten.

2 Aktueller Forschungsstand

Aktuelle Entwicklungen und bereits bestehende Anwendungen bilden die Basis dieser Arbeit. Der aktuelle Forschungsstand soll die Grundlagen zu verstehen geben und außerdem auf vorhandene Probleme eingehen, für die während der Entwicklung des Prototypen Lösungen gefunden werden sollen. Als Erstes wird der Begriff Augmented Reality geklärt und die dazugehörigen Verfahren der Umgebungserkennung aufgezeigt, mit ihren Vor- und Nachteilen. Anschließend werden die Anforderungen und Anwendungsgebiete von Tourist Guides erörtert. Zuletzt werden Beispiele für Tourist Guides mit Augmented Reality-Funktion beschrieben und die Funktionen und Problemfelder analysiert.

2.1 Augmented Reality-Systeme

Augmented Reality-Systeme sind computerbasierte Programme oder Applikationen, die das reale Bild mit digitalen Informationen in Form von Bild, Video oder 3D-Objekten überlagern. Somit muss man sich bei der Entwicklung eines Augmented Reality-Szenarios mit zwei Problemfeldern auseinandersetzen. Einerseits mit der genauen Überlappung von realem und virtuellem Koordinatensystem zur Platzierung von digitalen Informationen. Das heißt, dass zur optimalen Positionierung eines digitalen Objekts in die Echtwelt ein gemeinsamer Nenner für die Positionsbeschreibung gefunden werden muss. Andererseits mit dem Wahrnehmen der Umgebung mit Hilfe von Sensoren oder einer Kamera und den daraus resultierenden Bewegungen. Zur Adressierung dieser Probleme gibt es unterschiedliche Ansätze, die im Folgenden erörtert werden. [23]

2.1.1 Augmented Reality

Die ersten Gehversuche im Bereich der Augmented Reality-Technologie wurden in den 1960er Jahren gemacht, als Sutherland [37] ein Head Mounted Display verwendete, um 3D Graphiken zu projizieren [37]. Allerdings wurde erst 1997 durch Azuma [2] eine Klassifizierung des Feldes mit Beschreibung der Probleme und einer Zusammenfassung der bisherigen Entwicklungen veröffentlicht. Ab den späten 1990er-Jahren erschienen viele neue Arbeiten mit dem Thema Augmented Reality und diese wurden auf diversen Workshops wie beispielsweise dem *International Workshop and Symposium on Augmented Reality (ISMAR)* oder dem *International Symposium on Mixed Reality (ISMR)* vorgestellt und diskutiert. [3]

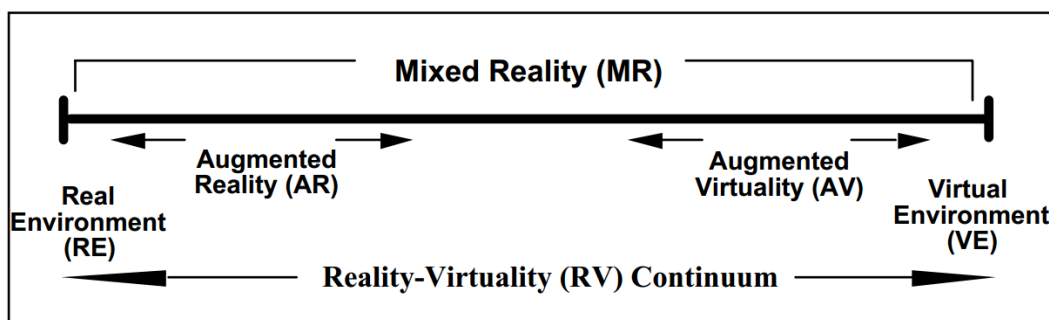


Abbildung 2.1: Realitäts-Virtualitäts-Kontinuum nach Milgram and Colquhoun [28].

Der Begriff Augmented Reality bedeutet übersetzt erweiterte Realität und beschreibt zutreffend gut die Bedeutung. Bei Augmented Reality wird eine Echtweltszene erweitert beziehungsweise augmentiert, indem virtuelle Informationen angezeigt werden. Betrachtet man Augmented Reality im Realitäts-Virtualitäts-Kontinuum (Abbildung 2.1) von Milgram und Colquhoun [28], dann sieht man, dass bei AR ein reales Umfeld bestehend aus realen Objekten um virtuelle Objekte wie Bilder oder Videos erweitert wird. Es ist somit die Vorstufe zu einer realen Umgebung,

die ausschließlich aus realen Objekten besteht, wie beispielsweise einer Fotoaufnahme. Das Pendant zur AR ist die Augmented Virtuality (AV) beziehungsweise das virtuelle Umfeld zum realen Umfeld. Anders als in der AR wird in der AV ein virtuelles Bild mit realen Daten erweitert. Wenn bei einer Computerspielfigur zum Beispiel ein echtes Foto einer echten Person als Gesicht dient, dann spricht man von AV. Ein virtuelles Umfeld hingegen besteht rein aus virtuellen Objekten. Als Beispiel kann man hier eine Computersimulation anführen. [5, 27]

Die Augmentierung bietet diverse Vor- und Nachteile. Milgram [27] arbeitete einige Problemfelder heraus, mit denen man sich bei der Implementierung von Augmentierungen beschäftigen muss. Dazu gehören Helligkeits-, Kontrast- und Auflösungsunterschiede sowie Verdeckungskonflikte. Eine ungleiche oder falsche Helligkeit lässt augmentierte Objekte dem Nutzer näher erscheinen oder in der Szene fremd wirken. Kontrastunterschiede machen das Objekt hingegen kleiner als gewünscht. Auflösungsunterschiede lassen im ungünstigen Fall das projizierte Objekt unklar erscheinen. Zusätzlich muss darauf geachtet werden, dass die virtuellen Objekte keine relevanten Objekte verdecken oder umgekehrt. [27]

Andererseits sprechen viele Vorteile für eine augmentierte Realität. So sind die Systeme meist sehr intuitiv (bedienbar), da die virtuellen Objekte im besten Fall wie reale Objekte agieren und somit die Interaktionsmöglichkeiten die gleichen sind. Dadurch ist es für den Nutzer einfacher und intuitiver, mit AR zu interagieren. [30]

Zusätzlich bieten sich dem Entwickler viele Möglichkeiten, Informationen anhand von Texten, Bildern oder Objekten direkt in der Szene darzustellen. Bei einer einfachen Navigationsapplikation beispielsweise zeigt ein digitales Objekt auf dem Kamerabild des Nutzers, in welcher Richtung sich der gewünschte Ort befindet und wie weit dieser noch entfernt ist.

2.1.2 Tracking

Um dem Nutzer eine sinnvolle AR-Erfahrung bieten zu können, muss das Gerät in den meisten Fällen wissen, wo genau sich der Nutzer innerhalb einer realen Szene, im Verhältnis zu dem im Kamerabild sichtbaren Teil, befindet. Somit kommt vor der eigentlichen Erfahrung zuerst die Umgebungserkennung, das sogenannte Tracking. Neben dem Display und der darauf stattfindenden Anzeige ist somit das Tracking die fundamentalste Technologie, um AR zu ermöglichen [39]. Anders als in einer rein virtuellen Umgebung werden bei Augmented Reality kleinste Abweichungen schon als Fehler registriert. Azuma [2] zeigt das am Beispiel einer virtuellen Hand, die sich an der Position der eigenen Hand befinden soll. Hat man eine rein virtuelle Umgebung, fällt einem eine falsche Positionierung um ein paar Zentimeter nicht auf. Anders hingegen bei einer erweiterten Realität, in der die Hand im realen Umfeld sichtbar ist und der Nutzer kleinste Abweichungen sofort erkennt. [2]

Das Tracking teilt also dem System die Position und die Orientierung eines oder mehrerer Nutzer mit, um somit Bewegung und Interaktion zu ermöglichen und die genaue, sinnvolle und vor allem (perspektivisch) korrekte Positionierung virtueller Objekte im realen Raum durchzuführen [32]. Nach der Initialisierung, zur Erkennung der Szene, muss somit auch für jedes weitere Frame (Bild) der Kamera das korrekte Tracking zur Nachverfolgung der Szene stattfinden. Dadurch wird gewährleistet, dass die Augmentierung auch fortwährend richtig angezeigt wird. Für das Tracking werden im Folgenden mehrere Verfahrensweisen vorgestellt, die jeweils unterschiedliche Vor- und Nachteile bieten. Nach Zhou et al. [39] kristallisierten sich bei den Internationalen *Symposien zur Mixed and Augmented Reality (ISMAR)* zwei grundlegende Arten des Tracking heraus, das sensorbasierte und das bildbasierte Tracking. Zusätzlich kann man noch das hybride Tracking und das Tracking mit externen Inputs anführen.

2.1.3 Sensorbasiertes Tracking

Beim sensorbasierten Tracking werden die Sensoren des Geräts, meist in Kombination miteinander, verwendet um die reale Szene zu erkennen. Bei einem Smartphone zählen unter anderem der GPS-Empfänger, der Beschleunigungssensor, der Magnetometer und das Gyroskop zu den wichtigsten Sensoren für das Tracking (siehe Abbildung 2.2).

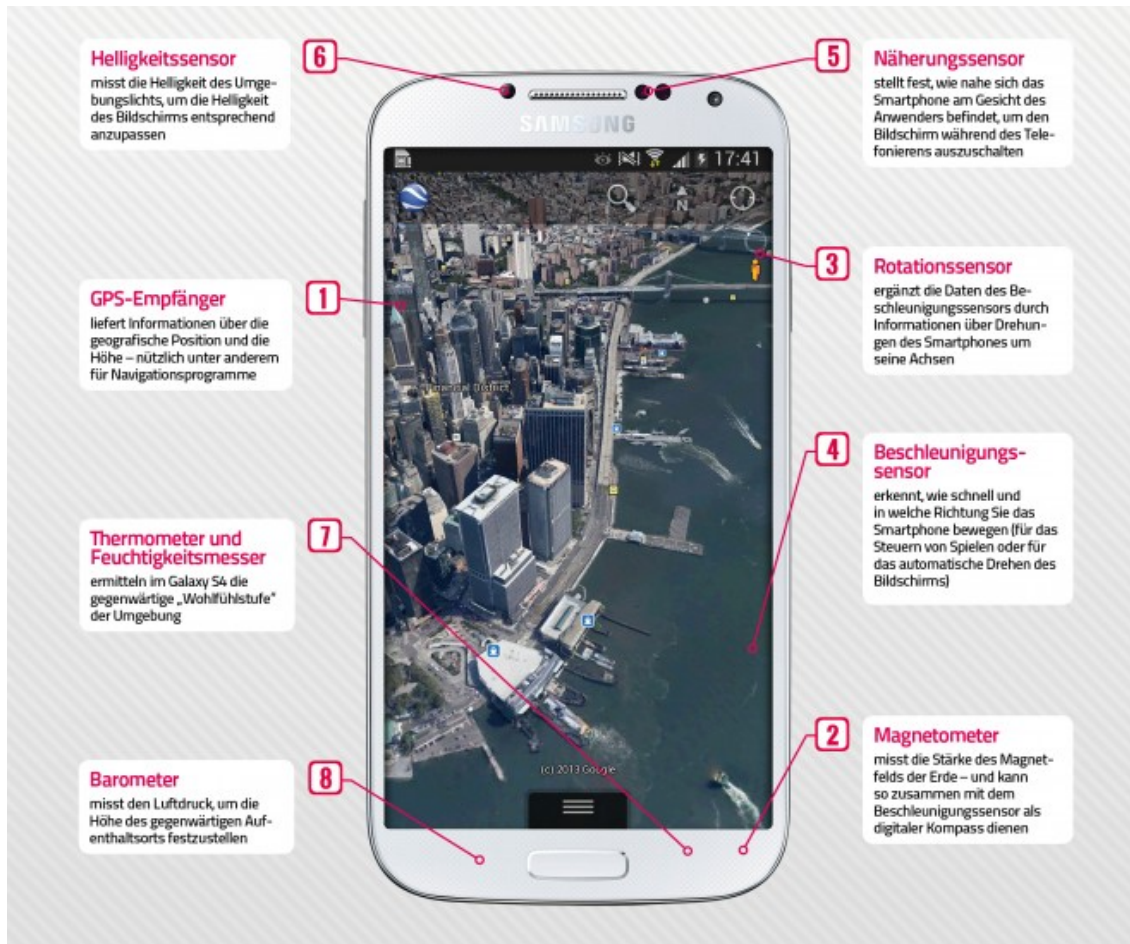


Abbildung 2.2: Die Sensoren eines Smartphones, die beim sensorbasierten Tracking zum Einsatz kommen und ihre Funktionen. [36]

Das Global Positioning System (GPS) nutzt die Signale von mindestens 24 Satelliten, die um die Erde kreisen, um die Position des Nutzers beziehungsweise des Geräts zu ermitteln. Je nach Smartphone wird teilweise auch das russische Satellitennavigationssystem *GLONASS* genutzt, um bei schlechtem Empfang die Genauigkeit der Positionserkennung zu verbessern. Zusätzlich nutzen Smartphones auch den Mobilfunksender und die WLAN-Access-Points, um den Aufenthaltsort festzustellen. Für das Tracking bedeutet dies, dass durch den GPS-Sensor bestimmt werden kann, wo sich der Nutzer befindet. Nachteil allerdings ist, dass die Positionsbestimmung relativ ungenau ist und keine zentimeter- beziehungsweise metergenaue Positionsbestimmung möglich ist. Somit eignet sich der Sensor nicht bei kleinen (metergenauen) Darstellungen, die exakt auf den Punkt sein müssen, sowie für Indoor-Anwendungen, da hier kein Signal der Satelliten empfangen werden kann. Zusätzlich variiert die Positionsermittlung in der Stadt und auf dem Land durch unterschiedliche äußere Einflüsse wie etwa Wolkenkratzer. Die großen Gebäude oder auch andere Objekte können die Send- und Empfangssignale des GPS stören. Zuletzt kennt die Anwendung durch das GPS nur die Position, allerdings nicht die Ausrichtung oder Neigung des Geräts. Hierfür sind weitere Sensoren notwendig. [32, 36]

Der Beschleunigungssensor oder auch Accelerometer erkennt die Beschleunigung, die auf das Gerät wirkt und misst somit wie schnell das Gerät in welche Richtung bewegt wird. Da auch die Schwerkraft als Beschleunigungskraft auf das Smartphone wirkt, kann das Gerät zwischen waagrechter und senkrechter Haltung unterscheiden. Somit wird entsprechend die Bildschirmausrichtung der Anzeige verändert. Zusätzlich verbessert der Accelerometer die Genauigkeit der Positionsbestimmung von Navigationsprogrammen. Sollten nach einer Positionsbestimmung lange keine GPS-Signale mehr empfangen worden sein, wird die Position anhand der Geschwindigkeit durch Berechnung mit Hilfe des Beschleunigungssensors berechnet. [36]

Die Berechnung der Geschwindigkeit wird außerdem vom Magnetometer unterstützt. Der Sensor misst die Stärke des Magnetfelds der Erde. Somit weiß das Gerät, wo sich der magnetische Norden befindet und kann als Kompass dienen. Mit Hilfe der Ausrichtung und den vorherigen Sensoren wird für die Navigation der Weg berechnet. Allerdings gilt es auch hier zu beachten, dass der Sensor durch starke Metallansammlungen ablenkbar ist und ungenau wird. Je nach Qualität des Sensors kann es auch schon durch vorbeifahrende Fahrzeuge zu Störungen kommen. [36]

Als letzter Sensor, der interessant für das sensorbasierte Tracking ist, kann man den Rotationsensor nennen, der auch als Gyroskop bezeichnet wird. Dieser Sensor erkennt durch das sogenannte Stimmgabelprinzip Drehungen um die Achsen des Geräts. Das heißt, sobald sich das Gerät dreht, ändert sich der Abstand zweier Stimmgabelzinken im Gerät und dadurch auch die Kapazität zwischen ihnen. Die Veränderung wird registriert und berechnet somit die Drehung des Geräts. [36]

In handelsüblichen Smartphones oder Tablets ist keiner der vorgestellten Sensoren zuverlässig robust und genau für die Verwirklichung einer präzisen Augmented Reality-Erfahrung. Das Ziel sollte sein, hybride Tracking-Systeme aus mehreren Sensoren zu nutzen, die eine unverzügliche 6DOF-(six-degrees-of-freedom)-Umgebungserkennung zu jeder Zeit und in allen Bedingungen liefern [2]. Für das sensorbasierte Tracking bedeutet dies, dass durch das GPS und den Beschleunigungssensor die Position des Nutzers bestimmt wird. Der Magnetometer gibt die Blickrichtung des Nutzers an und der Beschleunigungssensor und das Gyroskop letztendlich die Ausrichtung des Geräts. Somit eignen sich die Sensoren für eine performante Positionsbestimmung, die in unterschiedlichen Augmented Reality-Apps beispielsweise zur Anzeige von Point-Of-Interests genutzt werden kann (siehe Kapitel 3.4.2). Wie bereits beschrieben, muss aber mit Einschränkungen in Genauigkeit und Robustheit gerechnet werden.

2.1.4 Bildbasiertes Tracking

Beim bildbasierten Tracking wird zur Erkennung der Szene das Kamerabild des Geräts aufgezeichnet und gleichzeitig hinsichtlich bestimmter Merkmale (zum Beispiel Kontraste und Farbunterschiede) innerhalb des Bildes analysiert. Das hat unter anderem den Vorteil, dass das eingefangene Bild in Positionierung und Orientierung meist die zu erkennende Szene im Kamerabild enthält. Des Weiteren kann somit die Erkennung relativ zu den sich bewegenden Objekten geschehen, die Fehler bei der Berechnung der Positionierung der virtuellen Objekte fallen geringer aus und die Genauigkeit des Trackings wird abhängig von der sichtbaren Größe des Objekts. Allerdings leidet das bildbasierte Tracking laut You et al. [38] an fehlender Robustheit und hohem Rechenaufwand. [38]

Die ersten Methoden zur Bilderkennung nutzten laut Zhou et al. [39] sogenannte Marker oder Methoden der Linienerkennung. Weitere Marker wurden entwickelt und unterschiedliche Arten von Markern wurden studiert und auf ihre Effektivität getestet [7, 16, 22, 30]. Des Weiteren bestand die Bilderkennung aus der Erkennung von Punkten [13, 19], Linien [25], Konturen [15, 26] oder auch geometrischen Objekten [14]. Im Laufe der Jahre entwickelte sich die Bilderkennung weg von künstlich generierten Erkennungsmerkmalen über natürlich auftretende Merkmale hin zu modellbasierten Trackingmethoden. Dieses Trackingverfahren nutzt zur Bilderkennung Merkmale eines Objekts in Form eines CAD-Modells oder eines 2D-Templates. Dabei werden vor allem

Linien oder Kanten betrachtet (siehe Abbildung 2.3), was gegenüber Textur oder flächenbasierten Trackingverfahren den Vorteil mit sich bringt, dass dieses Erkennungsverfahren weniger anfällig für Lichtveränderungen ist und relativ einfach und effizient rechnerisch umsetzbar. Die ersten Vorstellungen solcher Methoden kamen von Comport et al. im Jahr 2003 [9] oder auch 2006 [10]. Comport et al. [10] sehen den größten Vorteil der Bilderkennungsverfahren darin, dass sie sehr exakt sind, warnen allerdings davor, dass bei falscher oder gar keiner Initialisierung die Abweichungen sehr stark variieren können. Wie in Abbildung 2.3 zu sehen, ist bei einem Kantenerkennungsverfahren außerdem ein Modell der jeweiligen Szene beziehungsweise des zu erkennenden Objekts von Nöten. Ist dieses Modell nicht vorhanden, muss auf andere Techniken zurückgegriffen werden. [10, 39]

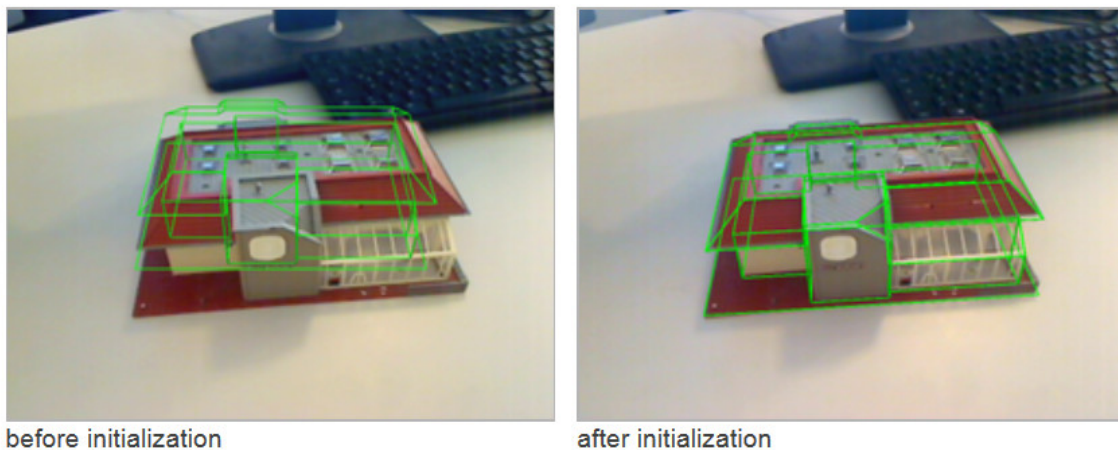


Abbildung 2.3: Bildbasiertes Tracking mit Hilfe von Kantenerkennung. [29]

Nach der Initialisierung der Szene ist es notwendig, für jedes weitere Frame die Position zu kennen und Veränderungen innerhalb der Szene nachzuverfolgen. Um ein gutes AR-Erlebnis bieten zu können, sollte die Augmentierung auch richtig dargestellt werden, sobald sich die Kamera oder Objekte in der Szene bewegen. Hierfür gibt es weiterhin diverse Verfahren, darunter SLAM (simultaneous localization and mapping) [12] oder die Echtzeit-Einschätzung der Bewegung der genutzten Kamera [31]. Weitere Verfahren stellen Comport et al. [10] in ihrer Arbeit vor.

2.1.5 Zusammenfassung

Schlussendlich bietet Augmented Reality durch die Überlagerung eines Echtbildes mit digitalen Informationen vielfältige Möglichkeiten der Informationsanzeige und -übermittlung. Damit die Einbettung der digitalen Informationen beziehungsweise Objekte exakt und perspektivisch korrekt funktioniert, muss auf die vorgestellten Trackingverfahren zurückgegriffen werden. Hierbei ist darauf zu achten, dass die Wahl zwischen sensorbasiertem oder bildbasiertem Tracking je nach vorhandenen Anforderungen und gegebenen Umständen stattfinden sollte und eine mögliche Kombination aufgrund ihrer Vor- und Nachteile in Betracht gezogen werden muss.

2.2 Tourist Guides

Nachdem die technischen Grundlagen von Augmented Reality in den vorangegangenen Kapiteln erläutert wurden, sollen im folgenden Abschnitt Tourist Guides analysiert werden. Dabei werden Anforderungen und Anwendungsgebiete, die jeweilige Nutzergruppe und auch die Problemfelder unterschiedlicher Tourist Guides dargestellt, um letztendlich die Anforderungen für diese Arbeit zu definieren und die gefundenen Lösungsansätze aufzuzeigen.

2.2.1 Anforderungen

Tourist Guide ist der englische Begriff für Fremdenführer, was laut der Definition des Duden [18] jemanden definiert, "der mit entsprechender Qualifikation gegen Bezahlung fremden Besuchern die Sehenswürdigkeiten eines Ortes, Gebäudes oder Ähnlichem zeigt und erläutert". In dieser Arbeit sollen allerdings keine Personen, sondern Programme als Fremdenführer dienen. Die genannte Definition spiegelt einen Großteil der Anforderungen an einen Tourist Guide wider. Ein Tourist Guide muss Informationen über einen Ort oder ein Gebäude enthalten, diese aufzeigen und beziehungsweise oder erläutern können. Zusätzlich sollten die Informationen authentisch und belegbar sein, was durch die entsprechende Qualifikation sichergestellt wird.

Für einen Tourist Guide in Form eines Programms bedeutet dies, dass das Programm aus sachlichen, dokumentarischen Quellen Informationen über eine Sehenswürdigkeit wiedergeben muss. Dies ist bei Software-Anwendungen über Multimedia-Kanäle möglich. Somit muss diese in der Lage sein, Fotos, Audios oder Videos abzuspielen. Die Punkte "gegen Barzahlung" und "fremden Besuchern" wären insofern abgedeckt, wenn die Anwendung kostenpflichtig und für jeden öffentlich zugänglich ist.

2.2.2 Anwendungsgebiete

Aus den genannten Anforderungen ergeben sich unterschiedliche Anwendungsgebiete für Tourist Guide-Software. Laut Definition sollen Informationen zu Sehenswürdigkeiten wiedergegeben werden. Diese Sehenswürdigkeiten könnten etwa Museen, historische Gebäude und Orte, Monumente oder Ähnliches sein. Somit ist der Anwendungsort klar definiert, Unterschiede existieren jedoch je nach Art des Ortes.

Ein Tourist Guide in einem Museum muss die Informationen im Normalfall nicht zwingend visuell wiedergeben, da der Besucher die Ausstellungsstücke vor Ort betrachten kann. Hingegen ist beispielsweise ein Audio-Guide eine gute Möglichkeit, Zusatzinformationen an den Besucher weiterzugeben. Das gleiche gilt auch bei historischen Gebäuden und Orten oder einfach bei Sehenswürdigkeiten, die man ohne Einschränkungen betrachten kann. Es bietet sich an, dem Interessierten zusätzlich auch weitere Informationen visuell per Bilder oder Videos mitzuteilen. Somit erhält der Interessierte fundierteres Wissen und kann auch Dinge erfahren, die nicht vor Ort zu sehen sind. Trotz Allem ist es nicht möglich, bei den genannten Sehenswürdigkeiten die visuellen Zusatzinformationen im Kontext anzubieten. Das heißt, sie müssen in einem zusätzlichen Medium angezeigt werden, in dem die Sehenswürdigkeit selbst nicht zu sehen ist. Somit wird der Nutzer in den wenigsten Fällen kontextsensitive Informationen erhalten. Des Weiteren wird bei Fremdenführern erwartet, dass diese nicht nur an einem Ort für ein einziges Ausstellungsstück Informationen wiedergeben. Der Nutzen für Entwickler und Nutzer wäre in diesem Fall zu gering. Deshalb ergibt sich daraus, dass sich mobile Geräte gut als Tourist Guides eignen. Bei großen Gebäuden möchte man sich eventuell um dieses herumbewegen, um es von allen Seiten betrachten zu können. Somit lässt sich schlussfolgern, dass sich vor allem PDAs, Smartphones oder auch Tablets anbieten, da diese Geräte Informationen audiovisuell wiedergeben können und zusätzlich besonders mobil sind.

2.2.3 Nutzergruppe

Aus den bisher angeführten Punkten kann man die Nutzergruppe von Tourist Guides herauskristallisieren. Da bei Tourist Guides Informationen über Sehenswürdigkeiten weitergegeben werden, ergibt sich somit eine kultur- beziehungsweise geschichtsinteressierte Nutzergruppe. Das Alter der Nutzergruppe kann dabei ein breites Spektrum vom Jugendlichen bis hin zum Rentner abdecken, da sowohl Jugendliche auf Studienreisen als auch Touristen aller Altersklassen Interesse an Sehenswürdigkeiten haben. Für einen klassischen Fremdenführer kann man davon ausgehen, dass dieser hauptsächlich von älteren Generationen genutzt wird, wohingegen der Tourist Guide als

Software aufgrund der vorausgesetzten Technikaffinität vermutlich von jüngeren Nutzern bevorzugt wird. Das kann man zusätzlich aus der Tatsache schließen, dass die jüngere Generation mehr Zugang zu mobiler Software besitzt.

2.2.4 Zusammenfassung

Zusammenfassend kann man sagen, dass ein Tourist Guide Informationen zu Sehenswürdigkeiten für Interessierte aller Altersklassen zugänglich macht. Merkmal ist dabei das fundierte Wissen, das direkt vor Ort in Form von mündlicher Übermittlung oder in Form von multimedialen Inhalten an den Mann gebracht wird. Für diese Arbeit bedeutet dies, dass eine Software erstellt werden soll, die fundierte Informationen zu einem Gebäude liefert. Mobilität und Navigation sollen dabei eine Rolle spielen, da auch diese Punkte einen Tourist Guide definieren. Ein weiteres Herausstellungsmerkmal des AR Tourist Guides soll sein, dass die Informationen durch die Nutzung von Augmented Reality kontextsensitiv dargestellt werden und somit dem Nutzer ein möglichst realitätsnahes Erlebnis bieten.

2.3 Augmented Reality Tourist Guides

Im Vergleich zu den beschriebenen Tourist Guides geben Augmented Reality Tourist Guides die Informationen mit Hilfe von Augmented Reality-Technologie wieder. Wie bereits in Kapitel 2.1.1 beschrieben, wird somit die Information direkt auf den Ort des Geschehens projiziert und der Nutzer bekommt vor Ort ein kontextsensitives, immersives Erlebnis der Informationen. Im Folgenden sollen einige Beispiele von Augmented Reality Tourist Guides oder auch Augmented City Guides gezeigt werden, die die Grundsätze eines herkömmlichen Tourist Guides befolgen, aber um die AR-Komponente erweitert sind. Es wird gezeigt, was die besonderen Merkmale dieser Anwendungen sind und wie sie als Inspiration für diese Arbeit dienen. Des Weiteren sollen Problemfelder erkannt werden, die in der Entwicklung des Prototyps vermieden oder durch sinnvolle Lösungen bewältigt werden.

2.3.1 Chronovisor

Der Tourist Guide *Chronovisor* ist eine Applikation für die mobilen Betriebssysteme *iOS* und *Android*. Diese App ermöglicht dem Nutzer, sogenannte Zeitfenster auf das Kamerabild zu legen. Nutzer müssen dafür eine Stadtszene in Bucharest mit dem Smartphone oder Tablet abscanen. Das Gerät erkennt die Position mit Hilfe von GPS-Daten und dem Tracking eines optischen Markers. Als Ergebnis wird ein Videoausschnitt mit Szenen aus dem Leben der Menschen vor 80 Jahren im Vordergrund der identischen, realen Szene angezeigt. [8]

Der rumänische Entwickler *Colorbitor* hat zur Identifikation der richtigen Positionen auf einer appinternen Stadtkarte die relevanten Positionen markiert. Zusätzlich befinden sich an den Orten Markierungen auf dem Bürgersteig, damit die Anwendung die Szene anhand dieser tracken kann. Diese Markierungen sind sogenannte LLA-Marker (longitude, latitude, altitude). Werden sie gescannt, aktualisiert die App die GPS-Daten mit vorgegebenen, zum Marker gehörenden Koordinaten, sodass das Gerät erkennt, an welcher Position sich der Nutzer befindet. Für eine korrekte Positionierung des Videooverlays muss der Nutzer die Kamera des Geräts auf die richtige Stelle richten. Wenn der Nutzer alles richtig macht, erscheint auf dem Kamerabild ein Video das passgenau in die Umgebung eingebettet ist. So sieht man beispielsweise ein Video mit dem Verkehr und den Fußgängern der Vergangenheit auf der Straße, auf der im Kamerabild der Gegenwart die Fahrzeuge vorbeifahren (siehe Abbildung 2.4). Positiv daran ist, dass man die Vergangenheitsszene im heutigen Kontext erleben kann und der Nutzer dadurch einen Eindruck bekommt, wie sich die Stadt und das Leben der Bürger im Laufe der Jahrzehnte verändert haben. [8]

Als Schwachpunkt könnte man anführen, dass sich der Nutzer aufgrund der technischen Möglichkeiten am richtigen Ort aufhalten muss, um die AR-Erfahrung zu erleben. Zusätzlich kann



Abbildung 2.4: Beispielszene aus der Applikation *Chronovisor* von *Colorbitor* [8].

es aufgrund der Abhängigkeit von den optischen Markierungen auf den Gehwegen zu Initialisierungsproblemen kommen, sollten diese aus unterschiedlichen Gründen nicht mehr sichtbar sein (wie etwa durch Verdeckung durch andere Objekte oder durch Schäden am Gehweg). Aufgrund dessen und aufgrund der Tatsache, dass die App ausschließlich Videoausschnitte anbietet, ist der Nutzer in der Interaktion eingeschränkt. Außerdem ist die Veränderung von Lichtverhältnissen und Verdeckung nicht konfigurierbar, weshalb das augmentierte Element sehr unnatürlich erscheint und nicht optimal in die Szene eingebettet ist. Abbildung 2.4 zeigt, dass zum einen das Fensterformat des Videos sehr abrupte Grenzen zum Echtbild aufweist und weiterhin der Grauton des Videos die Illusion einer wirklichen Einbettung in das reale Szenario raubt. Andererseits erhält das dargestellte Video durch die Wahl der Farben einen klassischen Flair, der dem Nutzer das Gefühl einer Zeitreise vermittelt. Bezogen kann die Anwendung als eigenständiger Kanal über den *Junaio Browser* von *Metaio* (siehe Kapitel 3.4.1). [8]

2.3.2 Landauer Walk

Die im Jahr 2014 entwickelte Anwendung *LandauerWalk* soll es dem Nutzer ermöglichen, besondere Stationen aus dem Leben des ehemaligen Präsidenten des *FC Bayern München* zu erfahren. Der Tourist Guide entstand in Kooperation zwischen dem *Bayerischen Rundfunk* und *mmc*, einer Agentur für interaktive Medien. Basierend auf *Apache Cordova* wurde das Programm plattformübergreifend für mobile Geräte mit *Android* und *iOS* entwickelt, unter Einbeziehung der *Open Graphics Library* und dem *Wikitude SDK*. [20]

Die journalistischen Inhalte werden mittels geolokalisiertem Storytelling mit Hilfe von Augmented Reality-Elementen vor Ort erzählt. Zu insgesamt neun Stationen kann der Nutzer geführt werden, um dort jeweils die Geschichte von Kurt Landauer zu erleben. Nähert sich der Nutzer der gewünschten Stelle an, muss er sein Gerät richtig ausrichten, um die AR-Erfahrung zu machen. Für die Umgebungserkennung setzt die Anwendung auf sensorbasiertes Tracking mittels GPS und Kompass. Dies hat den Vorteil, dass wenig Rechenaufwand betrieben werden muss, allerdings sind die Sensoren wie in Kapitel 2.1.3 beschrieben durch äußere Faktoren beeinflussbar. Des Weiteren mangelt es der App aufgrund des sensorbasierten Trackings an Genauigkeit. Die Überlagerung muss durch die Bewegung und Ausrichtung des Geräts optimiert werden. [35]

Nach erfolgreichem Tracking überlagert ein Bild oder Video von historischen Szenen aus Kurt Landauers Lebenszeit die heutige Stadtszene auf dem Display des Nutzers (siehe Abbildung 2.5). Zusätzlich erhält der Nutzer viele weitere Informationen zur Geschichte in Form von Texten, Fotos und Erzählungen via Ton. Weiterhin ist es mit dem *LandauerWalk* auch möglich, die Inhalte an jedem beliebigen Ort zu erfahren. Befindet man sich also nicht vor Ort und kann die jeweilige Szenerie für die AR-Erfahrung nicht tracken, bietet die Anwendung die Möglichkeit, die Erfahrung mittels einer Offline-Tour zu machen. Diese Offline-Tour ist rein virtuell und funktioniert wie ein interaktives Hörspiel mit 360-Grad-Panoramen. Bei der Live- als auch bei der Offline-Tour verändert der Nutzer über eine Leiste die Transparenz des virtuellen Bildes. Somit ist der städtische Kontext und die Veränderung von damals zu heute besser für den Nutzer zu erleben. [35]

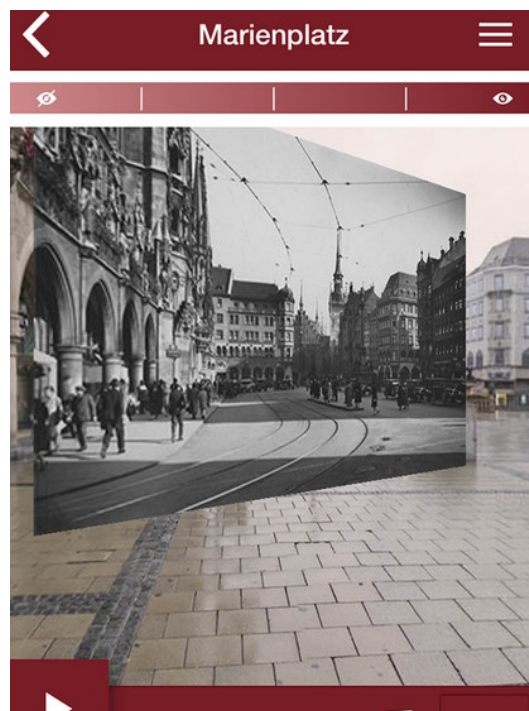


Abbildung 2.5: Beispielszene aus der Applikation *LandauerWalk* [35].

Möchte man die einzelnen Stationen der App ansteuern, kann man sich von der Software mittels einer Rundtour führen lassen, oder man wählt auf der Übersichtskarte eine beliebige Position aus und wird anschließend dorthin navigiert. Unabhängig von der Position ist man hingegen bei der Offline-Tour mit ihren Offline-Funktionen. Somit werden auch Performanz-Probleme umgangen, da die Datenmengen vorab geladen werden können. Vor Ort werden die Daten gestreamt, je nach Internet-Verbindung kann es aber zu Performanz-Einbußen kommen. Zum Zeitpunkt dieser Arbeit ist die Applikation nicht mehr offiziell verfügbar. [35]

2.3.3 Zeitfenster

Zeitreisen via App versprechen vier Masterstudenten der *Hochschule der Medien* in Stuttgart mit ihrer Anwendung mit dem Namen *Zeitfenster*. Mit Hilfe der Applikation soll das Smartphone oder Tablet einen Blick auf vergangene Zeiten freigeben. *Zeitfenster* soll sowohl der Vermittlung von Wissen und Kultur dienen, als auch Touristen vor Ort locken und ihnen eine erlebnisorientierte Stadttour bieten. Wie auch beim *LandauerWalk* (siehe Kapitel 2.3.2) überlagern Fotografien und Videos den aufgenommenen Ort, um dem Nutzer die Vergangenheit zu zeigen. Audioguides und Texte erweitern den Informationsgehalt der App. Bisher gibt es die App für *Android* und *iOS* mit den Städten Berlin, Stuttgart, Leipzig, Weimar und Nürtingen.

Die Funktionen rund um Augmented Reality sind Location-based Services, das heißt der Nutzer erfährt die Inhalte sobald er sich vor Ort befindet. Somit setzt auch *Zeitfenster* wie schon der *LandauerWalk* auf ein sensorbasiertes Tracking mit den bereits beschriebenen Vor- und Nachteilen (siehe Kapitel 2.1.3). Vor allem die Genauigkeit leidet aufgrund des gewählten Trackingverfahrens. Die Stationen lassen sich per Rundgang ansteuern oder einzeln unabhängig voneinander. Vor Ort legt der Nutzer die historische Aufnahme direkt über das Kamerabild und erlebt so die Vergangenheit in der Gegenwart. Die Daten werden dabei live vom Server geladen, was bei schlechter Internetverbindung zu Performanz-Problemen führen kann. [34]

2.3.4 TimeTraveler

Die letzte in dieser Arbeit vorzustellende Applikation *TimeTraveler* ist bei *Timetraveler Augmented Ltd.* im Jahre 2014 entstanden. Die Funktionen der App ähneln stark den bisher vorgestellten Augmented City Guides. Auch bei *TimeTraveler* wird ein Videoausschnitt aus der Vergangenheit auf den aufgenommenen Ort und das Kamerabild gelegt. Die App beschränkt sich dabei auf elf Punkte an der Mauergedenkstätte entlang der Bernauer Straße in Berlin. [1]

Wird *TimeTraveler* geöffnet, erfährt der Nutzer in einem Menü, wie mit der Software verfahren werden muss, um das Augmented Reality-Erlebnis zu starten. Zusätzlich hat man auch die Möglichkeit sofort mit den Funktionen der Software zu beginnen. Zu den Funktionen gehören die Navigation und die Augmented Reality-Erfahrung. Für das AR-Erlebnis muss die Kamera des Geräts auf die richtige Stelle ausgerichtet sein, daraufhin wird der entsprechende Kurzfilm auf die Szene gelegt. [1]



Abbildung 2.6: Beispielszene aus der Applikation *TimeTraveler* von *Metaio* [1].

Die Anwendung arbeitet mit GPS-Tracking und ortsbezogenen optischen Markern. Das GPS-Tracking schränkt durch die Positionserkennung die Auswahl der optischen Marker bei der darauffolgenden Bildererkennung ein, um im bildbasierten Tracking anhand des Markers die genaue Position zu erkennen. Dadurch wird die Augmentierung an exakt der richtigen Stelle platziert. Wird die Kamera beispielsweise auf die Wand der Wohnung von Frieda Schulze gerichtet, sieht man an dieser Wand einen Videoausschnitt, bei dem Frau Schulze aus dem Fenster flieht (siehe Abbildung 2.6). Durch die kontextsensitive Informationsbereitstellung in der gegenwärtigen Stadtszene,

bekommt der Nutzer einen außerordentlich emotionalen Einblick in das an dieser Stelle Geschehene. Außerdem soll der Nutzer durch das AR-Erlebnis das Gefühl bekommen, sich während des Geschehens direkt vor Ort zu befinden. [1]

Auch in *TimeTraveler* gibt es zu jeder Station zahlreiche Informationen zu dem Ort oder dem historischen Moment. Befindet man sich nicht vor Ort, bietet die Applikation die Möglichkeit, die AR-Funktion zu testen, indem man für drei Stationen die Kamera auf einen Test-Bildschirm richtet, der die reale Szene simulieren soll. Durch das in der Anwendung genutzte bildbasierte Tracking erkennt das Gerät die Szene (wie auch bei der Live-Tour vor Ort) und projiziert das entsprechende Video an die richtige Stelle. Da die digitalen Augmentierungen Videos sind und nicht in einen 3D-Raum projiziert werden müssen, reicht ein einfaches Marker-basiertes Tracking für die Erkennung der Szene und Platzierung der Videos. Sollte trotz Allem das Tracking einmal nicht klappen, wurde eine Fallback-Funktion in die App eingebaut, die nach zehn Sekunden ohne Erkennung den Inhalt als Standbild darstellt. Wie auch schon bei den bereits vorgestellten Augmented City Guides hat auch der *TimeTraveler* keine richtige Steuerung der Lichtverhältnisse, was ebenfalls zu einem ungewollten Hervorheben des virtuellen Inhalts führt. Die *TimeTraveler*-App ist sowohl für iOS- als auch für Android-Geräte im jeweiligen offiziellen Store beziehbar. [1]

2.3.5 Zusammenfassung

Die vorgestellten Augmented Reality Tourist Guides in Kapitel 2.3 zeigen, dass auch hier die in Kapitel 2.2 angeführten Kriterien für Tourist Guides gelten. Alle angeführten Apps geben dem Nutzer zu einer Sehenswürdigkeit fundierte Informationen in multimedialer Form und sind dabei portabel. Zusätzlich macht die Mobilität der Software und die Navigationsfähigkeit aller Apps die Programme zu einem Fremdenführer im weiteren Sinne.

Die Augmented City Guides zeigen, dass Augmented Reality sinnvoll genutzt werden kann, um dem Nutzer ein besonderes Erlebnis beziehungsweise eine besondere Erfahrung zu bieten und ihm so historische Informationen zu vermitteln. In allen angeführten Beispielen wurde die virtuelle Information bestmöglich in das reale Umfeld eingebettet, um dem Nutzer die Informationen im richtigen Kontext zu präsentieren. Allerdings hebt sich bei allen Anwendungen das Virtuelle deutlich vom Kamerabild ab, da Lichtverhältnisse und Verdeckung noch nicht optimal gelöst werden konnten. Auch sind einige der vorgestellten Anwendungen durch das gewählte Trackingverfahren stark eingeschränkt. Neben den bereits angesprochenen Funktionen von (AR-) Tourist Guides soll in dieser Arbeit auch ein großes Augenmerk auf diese Punkte gelegt werden, um dem Nutzer eine möglichst realitätsnahe AR-Erfahrung zu bieten.

3 Konzeption des Prototypen

Wie in den vorherigen Kapiteln erläutert, bietet es sich bei Tourist Guides an, virtuelle Informationen in das reale Umfeld zu augmentieren, um dem Nutzer ein möglichst informatives und authentisches Erlebnis der Vergangenheit zu bieten. Auf Basis der vorgestellten Technologien soll nun in diesem Kapitel die Entwicklung des Prototypen eines On-Site 3D Augmented Reality Tourist Guides erörtert werden. Hierbei wird auf die einzelnen Schritte von der Konzepterstellung mit Zielsetzung, Anforderungsanalyse und Architekturbeschreibung über die Produktion mit Modellanpassung, markerlosem Tracking und dem Einpassen der Geometrie in den städtebaulichen Raum bis hin zur Implementierung der Benutzeroberfläche und der Hintergrundinformationen eingegangen. Die Anwendung hat den Namen *HistoryTraveller* erhalten, in Anlehnung an das Vorbild *TimeTraveler* und da man ein Stück Geschichte mit Hilfe der App erleben können soll.

Zu Beginn des Projekts wurde das Konzept ausgearbeitet. Nach Aufstellung der Ziele und der dafür notwendigen Anforderungen wurden diese in einem Design Document festgehalten. Das Design Document sollte während der gesamten Entwicklung als Orientierung dienen, um das letztendliche Ziel der Anwendung nicht aus den Augen zu verlieren. Das Dokument besteht aus fünf Teilen, die das Konzept allgemein, die Zielanwendung, die Zielgruppe, die Bedienung und letztendlich die Technologie der Anwendung beschreiben.

3.1 Zielsetzung

Wie bereits im zweiten Kapitel beschrieben, soll die Anwendung im finalen Stadium einen Augmented Reality Tourist Guide beziehungsweise einen Augmented City Guide darstellen. Die Applikation soll es dem Nutzer ermöglichen, die *Alte Hauptsynagoge* an ihrem ehemaligen Standort in München zu erleben. Eine Navigationsfunktion soll den Nutzer bei Bedarf an den richtigen Ort leiten. Vor Ort sollen zwei Positionen zwei unterschiedliche Sichtweisen auf das Gebäude bieten (siehe Bild 3.1). Dort angekommen soll mit Hilfe eines geeigneten Trackingverfahrens die Synagoge als 3D-Modell auf dem Kamerabild in die städtische Umgebung eingepasst werden. Der Nutzer soll die Synagoge dort sehen, wo sie zu ihrer Zeit stand und außerdem Hintergrundinformationen zum Gebäude aufrufen können.



Abbildung 3.1: Städtische Umgebung rund um den Oberpollinger in München.

Zur Betrachtung der Synagoge wurden eine nahe Ansicht und eine entfernte Frontansicht festgelegt. Aufgrund der umgebenden Gebäude ist die Auswahl an verschiedenen Positionen relativ gering. Die umgebenden Häuser stehen sehr nah am Oberpollinger-Gebäude, wodurch sich viele Positionen ergeben, die zu nah am Oberpollinger sind und keine ansprechenden Ansichten auf das Gebäude ermöglichen. Von diesen Positionen würde man das Gebäude nicht ausreichend im Kamerabild erfassen können. Wie in Abbildung 3.1 zu sehen ist der ehemalige Standort von schmalen Straßen und einem direkt anschließenden Gebäude umgeben. Würde man die Herzog-Max-Straße oder auch die Kapellenstraße zur Betrachtung nutzen, dann würde sich dem Nutzer keine gute Erfahrung des Modells bieten. Aufgrund der Nähe zu dem großen Gebäude würde der Nutzer nur einen kleinen Ausschnitt des Gebäudemodells sehen. Aus diesen Gründen wurde einerseits Position 1 an der Maxburgstraße (siehe Abbildung 3.1) gewählt. Hier hat der Nutzer eine gute Entfernung zum Gebäude, um die Frontseite komplett zu sehen. Zusätzlich ist an dieser Stelle etwas Platz, um sich auch an die Seite der Synagoge zu bewegen. Als Zusatz soll eine entfernte Position (Position 2 in Abbildung 3.1) am Lenbachplatz zur Verfügung gestellt werden, um die Synagoge eingebettet in das städtische Umfeld zu betrachten, mit weniger Fokus auf die Architektur des Gebetshauses.

Aufgrund der Mobilität eignen sich Smartphones und Tablets als Anwendungsgeräte. Die wichtigsten Vertreter von mobilen Betriebssystemen *Android* und *iOS* sollen unterstützt werden und, auch aufgrund der Kompatibilität mit dem *Junaio Browser*, sollen *Microsoft Phones*, *Blackberrys* und andere mobile Systeme vernachlässigt werden. Sollte die Entwicklung über den Prototypen hinaus gehen, dann müssten diese Geräte allerdings nochmals aufgegriffen werden. Wie schon in Kapitel 2.2.3 besprochen soll die App ein möglichst breites Spektrum an Nutzern abdecken. Das heißt, es sollen Touristen und Kultur-, Architektur- und Geschichtsinteressierte jeder Altersgruppe angesprochen werden. Da die Altersspanne voraussichtlich stark variiert, sollte die Benutzeroberfläche möglichst einfach und intuitiv gehalten sein, um auch weniger technisch versierten Nutzern eine einfache Bedienung zu ermöglichen. Die Bedienung soll händisch mittels Touchscreen erfolgen, wobei hauptsächlich digitale Buttons vorhanden sind. Optional sollen auch die Hardware-Buttons des Geräts genutzt werden können. Um auch weniger technisch versierten Nutzern die Bedienung und die Funktionen der Anwendung nahe zu bringen, soll ein Tutorial aus Text und Bild das Wichtigste erklären.

Weiterhin soll der Prototyp die typischen Funktionen eines Tourist Guides erfüllen (siehe Kapitel 2.2). Dadurch ergeben sich neben der bereits erwähnten Navigation auch die Bereitstellung von fundierten Informationen, die dem Nutzer interessante Fakten und zusätzliches Wissen über die *Alte Hauptsynagoge* vermitteln. Des Weiteren soll eine Offline-Tour das Funktionsrepertoire der Applikation abschließen. Wie bereits beim *LandauerWalk* (Kapitel 2.3.2) oder auch beim *TimeTraveler* (Kapitel 2.3.4) soll der Nutzer die Möglichkeit bekommen, die Funktionen der App auch von jedem anderen Ort aus zu testen und die Synagoge im heutigen städtebaulichen Szenario zu erleben.

3.2 Anforderungen

Aus den beschriebenen Zielen ergeben sich folgende Anforderungen für die Umsetzung der Applikation. Das Gerät muss entweder das Betriebssystem *Android* oder *iOS* besitzen und die Metaio-Anwendung *Junaio Browser* muss darauf lauffähig sein. Das Smartphone oder Tablet benötigt eine funktionsfähige Kamera zur Initialisierung der AR-Erfahrung. Zum Laden der Daten vor Ort ist außerdem eine bestehende Internetverbindung nötig.

3.3 Entwicklungsumgebung

Zur Entwicklung der Szenerie stellt die Entwicklungsfirma den *Metaio Creator* zur Verfügung. Als Alternative stehen die *Augmented Reality Experience Language AREL* oder die native Ent-

wicklung zur Auswahl. Im Folgenden sollen diese Entwicklungsumgebungen vorgestellt werden und es wird erläutert, welche Variante aus welchen Gründen gewählt wurde.

3.3.1 Native Entwicklung

Um eine App zu entwickeln, gibt es die einfache Methode, diese nativ für das entsprechende Betriebssystem zu programmieren. Hierbei verwendet man die jeweilige Programmiersprache, die auf dem Zielsystem lauffähig ist. *Metaio* stellt dafür das Framework *Metaio SDK* zur Verfügung, mit dessen Hilfe das Programmieren von Applikationen für Windows und Unity sowie auch für die mobilen Plattformen *Android* und *iOS* möglich ist. In dem Framework sind die Komponenten für das Erfassen, das Tracking, die Sensoren und das Rendern enthalten. Mit Hilfe des Frameworks werden diese Komponenten angesprochen und somit wird eine einfache Implementierung von Augmented Reality-Applikationen ermöglicht. [29]

Da bei dieser Arbeit eine mobile Applikation für Smartphones beziehungsweise Tablets entstehen soll, hieße das, dass man mit Java für *Android* und mit Objective-C für *iOS* arbeiten müsste. Wählt man diese Methode der Programmierung, hat dies den Vorteil, dass jedes Detail durch die native Programmierung genauestens vorgegeben werden kann. Somit programmiert man die Applikation von Grund auf neu. Dadurch hat man mehr Einfluss auf das finale Resultat. Allerdings entsteht dadurch der Nachteil, dass gute Programmierkenntnisse vorhanden sein müssen und für jedes System eine eigene Applikation entwickelt werden muss. Im Zuge dieser Arbeit steht allerdings weniger der programmiertechnische Aspekt im Vordergrund, sondern der Vergleich der unterschiedlichen Technologien in Bezug auf die Realisierung der Augmented Reality-Erfahrung. [29]

3.3.2 AREL



Abbildung 3.2: Überblick über die Funktionsweise von AREL [29].

Metaio entwickelte neben dem oben beschriebenen SDK auch die *Augmented Reality Experience Language (AREL)*. Diese Programmiersprache basiert auf den üblichen Webtechnologien XML, Javascript und HTML5. Sie erlaubt das Skripten auf dem Gerät und Interaktionen ohne Server Kommunikation. Vorteil dieser Sprache ist, dass man durch die Vereinfachung vieler Schritte, wie etwa dem Erstellen eines User Interfaces oder das Individualisieren der Interaktivität, auch mit wenig Programmierkenntnissen vollwertige Apps programmieren kann. Ein Hauptvorteil ist

die Plattformunabhängigkeit, da es reicht, eine Applikation zu entwickeln. Diese ist dann über den *Junaio Browser* (siehe Kapitel 3.4.1) auf fünf verschiedenen Plattformen lauffähig. [29]

Mit *AREL* ist es also möglich, interaktive Augmented Reality-Erfahrungen zu erstellen, die anschließend über den *Junaio Browser* oder das *Metaio SDK* laufen. Die resultierende Applikation besitzt einen OpenGL Renderer, der die 3D-Objekte und das Szenario darstellt. Eine XML-Datei definiert hierbei die Eigenschaften der Objekte. Ein HTML5-Layer zeigt dem Nutzer das Graphical User Interface (GUI) an und interagiert über eine Javascript-Brücke mit den Objekten oder der Szenerie (siehe Abbildung 3.2). Somit werden Eingaben des Nutzers registriert und an die Szenerie weitergegeben [29]

3.3.3 Metaio Creator

Am einfachsten erstellt man Augmented Reality-Applikationen mit dem von *Metaio* entwickelten *Metaio Creator*. Nach der Installation besitzt man mit dem mächtigen Programm mehrere Möglichkeiten, unterschiedlichste Augmented Reality-Apps anzufertigen. Als Erstes muss man hierfür über das entsprechende Menü ein Objekt zum Tracken auswählen. Der Creator realisiert bildbasierte Verfahren zur Umgebungserkennung. Somit kann das Objekt unter anderem ein Bild, eine 3D-Map oder auch ein Kantenmodell sein (siehe Abbildung 3.3). Wie bereits in Kapitel 2.1.2 beschrieben besitzen diese unterschiedlichen Trackingverfahren je nach Szenerie und Anforderungen verschiedene Vor- und Nachteile.

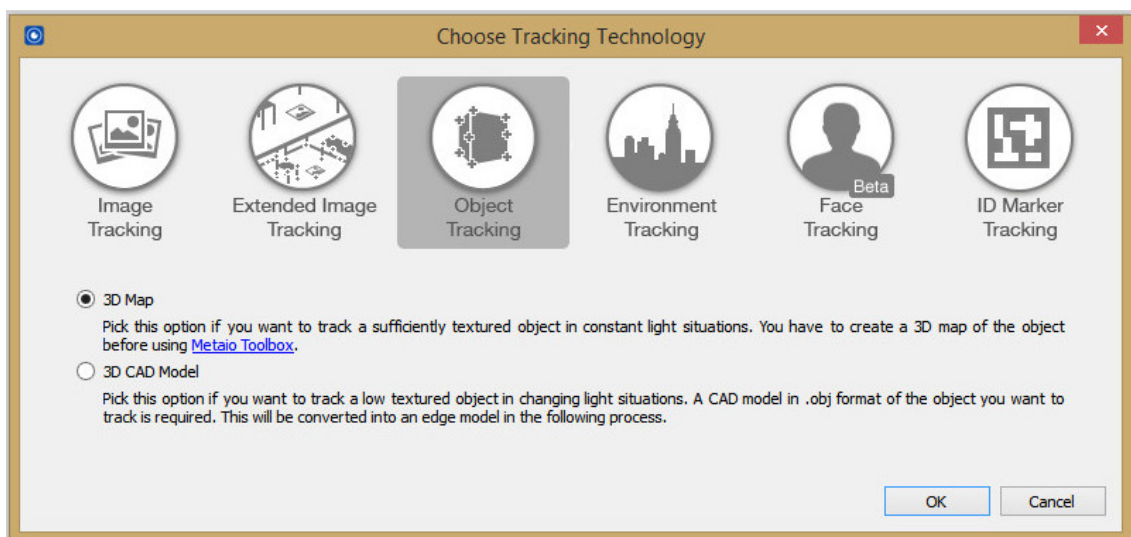


Abbildung 3.3: Auswahl der Trackingverfahren im *Metaio Creator* [29].

Hat man sich für ein Verfahren entschieden, setzt man als nächstes unterschiedliche Objekte in die Szene. Diese Objekte können das später erscheinende 3D-Objekt oder ein Videofilm sein oder aber auch Elemente, die als GUI schon beim Start erscheinen sollen. So ist es mit dem *Metaio Creator* beispielsweise auch möglich, Social Media Elemente mit einzubauen oder Schriftzüge und Buttons, die zur Bedienung der Applikation dienen sollen. Die Objekte werden mit Hilfe von Drag & Drop verschoben und platziert. Über die Einstellungen der Elemente werden diese nach den eigenen Wünschen konfiguriert, ohne dass Programmierkenntnisse erforderlich sind. Diese Möglichkeiten spiegeln aber auch Vor- wie Nachteile des *Metaio Creators* wider. Einerseits ist die Erstellung von eigenen Apps ohne spezifische Vorkenntnisse möglich, andererseits wird die Konfiguration durch die Vorgaben des Programms eingeschränkt.

Um auf die entwickelten Applikationen zuzugreifen, muss man diese als eigenen Kanal für den *Junaio Browser* von *Metaio* exportieren, oder man exportiert das Projekt als AREL-Paket. Letzteres kann nun über die entstehenden XML- und HTML5-Dateien weiter modifiziert werden,

wodurch eine Schnittstelle zwischen der AREL-Programmierung und dem Erstellen von Apps mit Hilfe des *Creators* entsteht.

Diese Arbeit wird mit der zuletzt beschriebenen Methode entwickelt. Einerseits ist durch den Creator das schnelle Umsetzen der Augmented Reality-Erfahrung möglich, aber durch zusätzliches HTML-Programmieren und Anpassungen über *AREL* kann andererseits die Individualisierung nach den in Kapitel 3 gesetzten Anforderungen und Zielen realisiert werden. [29]

3.4 Anwendungen

Basierend auf den angesprochenen architektonischen Möglichkeiten sind viele unterschiedliche Anwendungen entstanden, beziehungsweise ist es für jeden möglich, neue Augmented Reality-Anwendungen zu erstellen. Sollten die Anwendungen nicht nativ programmiert worden sein, müssen sie zur Nutzung als Channel für den *Junaio Browser* veröffentlicht werden, um genutzt werden zu können.

3.4.1 *Junaio Browser*

Der *Junaio Browser* von *Metaio* dient als plattformübergreifende Schnittstelle für Augmented Reality-Anwendungen. In diesem Browser gibt es mehrere Möglichkeiten, AR zu nutzen und zu erfahren. Startet man das Programm mit einem Smartphone oder Tablet, wird als Erstes der Channel "Junaio Around You" gestartet. Dieser Channel zeigt dem Nutzer auf seinem Kamerabild sogenannte Points-Of-Interests (POIs) an. Diese POIs werden als Stecknadeln markiert und weisen auf interessante Stellen in der unmittelbaren Umgebung des Nutzers mit Namen, Richtung und Entfernungsangabe hin. Zusätzlich kann man sich die Punkte wie bei einer Navigationsapplikation mit eigener Ortsangabe auf einer Karte anzeigen lassen. Für die Erkennung der Position, an der sich der Nutzer befindet, wird hier das in Kapitel 2.1.3 beschriebene sensorbasierte Verfahren verwendet.

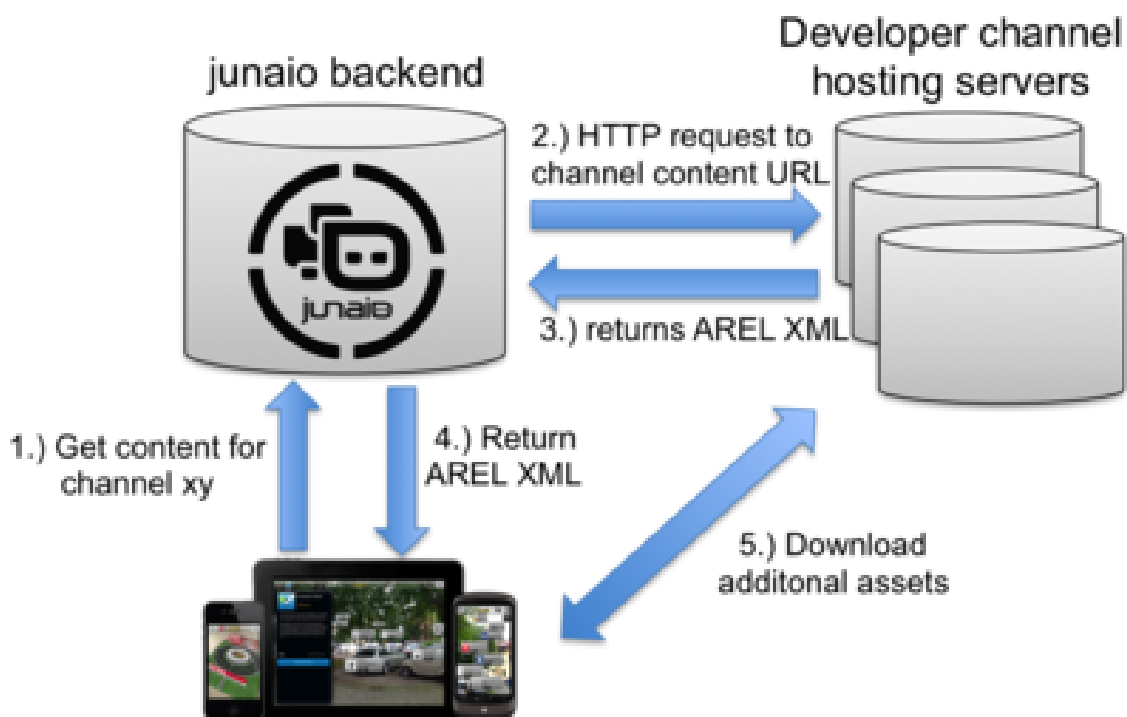


Abbildung 3.4: Überblick über die Funktionsweise der Junaio-Channels [29].

Die Channels im *Junaio Browser* funktionieren wie eine Webseite. Eine Anfrage nach dem Inhalt eines Channels wird vom Junaio-Backend bearbeitet, über die entsprechende URL wird der Channel auf dem Host-Server (meistens die *Metaio Cloud*) aufgerufen, der wiederum eine AREL XML zurückgibt. Diese XML wird durch das Backend an das Gerät weitergeleitet. Anschließend holt sich das Gerät noch weitere Assets wie etwa das HTML, Bilder, Modelle oder Ähnliches und zeigt diese auf dem Zielgerät an (siehe Abbildung 3.4).

Ein Vorteil der Nutzung des *Junaio Browsers* ist die Plattformunabhängigkeit, da der Browser sowohl auf iOS- als auch auf Android-Geräten lauffähig ist. Somit muss der Ersteller der Anwendung keine individuellen Anpassungen vornehmen. Der Nachteil allerdings ist, dass die als Channel erstellte Applikation somit von dem Browser und daraus resultierend auch von einer bestehenden Internetverbindung abhängig ist. Zusätzlich sind die Funktionen auf die Möglichkeiten des Browsers beschränkt. Somit können im Channel nicht ohne Weiteres die Hardware-Buttons des Geräts verwendet werden. Außerdem ist die Menüleiste von *Junaio* stets im Sichtfeld des Nutzers und kann nicht weggeschaltet werden. Des Weiteren hängt die Performanz auch stark von der Lauffähigkeit des Browsers ab. [29]

3.4.2 Beispiel

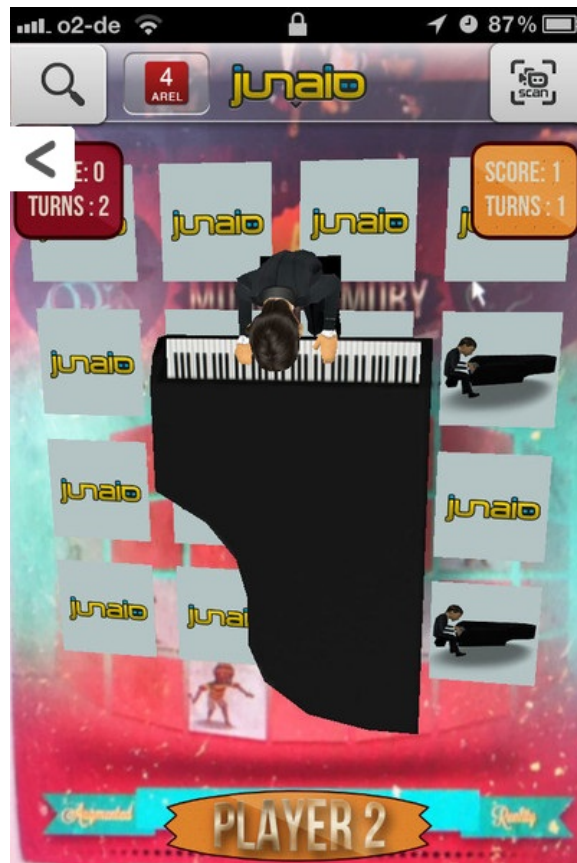


Abbildung 3.5: Beispielapplikation im *Junaio Browser* von *Metaio* [29].

Ein Beispiel von einem Augmented Reality-Kanal (zu sehen in Abbildung 3.5) soll im Folgenden die Möglichkeiten von Anwendungen in *Junaio* aufzeigen. Die einfachste Anwendung ist das Erkennen eines markanten Bildes mit Hilfe von bildbasiertem Tracking und das Projizieren eines 3D-Objekts auf das Bild. Durch Bildererkennung weiß das Gerät, wo das Objekt hinprojiziert werden muss. Erweitert man den Channel um weitere Funktionen, so ist es möglich, Animationen des Objekts und sogar durch Gesten ausgelöste Animationen hinzuzufügen. Mit diesen einfachen

Komponenten kann ein Augmented Reality Memory Spiel realisiert werden. Hierbei scannt man die reale Spieloberfläche mit der Kamera des Smartphones oder Tablets, sodass die Karten darauf als virtuelle Objekte erscheinen. Aufgedeckt werden die Karten durch Antippen des Touchdisplays an den entsprechenden Stellen. Deckt man das richtige Paar auf, wird eine kurze Animation der Figur abgespielt, die sich auf den Karten befindet (siehe Abbildung 3.5). [29]

3.5 Zusammenfassung

Die beschriebenen Funktionen und Beispiele der Metaio-Technologien dienen als Grundbaustein für die Funktionen des On-Site 3D Augmented Reality Tourist Guides. Die AR-Erfahrung wird mit Hilfe des *Creators* erstellt und dabei werden bildbasierte Erkennungsverfahren realisiert. Da allerdings eine 3D-Szene erkannt werden soll, reichen einfache 2D-Trackingverfahren wie bei dem beschriebenen Memory nicht aus. Hierfür werden die vom *Creator* zur Verfügung gestellten 3D-basierten Bilderkennungsverfahren verwendet (siehe Kapitel 4.3). Wie bereits beschrieben, werden Details wie die Benutzeroberfläche und weitere Funktionen mit Hilfe von AREL und HTML5-Elementen verbessert und spezifiziert. Zuletzt wird die Applikation in die *Metaio Cloud* hochgeladen, sodass diese über einen Channel im *Junaio Browser* gestartet werden kann. Dadurch ist der Tourist Guide für iOS- und Android-Geräte zugänglich, muss allerdings mit Einschränkungen (in Kapitel 3.4.1 besprochen) leben. Für die Umsetzung dieses Prototyps werden diese aufgrund der Einfachheit der Umsetzung in Kauf genommen.

4 Entwicklung des Prototypen

Nachdem das Konzept ausgearbeitet wurde und die Anforderungen, Ziele und Voraussetzungen definiert sind, ist der nächste Schritt die Produktion und Implementierung des Prototyps. Die Grundfunktionen der Applikation sollen hier umgesetzt sowie die Benutzeroberfläche gestaltet und mit Informationen gefüllt werden. Des Weiteren werden das 3D-Modell für die Anzeige und das Tracking zur Initialisierung der Anzeige realisiert.

Im Folgenden wird als Erstes die Anfertigung der Benutzeroberfläche mit dazugehöriger Evaluierung beschrieben. Anschließend wird die Modellanpassung und die Umsetzung des Trackings aufgezeigt, um im letzten Schritt der Produktion das Modell und das Tracking zusammenzusetzen. Beim Einpassen der Geometrie in das Szenario wird dabei auf die entscheidenden Punkte Transparenz und Lichtverhältnisse eingegangen.

4.1 Benutzeroberfläche

Die Benutzeroberfläche (GUI) ist der wichtigste Anlaufpunkt für den Nutzer. Kommt der Nutzer mit der GUI nicht zurecht, dann wird er oder sie schon bei der Bedienung scheitern und die Anwendung wird trotz aller Funktionen bereits vorab in Frage gestellt. Zur Umsetzung einer benutzerfreundlichen GUI wurden die vorhandenen Augmented Reality Tourist Guides analysiert und versucht, die gewonnenen Erkenntnisse bei der eigenen Anwendung umzusetzen.

Betrachtet man die Struktur der Applikation *LandauerWalk* (Abbildung 4.1), dann sieht man, dass vom Startbildschirm direkt die wichtigsten Punkte angesteuert werden können. Dies sind die Vor-Ort-Tour, die Couch-Tour und die Navigation über die Übersichtskarte. Zusätzlich hat der Nutzer über ein aufklappbares Dropdown-Menü Zugang zu allen Inhalten der Anwendung (unter anderem auch Impressum, Datenschutz und weitere Punkte).

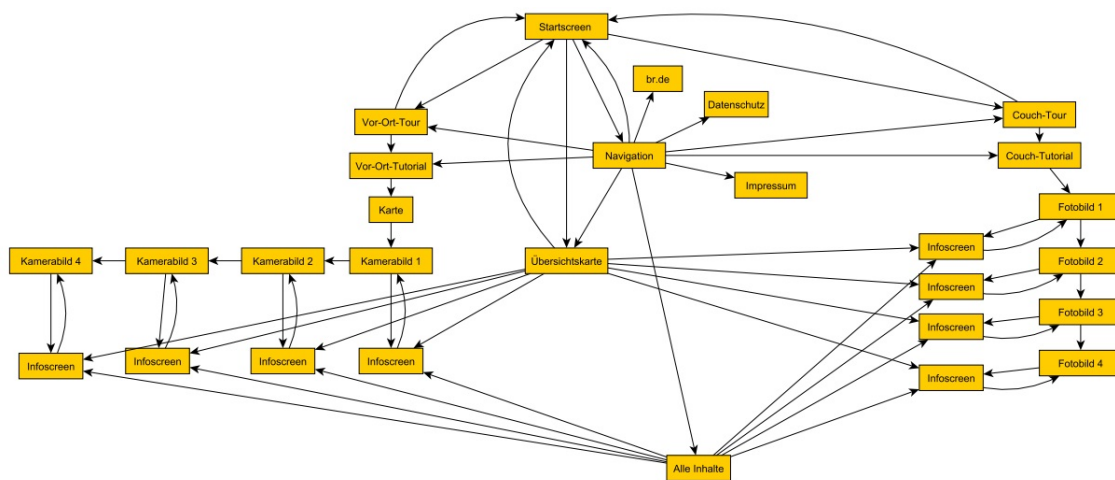


Abbildung 4.1: Menüstruktur der Anwendung *LandauerWalk*.

Wählt der Nutzer entweder die Couch- oder auch die Vor-Ort-Tour, dann gelangt er zum Tutorial. Hier wird beschrieben, welche Funktionen die Applikation bietet und wie diese funktionieren. Das Tutorial besteht aus mehreren Slider-Folien, zwischen denen man beliebig wechseln kann. Auf Wunsch kann man auch das komplette Tutorial überspringen. Nach dem Tutorial gelangt der Nutzer zum Kamerabild oder Fotobild, je nach gewählter Anwendung. Auf diesen Bildern hat der Nutzer mehrere Möglichkeiten sich die Informationen zu beschaffen, die in unterschiedlicher Form vorhanden sind. Wählt man im Startbildschirm die Navigation und anschließend den gewünschten Ort, dann gelangt man ebenso in einen Informationsbildschirm über den jeweiligen Ort und kann von dort aus auf das Foto- beziehungsweise Kamerabild navigieren.

Einen anderen Ansatz verfolgt die Applikation *TimeTraveler*. Hier gelangt man am Start nicht in ein Menü, sondern landet auf einer Karte der Stadt Berlin mit den für die Anwendung wichtigen Punkten als Markierungen. In einer Menüleiste am oberen Rand hat der Nutzer die Punkte Karte, Tutorial und Orte zur Auswahl. Wählt der Nutzer einen Ort in der Liste aus, öffnet sich der Informationsbildschirm zu dem Punkt und es wird die Szene angezeigt, die getrackt werden muss. Wählt der Nutzer hingegen einen Ort auf der Karte aus, kann man die AR-Erfahrung direkt starten.

Stellt man nun die beiden Menüführungen gegenüber, ergeben sich zwei unterschiedliche Ansätze. Zum einen ein Menü mit den Funktionen zur direkten Anwahl im Startbildschirm (klassisch), zum anderen der direkte Sprung in die Navigationsfunktion ohne klassische Menüführung. Aus diesen Erkenntnissen wurden für die in dieser Arbeit entstehende Anwendung zwei GUI-Varianten erstellt.



Abbildung 4.2: Unterschiedliche Versionen der Benutzeroberfläche.

Die erste Variante (Abbildung 4.2 links) eröffnet mit einem klassischen Menü, bei dem alle Funktionen direkt anwählbar sind. Das heißt, der Nutzer kann vom Hauptmenü direkt auf die Live-Tour, Offline-Tour, Navigation, das Tutorial und die Hintergrundinformationen zum Gebäude zugreifen. Die zweite Variante (Abbildung 4.2 rechts) hat dagegen im Startbildschirm ausschließlich die Punkte Live-Tour, Offline-Tour und Hintergrundinformationen zur Auswahl. Hat man eine der Touren ausgewählt, gelangt man in das Tutorial und anschließend in die Navigationsfunktion, bevor die eigentliche Tour startet. Bei beiden Versionen war am unteren Rand der Applikation die Möglichkeit vorhanden, das Impressum und eine FAQ-Seite aufzurufen. Somit steht dem direkten, aber mutmaßlich unübersichtlicheren Menü das einfache aber vermeintlich komplexere Menü gegenüber. In einer ersten Studie wurden diese beiden Versionen von unterschiedlichen Nutzern getestet und bewertet.

4.1.1 Evaluierung

Um die Anwendung möglichst benutzerfreundlich zu gestalten, wurde eine erste Umfrage durchgeführt, die sich ausschließlich auf die Gestaltung der Benutzeroberfläche bezog, während bei einer zweiten Umfrage (siehe Kapitel 5.2) die Funktionen der Anwendung evaluiert wurden. Für jede der beiden Umfragen wurden zuerst die Ziele der Evaluierung gesammelt und aufgelistet.

Die erste Evaluierung wurde so gestaltet, dass zwei Versionen der Umfrage vorhanden sind, wobei die Teilnehmer jeweils nur eine Version erhalten haben. Nach der Umfrage wurden die Ergebnisse miteinander verglichen. Die Versionen besaßen jeweils unterschiedliche Benutzeroberflächen, wie in Kapitel 4.1 beschrieben: das einfache aber vermeintlich komplexere Menü (Version 1, Abbildung 4.2 rechts) und das direkte mit dem mutmaßlich unübersichtlicheren Menü (Version 2, Abbildung 4.2 links). Zum besseren Verständnis wurde bei jeder Evaluation anfangs ein Hinweis vorangestellt, der über die Voraussetzungen der Umfrage informierte und außerdem wurde das Szenario dargestellt. Anhand des Szenarios sollte auch der technisch weniger versierte Nutzer verstehen, welche Funktionen die Anwendung bietet. Die Fragebögen besaßen Vorfragen, die das Vorwissen der Partizipanten herausstellen sollten. Um die Nutzergruppe weiter differenzieren zu können, schloß der Bogen mit demographischen Fragen ab. Ziel der ersten Evaluierung war es die Verständlichkeit und Übersichtlichkeit der Strukturen und die Ästhetik zu bewerten. Außerdem sollten Fehler gefunden und subjektive Meinungen abgegeben werden.

4.1.2 Ergebnis der Evaluierung

Die Umfrage beinhaltete fünf Vorfragen, sieben zu lösende Aufgaben und 28 Hauptfragen, davon drei demographische. Die Fragen bestanden aus Ja-Nein-Fragen, Wissensfragen und Bewertungsfragen. An der Umfrage beteiligten sich 14 Teilnehmer, darunter neun männliche und fünf weibliche (Graph 1, Abbildung 4.3). Alle Probanden, bis auf eine, nutzten Apps auf Ihrem Smartphone oder Tablet und schätzten sich erfahren bei der Benutzung von Apps und insbesondere auch Navigationsapps ein. Über 70% der Partizipanten haben angegeben, dass sie keine Erfahrung mit Augmented Reality beziehungsweise AR-Apps besitzen. Daraus kann man schließen, dass die Teilnehmer im Normalfall mit dem Umgang mit Applikationen keine Probleme haben. Trotz der Unerfahrenheit mit Augmented Reality-Apps waren die Funktionen der Anwendung für 85% der Teilnehmer verständlich.

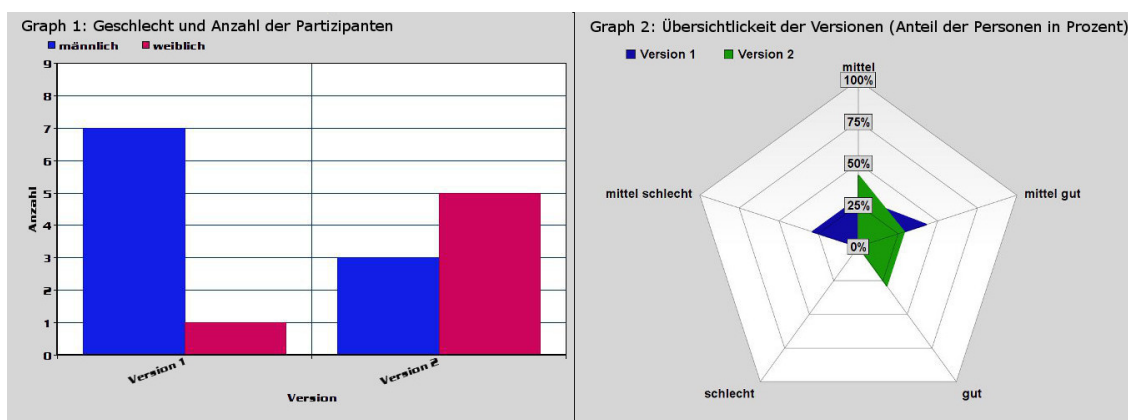


Abbildung 4.3: Ergebnisse der ersten Evaluierung bezüglich Benutzeroberfläche.

Deutliche Unterschiede sind bei der Findung der Funktionen zu sehen. Vor allem die Hauptfunktion, die Navigation und das Tutorial wurden bei Version 2 mit jeweils 86% schnell bis sehr schnell gefunden, die Hintergrundinformationen sogar zu 100% (Graph 4, Abbildung 4.4). Probleme hatten die Nutzer dabei, die Augmented Reality-Funktion zu finden, die nochmal extra

abgefragt wurde, wobei nicht ausgeschlossen werden kann, dass die fehlende Implementierung dieser Funktion das Ergebnis negativ beeinflusst hat. Vor allem, da bei den gleichen Nutzern der Aufruf der Hauptfunktionen wie beschrieben keine Schwierigkeiten mit sich brachte. Um zu den einzelnen Funktionen zu gelangen, wurden in der Anwendung keine zusätzlichen Möglichkeiten gewünscht. Die Übersichtlichkeit der Anwendung wurde von mittel bis sehr gut bewertet, kein Proband beurteilte diese als schlecht (Graph 2, Abbildung 4.3).

Version 1 bot auch für 85% der Teilnehmer verständliche Funktionen, allerdings bleibt anzumerken, dass kein Teilnehmer die Hauptfunktion auf Anhieb gefunden hat. Auch merkten fünf von sieben Teilnehmern an, dass ein Tutorial hilfreich wäre, bei Version 2 wurde dieser Vorschlag nur von drei der sieben Probanden gemacht (Graph 5 und Graph 6, Abbildung 4.4). Selbiges spiegelt sich auch bei der Verständlichkeit der Programmstruktur heraus. Bei Version 1 wurden zudem die erweiterten Funktionen (Navigation, Tutorial) nicht auf Anhieb gefunden, mit Ausnahme der Hintergrundinformationen, die von allen mittel schnell bis schnell gefunden wurden (Graph 3, Abbildung 4.4). Auch haben sich die Teilnehmer weitere Möglichkeiten gewünscht zu den einzelnen Funktionen zu gelangen. Das kann man mit der schlechten Auffindbarkeit der Funktionen in Verbindung bringen. Die Bewertung der Übersichtlichkeit spiegelt auch klar die Präferenz der Teilnehmer wieder. Nur drei von sieben haben diese mit gut, zwei mit mittel und zwei sogar als schlecht bewertet (Graph 2, Abbildung 4.3). Nach Auswertung der Studie ergibt sich für die Anwendung schließlich, dass eine übersichtliche GUI mit direkt ansteuerbaren Funktionen vom Großteil der Nutzer präferiert wird.

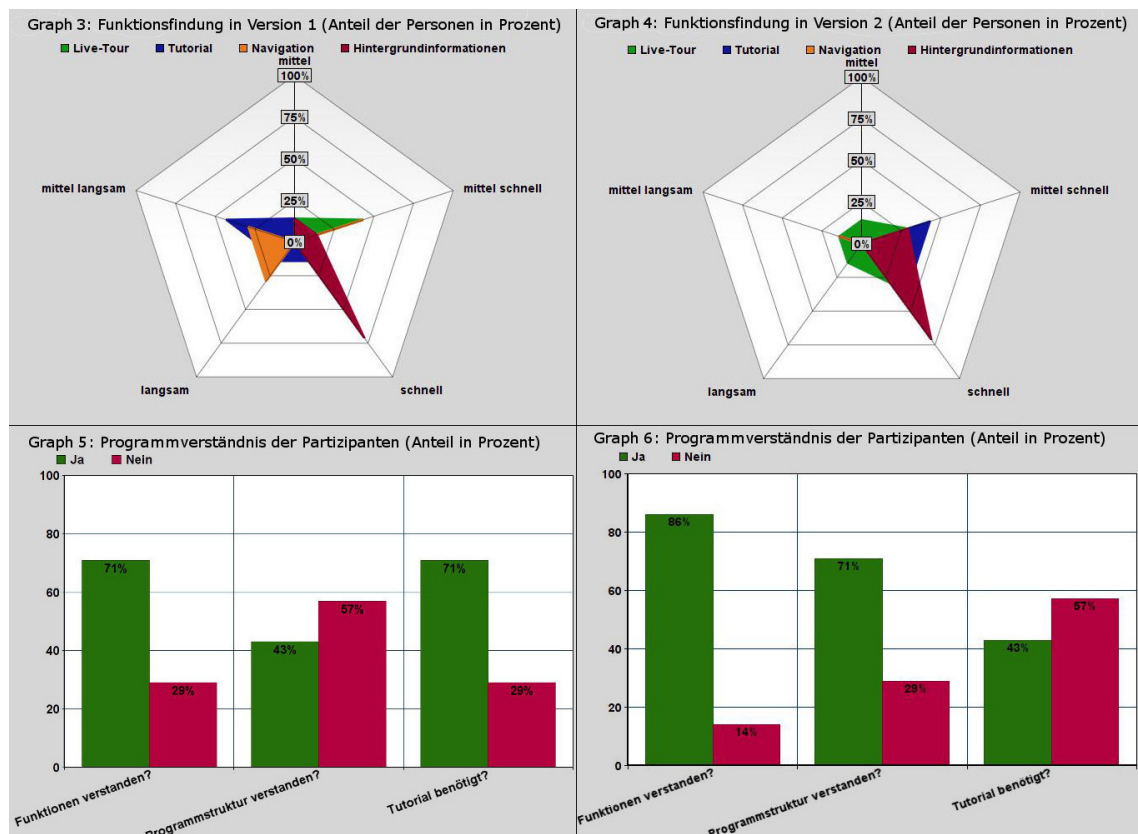


Abbildung 4.4: Ergebnisse der ersten Evaluierung bezüglich Benutzeroberfläche.

Des Weiteren wurden in der Umfrage unterschiedliche Button-Farben miteinander verglichen und auch drei verschiedene Schriftarten sollten bewertet werden. Die Buttons hatten die vier unterschiedlichen Grüntöne #22900F, #0FSD30, #2A902E und #255D33 (siehe Abbildung 4.5), die allesamt zu dem grauen Hintergrund passten. Am wenigsten gefallen hat den Nutzern der Ton

#255D33 mit einer Bewertung von 3,64 in Schulnoten. Am besten abgeschnitten hat der Farbton #22900F mit einer Schulnote von 1,92, wobei die anderen beiden Töne mit 2 und 2,36 bewertet wurden. Als Schriftart wurde den Teilnehmern Times New Roman, Verdana und Arial angeboten. Sowohl ästhetisch, als auch in der Lesbarkeit hat den Partizipanten Verdana mit einer Benotung von 1,36 und 1,79 am meisten zugesagt. Die Schrift mit Serifen Times New Roman hat sowohl in der Leserlichkeit als auch in der Ästhetik am schlechtesten bei den Nutzern abgeschnitten.



Abbildung 4.5: Unterschiedliche Grünfarbtöne für die Buttons der Anwendung.

Zusammenfassend kann man sagen, dass die Teilnehmer der Studie ein direktes Menü mit sofortigem Zugang zu allen Funktionen auf Grund der Übersichtlichkeit und der schnellen Zugänglichkeit bevorzugen. Die große Anzahl der Menüpunkte hat der Übersichtlichkeit keinen Abbruch getan. Die Buttonfarbe sollte ein klares, dunkleres Grün sein, das noch hell genug ist, um sich von dem Grund-Grauton abzuheben. Bei der Schriftart wurde der serifenlose Font Verdana präferiert. Die daraus resultierende finale GUI des Prototyps wird in Kapitel 5 vorgestellt.

4.2 Modellanpassung

Nachdem das Aussehen der Anwendung bestimmt wurde, sollten im nächsten Schritt die Funktionen des Augmented Reality Tourist Guides eingebaut werden. Im ersten Schritt sollte ein passendes 3D-Modell für die Darstellung erstellt werden. Die *Technische Universität Darmstadt* stellte dafür ein detailreiches 3D-Modell der *Hauptsynagoge München* zur Verfügung. Dieses Modell wurde im Rahmen eines Projekts angefertigt mit der Motivation, die zerstörten Synagogen in Deutschland virtuell zu rekonstruieren. Seit dem Jahr 1995 werden im Fachgebiet CAD der Architektur unter Leitung von Prof. Manfred Koob und Dipl. Ing. Marc Grellert die Synagogen nachgebildet, die 1938 von den Nationalsozialisten zerstört wurden, mit dem Ziel den kulturellen Verlust aufzuzeigen. [24]

Das originale Modell der *TU Darmstadt* hat über 415.000 Polygone und war in einer groben Modellierung der umgebenden städtischen Situation eingebunden (siehe Abbildung 4.6). Mit einer Dateigröße von 110 Megabytes ist eine direkte Integration des Modells in ein echtzeitbasiertes, mobiles AR-System wie *Junaio*, sowie auch das Erstellungstool *Metaio Creator* aus Performanzgründen nicht möglich. Aufgrund dessen und auch aufgrund der verschiedenen Betrachtungspositionen wurde entschlossen, ein neues, eigenes Modell nach dem Vorbild der Darmstädter Synagoge anzufertigen. Dieses Modell soll einerseits eine gute Rendering-Performanz ermöglichen, andererseits möglichst viel unnötigen Datentransfer vermeiden. Dies wird unter Anderem dadurch realisiert, dass an den Stellen, die in der Anwendung nicht sichtbar sind, keine detailreiche Modellierung stattfindet. Zusätzlich hält eine niedrige Detailstufe in der Texturierung die Dateigröße des 3D-Modells möglichst gering. Da die Internetverbindung grundsätzlich mit mobilen Daten stattfindet, sollen durch die richtige Modellgröße der Datentransfer und somit die Kosten für den Nutzer möglichst gering gehalten werden. Das finale texturierte Modell hat mit rund 12.000 Polygonen

eine Größe von circa 1,2 Megabyte und ist in einer Tag- und in einer Nachtversion enthalten. Das heißt, es wurden jeweils unterschiedlich beleuchtete Modelle angefertigt, die entsprechend ihrer Nutzung angepasste Texturen besitzen. Die finalen Synagogen-Modelle werden in Kapitel 5 vorgestellt.



Abbildung 4.6: Modell der Alten Hauptsynagoge München der TU Darmstadt. [24]

4.3 Markerloses Tracking

Wie schon in Kapitel 2.1.4 beschrieben, bietet es sich für das Tracking der realen, dreidimensionalen Szene vor Ort an, eines der bildbasierten Verfahren zu nutzen. Die sensorbasierten Verfahren sind aufgrund der Ungenauigkeit nicht geeignet, da die Platzierung der Synagoge im städtischen Raum exakt erfolgen muss, um beim Nutzer einen möglichst immersiven Eindruck zu erzielen. Eine ungenaue Platzierung würde diesen aus der Illusion reißen und die Anwendung, beziehungsweise der Einsatz der AR-Technologie, würden viel vom gewünschten Effekt oder sogar ihren Sinn verfehlen.

Im Sortiment der bildbasierten Trackingverfahren befinden sich im *Metaio Creator* grundsätzlich Marker-, Bild- und Objekttracking. Wie bereits in Kapitel 2.1.4 beschrieben, eignen sich das Markertracking und das Bildtracking nur bedingt für die Realisierung. Anders als in den erwähnten Augmented City Guides soll hier nicht nur eine 3D-Umgebung erkannt und anschließend ein 2D-Objekt darauf projiziert werden, sondern ein dreidimensionales, virtuelles Objekt in Form eines 3D-Modells. Zur besseren Bilderkennung, aber vor allem für die richtige Platzierung eignen sich somit das Tracking mit Hilfe einer 3D-Map oder eines 3D CAD Modells. Bestätigt wurden diese Erkenntnisse auch durch erste Feldversuche und Tests. Das Tracking mittels eines 2D-Bildes kann zu den optimalen Verhältnissen, bei gleichen Tageszeiten, Lichtverhältnissen und Wetterverhältnissen funktionieren, allerdings fehlt dieser Methode die Flexibilität und sie ist somit zu instabil. Zusätzlich ist es schwer realisierbar, das 3D-Modell gut in die Szene zu positionieren, da hierfür im 2D-Raum (im Gegensatz zum 3D-Raum) weniger Orientierungspunkte für Platzierung und Ausrichtung des dreidimensionalen Modells sind.

Zur Umsetzung des Trackings mittels 3D-Map ist somit eine solche 3D-Punktwolke erforderlich. *Metaio* stellt zur Schaffung solch einer Map die *Metaio Toolbox-App* für Smartphones und Tablets zur Verfügung. Mit Hilfe dieses Tools nimmt man die Umgebung auf (“recording“ oder auch “teaching“), die später getrackt werden soll, und erhält eine 3D-Punktwolke, die die Szene darstellt. Diese roten Punkte repräsentieren besonders markante Punkte innerhalb der Szene,

die später für die Erkennung dienen sollen. Die mit der Toolbox erstellte Map kann zur weiteren Bearbeitung in den *Metaio Creator* laden.

Beim Versuch, die 3D-Map am Gebäude des Oberpollingers in München zu erstellen, sind allerdings einige Schwierigkeiten aufgetreten. Für die Naherfahrung von Position 1 (siehe Abbildung 3.1) wird für die Bilderkennung vor allem die Fassade des Gebäudes berücksichtigt. Mit Ausnahme zweier großer Fenster und dem Fassadenschriftzug "Oberpollinger" bietet die Fassade relativ wenige Merkmale beziehungsweise viele eintönige, beige Flächen und sich wiederholende Muster. Hinzu kommen Spiegelungen innerhalb der großen Fenster, die das 3D-Objektracking negativ beeinflussen. Aus diesem Grund ist die erstellte 3D-Punkt看ke nur wenig repräsentativ und liefert kein gutes Ergebnis (siehe links in Abbildung 4.7). Auch bei variiertem Belichtung erhält man keine ausreichend gute 3D-Map für ein stabiles und robustes Tracking. Als Notlösung wurde die bestmögliche 3D-Map genommen und alle unnötigen und fehlplatzierten Punkte gelöscht. Somit erhält man eine reduzierte Punkt看ke, die zumindest bei optimalen Verhältnissen vor Ort zu einem ausreichend funktionsfähigen Tracking führt.

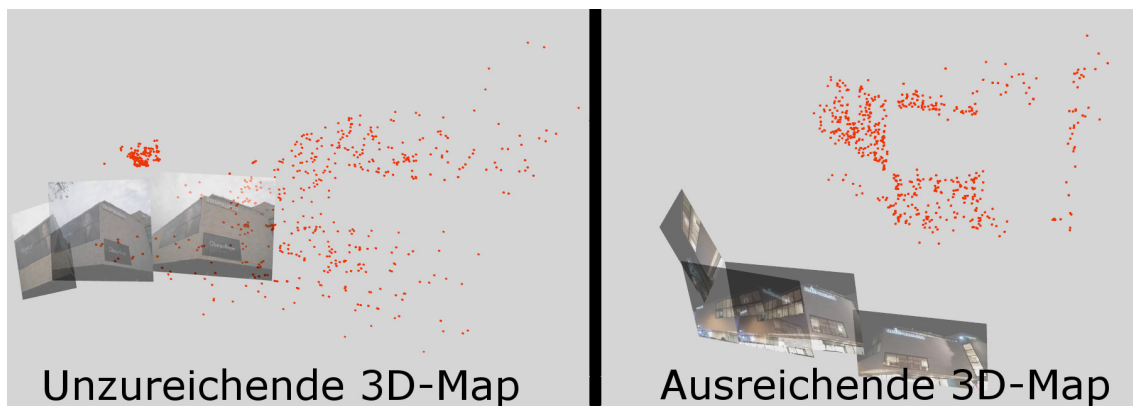


Abbildung 4.7: Fehlerhafte und nutzbare 3D-Map des Oberpollingers.

Für die Fernerfahrung von Position 2 (siehe Abbildung 3.1) gilt Obiges nicht. Hier kann man eine 3D-Punkt看ke in ausreichender Qualität erstellen, die auch gute Ergebnisse bei der Initialisierung zur Bilderkennung liefert. Da die Punkte zu unterschiedlichen Tageszeiten in ihrer Farbe variieren, ist es grundsätzlich nicht möglich, die tagsüber erstellten Konfigurationen auch nachts zu nutzen. Somit müssen für beide Positionen weitere Varianten der 3D-Map angefertigt werden. Beim Testen dieser Varianten wurde festgestellt, dass sich vor allem das Ergebnis der Naherfahrung nachts deutlich verbessert. Dies kann man damit erklären, dass nachts durch Beleuchtungen am Gebäude markantere Punkte gesammelt werden. Diese Punkte variieren auch nicht in der Darstellung, da sich die Lichtfarbe nicht ändert. Aufgrund der Dunkelheit entstehen auch weniger Reflexionen, die die Positionierung der erkannten Punkte verfälschen. Die für das Tracking geeignete 3D-Map sieht man in Abbildung 4.7 auf der rechten Seite. Gut zu sehen ist auch, dass bei der Punkt看ke die Form der Fassade aufgrund der richtigen Positionierung der Punkte zu erkennen ist. Die Robustheit dieses Trackingverfahrens wird in weiteren Tests während der Evaluation des finalen Prototyps (siehe Kapitel 5.2) untersucht.

Da aber vor allem bei der Naherfahrung das genannte Bildererkennungsverfahren weniger robust arbeitet, findet zusätzlich ein zweites Verfahren zur Gegenüberstellung und Vergleichbarkeit Anwendung. Mit der Nutzung eines 3D CAD Modells (Kantenmodell) verspricht man sich, die Einschränkungen, die bei der 3D-Map auftreten, umgehen zu können. Im Gegensatz zu obigem Verfahren, nutzt dieses Verfahren die Erkennung von Kanten eines nachmodellierten Objekts. Diese Kantenerkennung sollte weniger von Lichtverhältnissen und kleineren Positionsveränderungen beeinflusst werden. Zusätzlich wird die Positionierung der Synagoge durch das Nachmodellieren des vorhandenen Objekts vereinfacht.

Somit wurde, wie in Abbildung 4.8 zu sehen ist, ein maßstabsgetreues Modell der Fassade angefertigt, bei dem die Ober- und Unterkanten der Fassade deutlich zu erkennen sind. Zusätzlich sind auch die Fenster mit ihren markanten Ecken im Modell zur Bilderkennung enthalten. Dieses Modell muss im optimalen Fall nur einmal für alle Positionen und Tageszeiten angefertigt werden, da sich bei Veränderung der Position ausschließlich der Blickwinkel auf das Gebäude und seine Kanten ändert, nicht aber die Kanten ansich in Ihrer Form und Darstellung. Im Rahmen von ersten Tests hat die Initialisierung mittels dieses Verfahrens im Nahbereich zu deutlich besseren Ergebnissen geführt, allerdings weiterhin mit Problemen, vor allem beim weiteren Tracking, sobald die virtuelle Synagoge das erste Mal angezeigt wurde. Bei Bewegungen des Nutzers beziehungsweise des Smartphones oder Tablets verhält sich die Position des virtuellen Modells relativ instabil und das Modell wird oft nicht konstant in der korrekten Perspektive angezeigt. Des Weiteren werden bei der Bilderkennung aus der Ferne zu viele andere Objekte als Oberpollinger erkannt. Diese Fehler sollten allerdings nach ausführlichen Tests mit Hilfe von Konfigurationsoptimierungen ausgebessert werden. Auch hier wird die Robustheit dieses Trackingverfahrens in weiteren Tests während der Evaluation des finalen Prototyps in Kapitel 5.2 untersucht.

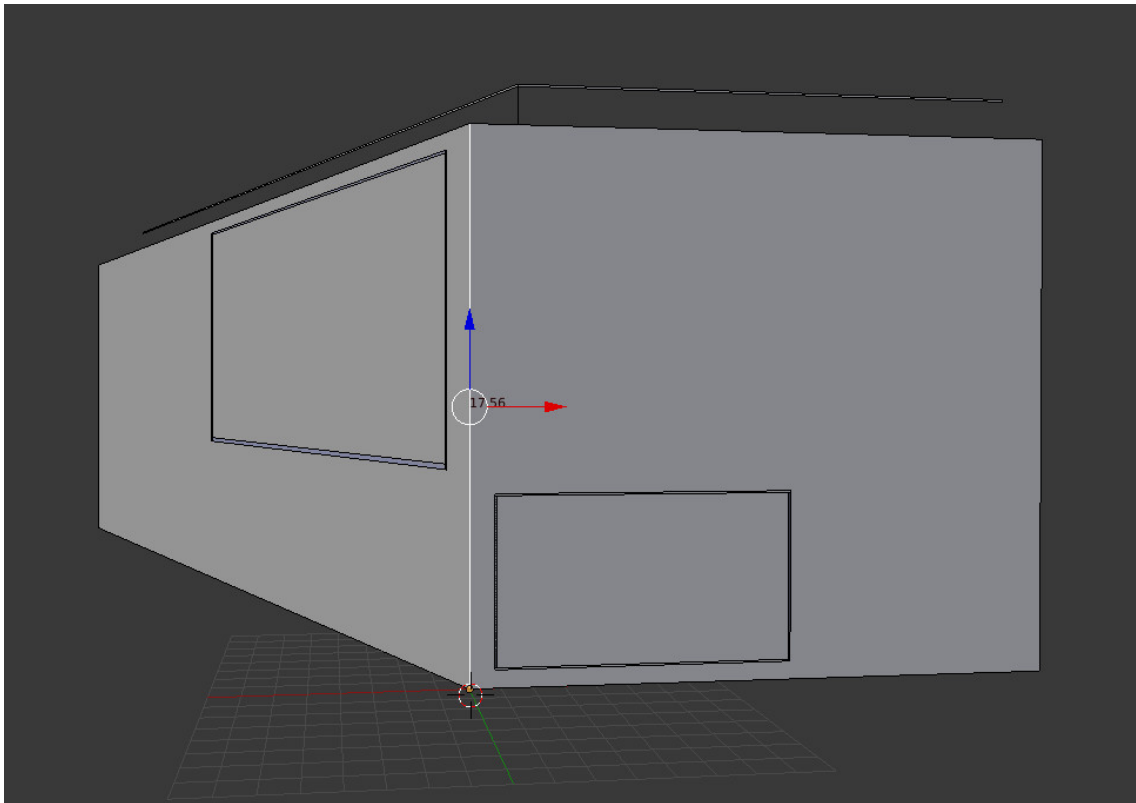


Abbildung 4.8: Maßstabsgetreues 3D CAD Modell der Fassade des Oberpollingers in München.

4.4 Einpassen der Geometrie

Nachdem die Initialisierung der Position mittels des geeigneten Trackingverfahrens funktioniert und die Synagoge modelliert ist, muss als nächstes beides zusammengeführt werden, damit das 3D-Modell nach dem Tracking an die richtige Stelle positioniert wird. Dabei ist darauf zu achten, dass das Modell exakt platziert wird, damit dem Nutzer später nicht die Illusion genommen wird, das damalige Gebäude zu betrachten. Anders als in einigen der bereits vorgestellten Augmented City Guides aus Kapitel 2.3 wird anhand des bildbasierten Trackingverfahrens die Position der Synagoge bestimmt. Die Platzierung findet also komplett ohne Sensoren statt. Wie auch schon bei

der Bilderkennung muss aber auch hier das Modell den Umgebungskriterien angepasst werden. Somit soll im Vergleich zu den anderen Augmented City Guides eine realistischere Immersion stattfinden. Das heißt, die Lichtverhältnisse und die Verdeckung beziehungsweise Transparenz spielen für die authentische Platzierung eine wichtige Rolle.

4.4.1 Berücksichtigung der Lichtverhältnisse

Bei der Berücksichtigung der Lichtverhältnisse sind die Tageszeit und die städtebauliche Situation wichtige Faktoren, auf die geachtet werden muss. Um eine Desillusionierung wie bei den vorgestellten Augmented City Guides zu bekommen, werden für unterschiedliche Tageszeiten unterschiedliche Modelle in die Szene geladen. Je nach Uhrzeit soll somit eine andere Textur für das 3D-Modell geladen werden. Das bedeutet konkret, dass es wie in Abbildung 4.9 ein Modell für Tageslicht und ein Modell für die Dunkelheit bei Nacht gibt. Bei der Nachtversion wurden die Lichtquellen abgedunkelt und es muss weniger auf externe Lichtquellen geachtet werden, die das Modell stark anleuchten könnten. Tagsüber hingegen spielen zusätzlich auch die Wetterverhältnisse eine entscheidende Rolle.

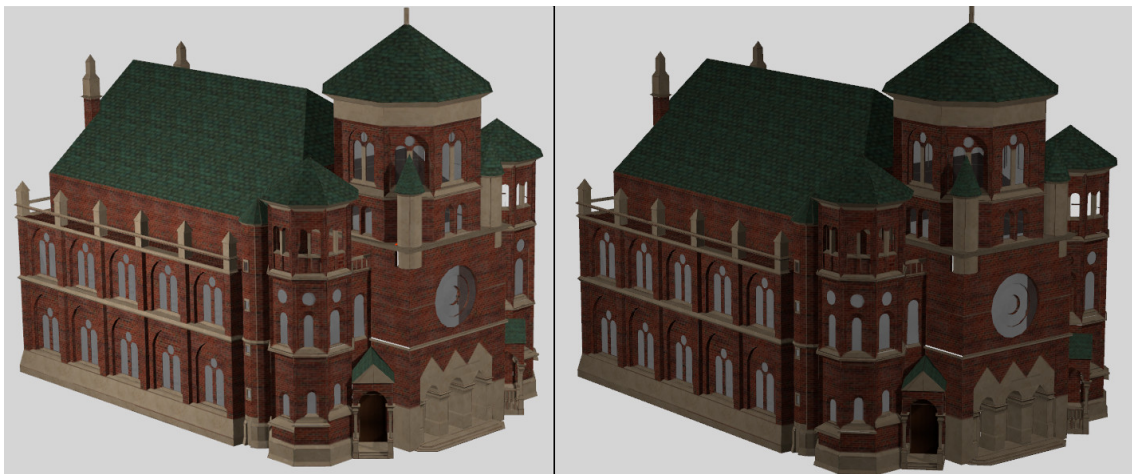


Abbildung 4.9: Zwei unterschiedlich texturierte Gebäudemodelle, jeweils eins für tagsüber (links) und nachts (rechts).

Aus Performanzgründen wurde vorerst nur eine hell beleuchtete Version erstellt. Es wurde bei ersten Feldtests beobachtet, dass die reale Szene und das Oberpollinger-Gebäude zu fast allen Jahreszeiten sehr hell erscheint und nur bei starkem Regenschauer wirklich trist und dunkel wirkt. Somit wurde eine freundliche, helle Version angefertigt, die hauptsächlich für gutes Wetter geeignet ist, da auch davon ausgegangen werden kann, dass die Synagoge weitestgehend bei gutem Wetter betrachtet wird. Für nachfolgende Arbeiten erscheint es als sinnvoll, weitere Differenzierungen bei der Texturierung anzufertigen, um dem Nutzer ein noch besseres Erlebnis zu ermöglichen.

4.4.2 Verdeckung und Transparenz

Ein qualitativ hochwertiges Erlebnis der Hauptsynagoge hängt aber nicht nur von der korrekten Darstellung der Lichtverhältnisse, sondern auch stark von der Verdeckung und Transparenzeinstellung des digitalen Modells ab. Sollten reale Objekte die Szene nicht korrekt verdecken oder von der digitalen Synagoge verdeckt werden, wird der Nutzer wiederum aus der Illusion gerissen, das Modell repräsentiere die Darstellung der *Alten Hauptsynagoge*. In den genannten Beispielen von Augmented Reality Tourist Guides wurde dieses Problem nicht bearbeitet. Hier fand eine

komplette Verdeckung der realen Szene an der Stelle statt, an der die digitalen Informationen geladen wurden. Der *Metaio Creator* bietet allerdings die Möglichkeit, das dargestellte Objekt, in diesem Fall die Hauptsynagoge, an den richtigen Stellen zu verdecken.

Zu diesem Zweck wurde ein 3D-Modell erstellt, das die Silhouette des Künstlerhauses, von Position 2 (siehe Abbildung 3.1) aus gesehen, darstellen soll. Wie Abbildung 4.10 zeigt, muss das Modell genauso wie das Modell der *Alten Hauptsynagoge* in die Szenerie im Creator eingefügt werden. Für die letztendliche Anwendung bedeutet dies, dass alles was sich in der realen Szene hinter dem virtuellen Verdeckungsmodell befindet, nicht zu sehen ist.

Bei der Erstellung des Modells sind die für die Verdeckung wichtigsten Konturen eingefügt worden. Diese Konturen betreffen aber nur die Ecke Maxburgstraße und Herzog-Max-Straße, da nur hier die Silhouette des Künstlerhauses heraussticht. Weiterhin wurde das Verdeckungsmodell bis zu Position 2 als Quader modelliert, um sicherzustellen, dass bei Bewegung konstant die Synagoge verdeckt bleibt. Im Idealfall sollte aufgrund des Quaders die Synagoge auch dann verschwinden, wenn man sich hinter dem Künstlerhaus aufhält. Da an dieser Stelle allerdings kein Tracking mehr greift, dient dies nur als Sicherheitsmaßnahme.



Abbildung 4.10: Einbettung eines Verdeckungsmodells, damit die Hauptsynagoge hinter dem Künstlerhaus positioniert erscheint.

Die optimale Illusion eines realen Gebäudes würde bestehen, wenn Objekte wie parkende Fahrzeuge oder sogar sich bewegende Personen ebenfalls die Synagoge bedecken würden. Allerdings ist es einerseits technisch nicht umsetzbar, parkende Autos die Hauptsynagoge verdecken zu lassen, da im Vorhinein nicht erahnt werden kann, wo welche Fahrzeuge geparkt sind. Auch Menschen, die sich vor der Synagoge bewegen, oder vorbeifahrende Fahrzeuge und Fahrräder sind aufgrund der Unregelmäßigkeit und Unvorhersehbarkeit nicht als verdeckende Objekte zu erstellen. Somit kann zum derzeitigen Stand die Illusion eines realen Gebäudes nur teilweise durch Verdeckung gelöst werden. Für die Verdeckung von bewegenden Objekten wäre ein System notwendig, das automatische und dynamische 3D-Rekonstruktion in Echtzeit bewerkstelligen kann. Erste Ansätze solcher Systeme sind bereits vorhanden, allerdings ist der reale Einsatz (vor allem auf mobilen Endgeräten) noch nicht möglich [21].

4.5 Offline-Tour

Neben der Live-Tour soll der *HistoryTraveller* außerdem eine Offline-Funktion zur Verfügung stellen. Wie auch im *TimeTraveler* oder beim *LandauerWalk* soll diese Funktion dem Nutzer die Möglichkeit bieten, ein Alternativerlebnis zu bekommen, sollte sich dieser nicht vor Ort aufhalten können. Sollte das Tracking aus undefinierbaren Gründen nicht greifen, so ist es zusätzlich möglich die Offline-Tour als Notlösung einzuschalten. Auch für ausländische Nutzer, die aufgrund des Datenverbrauchs Roaming-Gebühren zahlen müssten, wäre dies eine Alternativlösung, um Kosten

zu sparen.

Aufgrund der unveränderlichen Vorgaben durch *Junaio* ist es derzeit noch nicht möglich, die Offline-Tour komplett ohne Internet zu nutzen. Dies sollte allerdings bei einer Weiterentwicklung implementiert werden. Die Offline-Tour bietet dem Nutzer drei Positionen zur Betrachtung der *Alten Hauptsynagoge* an. Wie in Abbildung 4.11 zu sehen ist, kann mittels eines Radiobuttons zwischen den Positionen gewechselt werden. Gleichzeitig ist es möglich, die Sichtbarkeit des virtuellen Objekts zu variieren. Somit soll der Nutzer die Einbettung ins städtische Szenario beziehungsweise den Wandel von damals zu heute besser nachvollziehen können. Aufgrund der technischen Möglichkeiten und aus Performanzgründen wurde jedoch kein 3D-Modell in die Offline-Tour integriert. Hier wird wie auch schon bei den vorgestellten Augmented City Guides ausschließlich mit Bildern gearbeitet.

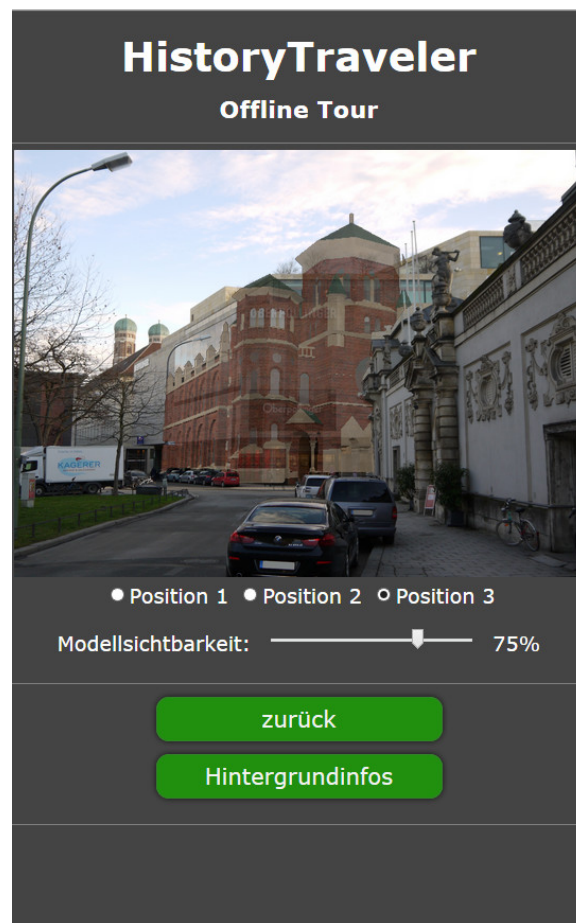


Abbildung 4.11: Die Offline-Tour des Prototypen *HistoryTraveller*.

4.6 Navigation zum Ziel

Nachdem die Hauptfunktionen realisiert wurden, fehlen noch die Zusatzpunkte, die einen Tourist Guide ausmachen. Von einem Tourist Guide erwartet man die Navigation an das Zielobjekt, damit Nutzer, die sich vor Ort nicht auskennen, auch die Möglichkeit haben den Ort zu finden, den sie sehen wollen. Auf lange Sicht gesehen soll der Tourist Guide mehrere Stationen anbieten, an denen man die zerstörten Gebäude wieder als Augmented Reality-Erfahrung erleben kann. In diesem Fall würde der Tourist Guide als Stadtführer dienen. Aus diesem Grund wurden zwei Navigationskarten in die Applikation eingebaut. Die erste Karte dient der Navigation durch München an die Stellen, an denen Gebäude mit der Augmented Reality Erfahrung betrachtet werden können,

die zweite soll spezifisch an die wichtigen Positionen zur Initialisierung des AR-Erlebnisses leiten. Wie bei den bereits vorgestellten Applikationen dienen die Karten nicht nur zur Übersicht, sondern können auch zum Starten einer Routenplanung via Navigationsapplikation verwendet werden.

4.7 Hintergrundinformationen

Die Hintergrundinformationen zu den Gebäuden sind ein wichtiger Bestandteil der Anwendung, da durch den Informationsgehalt der Mehrwert des Tourist Guides gewährleistet wird. Da sich die Arbeit auf die Hauptsynagoge beschränkt, wurden auch nur die Informationen zu dem Gebetshaus in die Anwendung integriert. Die Hintergrundinformationen sind über einen separaten Menüpunkt anwählbar oder während der Live- beziehungsweise Offline-Tour, wenn man die Hauptsynagoge betrachtet.

Die Informationen werden in einer Slider-Ansicht mit drei Folien angezeigt. Die erste Folie beinhaltet allgemeine Informationen zur Hauptsynagoge. Hier erfährt der Nutzer, wo sich die Synagoge befand, wann sie entstand, seine Besonderheiten und letztendlich warum sie zerstört wurde. Die zweite Folie informiert über die Architektur des Gebäudes und was es so besonders machte und welche Merkmale bei der Betrachtung erkennbar sind. Auf der dritten Folie wird die Baugeschichte der *Alten Hauptsynagoge* beschrieben und wie das imposante Bauwerk entstand.

5 Ergebnis der Entwicklung

In diesem Kapitel soll nun der Prototyp der Anwendung vorgestellt werden. Dieser wurde in einer weiteren Nutzerstudie evaluiert und anschließend um weitere Punkte verbessert. Nach der Beschreibung der Durchführung der Evaluation werden die Ergebnisse dargestellt.

5.1 Finaler Prototyp

Wie bereits in Kapitel 4 erläutert, beinhaltet der finale Prototyp fünf Funktionen, davon zwei Hauptfunktionen. Die Hauptfunktionen der Anwendung sind zum einen die Live-Tour, mit der man die Hauptsynagoge vor Ort mit Augmented Reality erleben kann, zum anderen die Offline-Tour, bei der man ein Alternativ-Erlebnis erhält, sollte man sich nicht vor Ort aufhalten können oder die App nur ohne Internetverbindung nutzen können. Des Weiteren sind in der Anwendung die typischen Funktionen eines Tourist Guides enthalten; dazu gehören eine Navigationsfunktion und eine Informationsfunktion. Zuletzt hat man noch die Möglichkeit, sich direkt über den Punkt Hintergrundinformationen über die Hauptsynagoge zu informieren. Zum erleichterten Einstieg für unerfahrene Nutzer ist außerdem eine Hilfestellung in Form eines Tutorials integriert. Zusatzinformationen über die Applikation, verwendete Begriffe oder die Entwickler findet man unter der FAQ-Seite oder dem Impressum am unteren Rand des Hauptmenüs.

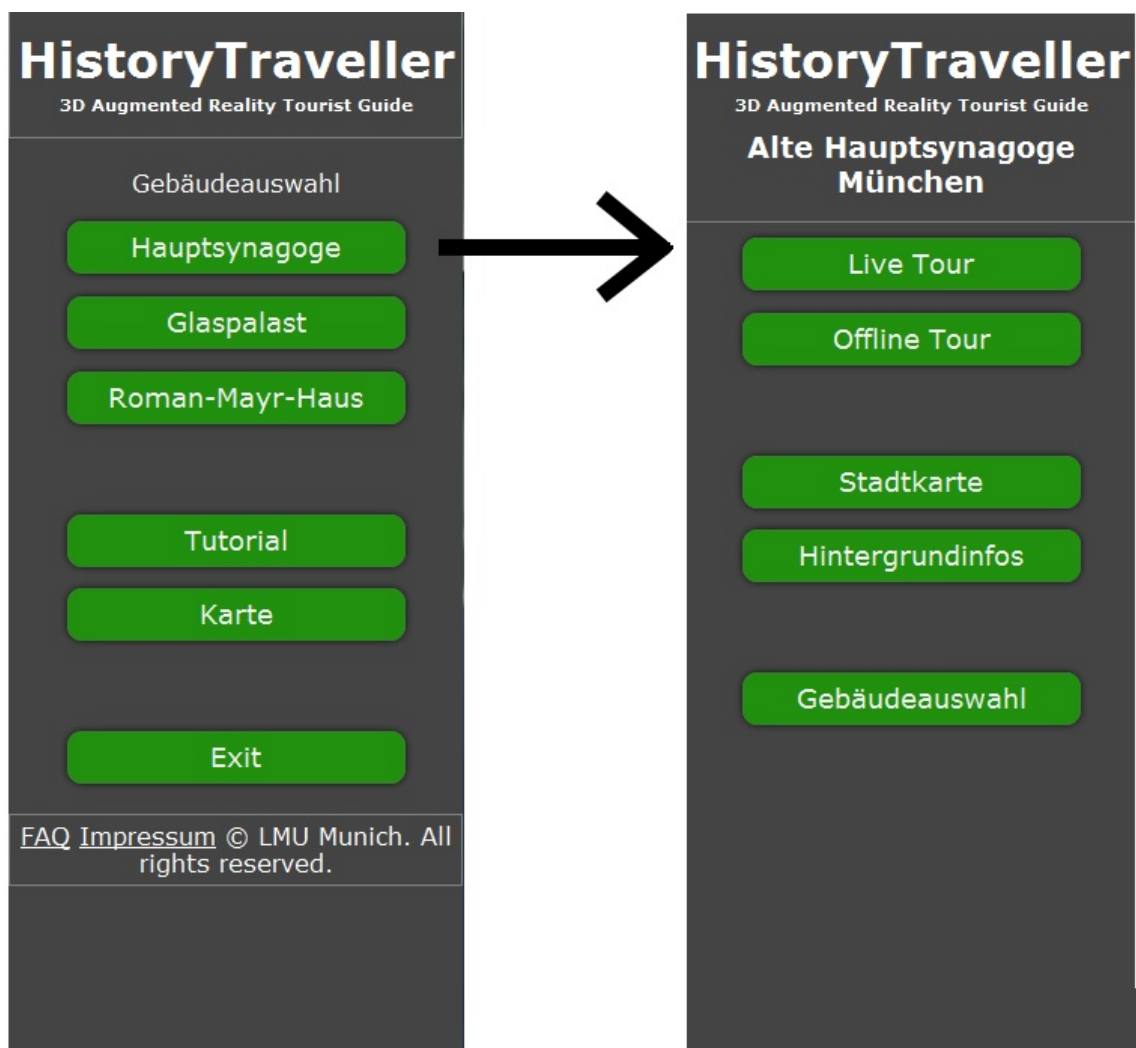


Abbildung 5.1: Finales Hauptmenü der Anwendung *HistoryTraveller* mit anschließendem Menü.

Wie in Abbildung 5.1 zu sehen ist, gelangt man im fertigen Prototyp nach dem Starten der Anwendung ins Hauptmenü, in dem man aus einem der verschiedenen, besuchbaren Gebäude auswählt. Außerdem hat man hier Zugang zum Tutorial, um die Funktionen der Anwendung erklärt zu bekommen, oder zur Karte, um dort zu den Standorten der Gebäude navigiert zu werden. Hat man sich für ein Gebäude entschieden, in dieser Arbeit ist ausschließlich die Hauptsynagoge implementiert, dann bekommt man die Auswahl sich für die Live-Tour oder die Offline-Tour zu entscheiden. Alternativ wählt man den Punkt Stadtkarte, um sich zur Synagoge und dort zu den richtigen Positionen navigieren zu lassen, oder man informiert sich direkt über die Synagoge, wenn man den Punkt "Hintergrundinfos" auswählt.

Nach dem Starten der Live-Tour startet die Kamera. Mit ihrer Hilfe kann man die Szene tracken und das 3D-Modell der Synagoge erleben. Hier kann man ebenfalls zu den Hintergrundinformationen oder zurück zum Menü gelangen. Man muss bei der Live-Tour die Kamera mittig auf das Gebäude des Oberpollingers richten und erhält je nach Tageszeit das 3D-Modell in der entsprechenden Form (siehe Abbildung 5.2). Zusätzlich ist es weiterhin möglich, dass sich der Nutzer um die Synagoge herum bewegt. Bei der Naherfahrung reicht hier das Spektrum von der Vorderseite bis hin zur kompletten Seitenansicht.

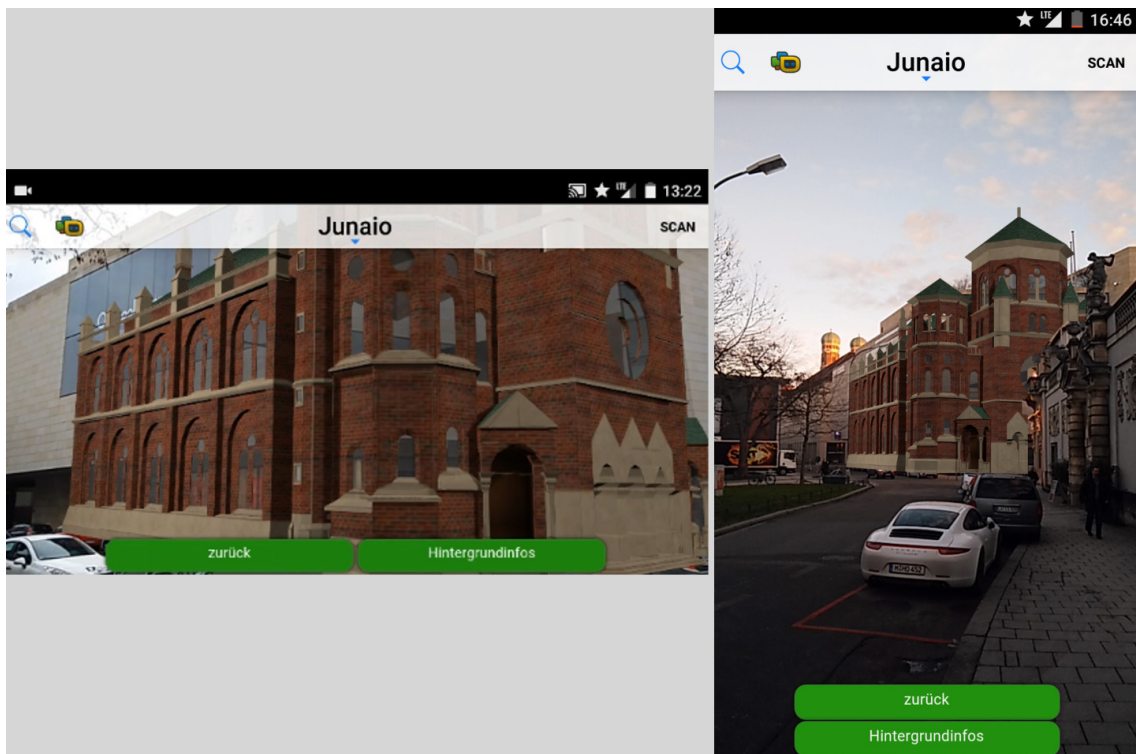


Abbildung 5.2: Live-Tour im finalen Prototypen. Links in der Naherfahrung und rechts in der Fernerfahrung.

Alternativ oder auch wenn man sich nicht vor Ort aufhalten kann, gelangt man über den Punkt Offline-Tour zu ebendieser. Die dortigen Funktionen sind wie bereits in Kapitel 4.5 beschrieben zum einen die Veranschaulichung der Synagoge im städtischen Raum auf einem Bild von drei wählbaren verschiedenen Positionen. Andererseits kann man die Modellsichtbarkeit mit einem Schieberegler verändern. Zuletzt sind auch von hier die Hintergrundinformationen aufrufbar.

5.2 Evaluierung finaler Prototyp

Diese Version des Prototypen wurde in einer zweiten Studie weiteren Tests unterzogen. Diese Tests hatten zum Ziel, die Verbesserung der Benutzeroberfläche nach der ersten Studie zu evaluieren.

Außerdem sollten die beiden implementierten Trackingverfahren, mittels 3D-Map und mittels 3D CAD Modell, gegenübergestellt werden.

Die Evaluierung fand direkt vor Ort am Oberpollinger statt. Es nahmen insgesamt 16 Personen daran teil. Jeweils zwei Teilnehmer bekamen eine Version, in der sowohl für das nahe Tracking als auch das ferne Tracking nur ein Verfahren genutzt wurde, entweder die Punktwolke oder das Kantenmodell. Die restlichen zwölf Partizipanten erhielten jeweils einen Mix aus Kantentracking bei der Naherfahrung und 3D-Map-Tracking bei der Fernerfahrung, da diese Kombination bei den Vorab-Tests das beste Ergebnis geliefert hatte. Getestet wurde sowohl tagsüber als auch nachts und auch bei unterschiedlichsten Wetterverhältnissen, sprich bei Sonnenschein, Regen oder Schneefall.

Jeweils acht Frauen und acht Männer im Alter von 21 bis 55 Jahren hatten zehn Aufgaben zu lösen und fünf Vor- und 31 Hauptfragen zu beantworten. Die Vorfragen betrafen das Smartphone-Nutzungsverhalten der Nutzer und die Vorkenntnis von Augmented Reality. Die Hauptfragen unterteilten sich in die Kategorien allgemeine Fragen, Fragen zur Live-Tour, Fragen zur Offline-Tour, Fragen zu weiteren Funktionen der App, Fragen zur Applikation allgemein und letztendlich demographische Fragen. Bei den Aufgaben galt es, die Anwendung allgemein zu bedienen und die unterschiedlichen Funktionen auszuprobieren. Hier lag der Fokus auf den Hauptfunktionen Live-Tour und Offline-Tour, allerdings sollten auch die Benutzeroberfläche und Verbesserungen nach der ersten Evaluierung hinterfragt werden.

Zum Testen der Live-Tour galt es jeweils von Position 1 (Naherfahrung) und Position 2 (Fernerfahrung) die Anwendung das Oberpollinger-Gebäude erkennen zu lassen. Nicht nur die Initialisierung selbst sollte beobachtet werden, sondern auch das projizierte Modell bei Bewegung der Kamera. Die Nutzer konnten sich mehrere Meter von der Initialisierungsposition weg bewegen und die Nachverfolgung des Trackings beobachten. Zuletzt galt es bei beiden Positionen die Reinitialisierung zu testen. Das heißt, dass die Nutzer mehrfach untersuchen sollten, wie robust die Initialisierung der Szenerie funktioniert, wenn die Kamera auf etwas anderes gerichtet wurde.

Zum Testen der Offline-Tour wurden den Partizipanten freie Bedienungsmöglichkeiten gegeben. Hier war die Aufgabenstellung ausschließlich, die Funktionen frei zu testen. Somit sollten Interaktionsvorgaben vermieden und mögliche Fehler durch individuelle und variierende Bedienungsführungen besser gefunden werden.

5.3 Ergebnis der Evaluierung

Anhand der Antworten auf die Vorfragen ist zu erkennen, dass ein breites Nutzerspektrum die Studie durchführte. Die eigene Erfahrung mit Smartphone-Apps und auch Navigationsapplikationen wurde überwiegend als viel, aber auch als mittel und wenig eingeschätzt. Bei der Erfahrung mit Augmented Reality allgemein gaben rund 31% an, gar keine Erfahrung zu besitzen, bei AR-Apps waren es 44%. Trotz allem gaben 100% der Teilnehmer an, die Funktionen der Anwendung verstanden zu haben. Im Vergleich zur ersten Umfrage (siehe Kapitel 4.1.1) konnten einige Fortschritte in Bezug auf die Übersichtlichkeit und Bedienung der App erzielt werden. So wurde die Programmstruktur dieses Mal zu 100% als verständlich deklariert. In der vorherigen Evaluierung waren es nur 43% (Version 1) beziehungsweise 71%. Die Übersichtlichkeit wurde von 87,5% mit gut bis sehr gut bewertet. Das ist eine Steigerung von rund 30% im Vergleich zur besser bewerteten Version 2 aus der ersten Umfrage. Dies bestätigt sich auch dadurch, dass 94% der Teilnehmer jederzeit wussten, wo sie sich im Programm befanden. Zusätzlich waren die Erläuterungen des Tutorials für alle und die verwendeten Begriffe allgemein für 94% verständlich.

Weiterhin gaben nur ein Viertel der Befragten an, dass sie eine Unterweisung oder ein Tutorial vor der Nutzung der Anwendung benötigt hätten. Bei Umfrage 1 waren es noch 71% (Version 1) beziehungsweise 43% (Version 2). Auch die Auffindbarkeit der Funktionen wies keine Probleme auf. So gaben 62,5% an, die Live-Tour schnell bis sehr schnell gefunden zu haben, bei den Hintergrundinformationen waren es sogar alle 16 Probanden (siehe Abbildung 5.3). Ebenso wurde auch die Offline-Tour von jedem sofort ausgemacht. Diese subjektiven Meinungen lassen sich

auch von den Videoaufzeichnungen bestätigen, die auf das gleiche Ergebnis schließen lassen. Aus diesen Ergebnissen lässt sich herleiten, dass die Optimierungen nach der vorherigen Umfrage die gewünschten Effekte geliefert hat.

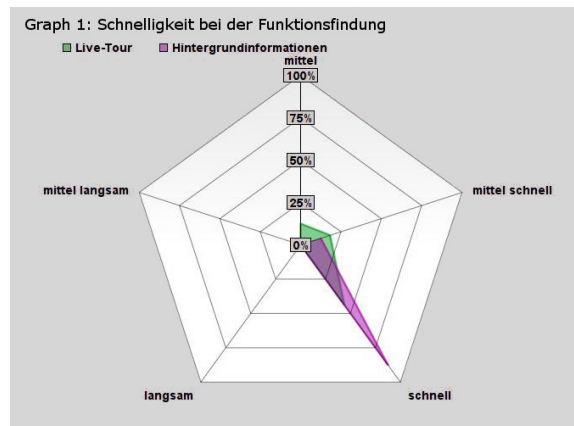


Abbildung 5.3: Schnelligkeit bei der Auffindung der Funktionen des finalen Prototyps.

Hauptziel der Evaluation war es außerdem, die unterschiedlichen Bilderkennungsverfahren auf ihre Nutzbarkeit und Robustheit zu testen. Wie bereits erwähnt, wurden anfangs zwei Tests ausschließlich mit dem Tracking mittels 3D-Map und zwei Tests ausschließlich mit dem Tracking mittels 3D CAD Model durchgeführt. Diese Tests sollten die Vortests in der Annahme bestätigen, dass die Bilderkennung mit nur einem Verfahren suboptimal ist. Dies wurde auch dadurch bestätigt, dass das 3D-Map-Tracking bei der Naherfahrung keine Ergebnisse lieferte, bei der Fernerfahrung hingegen schon. Diese waren auch ausreichend stabil und wurden in weiteren Tests evaluiert. Das Kantentracking hingegen lieferte bei der Naherfahrung gute Ergebnisse, konnte allerdings aus der Ferne keine gute Initialisierung durchführen. Somit wurden die Erkenntnisse aus Kapitel 4.3 bestätigt und weitere Tests mit der optimalen Variante durchgeführt.

Abbildung 5.4 zeigt, dass die Initialisierung mittels Kantentracking bei der Naherfahrung wurde von 25% der Nutzer mittel schnell bis schnell und von 58% als mittel empfunden. Sobald das Bild erkannt wurde und das Modell der Hauptsynagoge zu sehen war, gaben alle Partizipanten an, dass das Modell gut zu erkennen war. Die Einbettung in das städtische Szenario wurde nur von 17% als mittel schlecht bewertet. Probleme waren bei der Stabilität des Modells zu erkennen. 75% der Nutzer hatten hier mittlere bis mittel schlechte Erfahrungen gemacht, da das Modell noch sehr wackelig an seiner Position stand. Aber die Reinitialisierung, das heißt die Neupositionierung der Synagoge bei Verlust des Trackings, funktionierte bei 58% mittel gut bis gut (siehe Abbildung 5.4). Weiterhin war keine auffällige Beeinflussung durch den Tageszeiten- oder Belichtungswechsel zu erkennen. Nur nachts konnte es zu Problemen kommen, wenn sich die Kante des Gebäudes nicht mehr so eindeutig vom Nachthimmel im Hintergrund abhebt. Letztendlich ist zu sagen, dass einige Feineinstellungen vorgenommen werden müssen, um die Stabilität der Nachverfolgung zu verbessern.

Die Fernerfahrung lieferte bei 75% eine mittel schnelle bis schnelle Initialisierung, bei 17% eine langsame beziehungsweise sogar gar keine Erkennung (siehe Abbildung 5.4). Diese zwei Ausfälle sind zum einen mit einem Sicht-blockierenden Fahrzeug zu erklären und zum anderen mit einem starken Belichtungsunterschied zur frühen Tageszeit. Allerdings konnten auch hier alle Probanden nach Initialisierung der realen Szene die Projektion der Synagoge gut erkennen und bewerteten die Einbettung zu 67% mit mittel gut und zu 25% mit gut. Auch die Stabilität des Modells wurde von drei Viertel positiv bewertet. Zu sehen war hier, dass das Modell nicht wackelte oder sich verzog, sobald sich der Nutzer bewegte. Aufgrund des an dieser Position genutzten Trackingverfahrens kam es bei Bewegung in einigen seltenen Fällen zum Verlust der Erkennung, diese konnte allerdings meist wieder hergestellt werden. Zusätzlich empfanden einige Tester die

entferntere Position als die angenehmere, da hier die Hauptsynagoge besser im städtischen Umfeld zu erkennen war und auch durch die Verdeckung durch das Künstlerhaus authentischer im städtischen Szenario eingebettet wirkte.

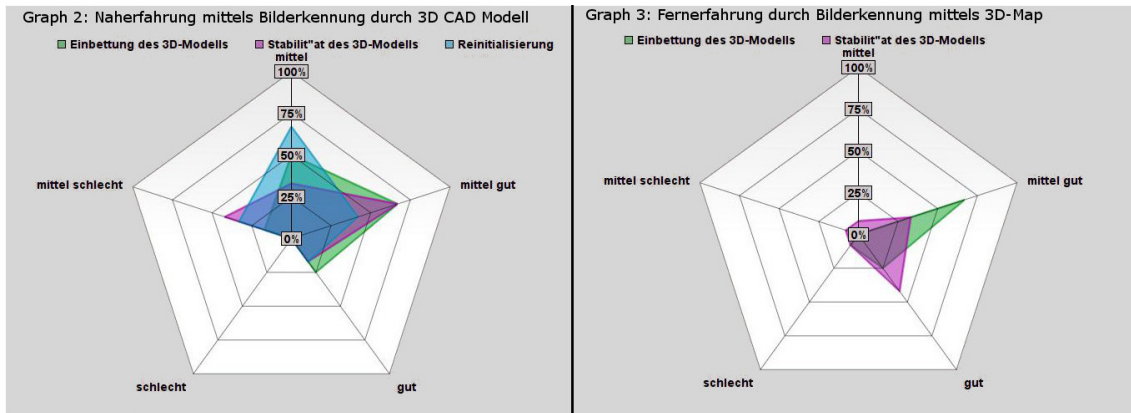


Abbildung 5.4: Bewertung der Trackingverfahren bei der Nah- und Fernerfahrung des finalen Prototyps des *HistoryTravellers*.

Zuletzt wurde die Offline-Tour durch die Partizipanten der Studie evaluiert. Hier sollten die Positionen durchgeschaltet und jeweils die Transparenz des Modells der Synagoge variiert werden. Die Funktionen waren für 92% der Teilnehmer verständlich. Die Performanz wurde von 25% als mittel gut und sogar 75% als gut befunden. Die Positionierung der Bilder wurde ebenfalls fast ausschließlich als gut befunden. Der Nutzen solch einer Offline-Tour wurde nur von der Hälfte der Teilnehmer als gut empfunden. Einige Tester äußerten, dass sie solch eine Funktion nicht benötigten, da sie die App nur zur Erfahrung von Augmented Reality starten würden. Des Weiteren konnte man beobachten, dass einige Partizipanten den Drang hatten, das Modell zu bewegen beziehungsweise an die Szenerie heranzuzoomen. Hier ergab sich bei weiteren Nachforschungen, dass die Nutzer bei der Offline-Tour die Szene gerne genauer betrachtet hätten und sich einige auch ein interaktives 3D-Erlebnis gewünscht hätten.

Summa summarum ist die Anwendung und vor allem die Idee dahinter bei allen Befragten sehr gut angekommen. Die App und auch ihre Funktionen wurden, trotz noch auftretender Fehler, mit einer Durchschnittsnote von 2,44 bewertet (nach Schulnoten). Dies entspricht einer guten Beurteilung. Des Weiteren zeigt das Resultat der zweiten Evaluierung, dass alle gesetzten Ziele erreicht und die Funktionen eines Augmented Reality Tourist Guides umgesetzt wurden. Sollten noch einige Einstellungen beim Tracking optimiert werden, so kann man davon ausgehen, dass die Anwendung noch besser bewertet wird.

6 Fazit

Zum Abschluss dieser Arbeit werden noch einmal die wichtigsten Punkte zusammengefasst und die Erkenntnisse dargestellt. Anschließend werden auf dieser Basis Verbesserungsmöglichkeiten erörtert, die anhand der Ergebnisse Sinn machen, oder die das Ergebnis der Arbeit um weitere Faktoren bereichern würden. Zusätzliche Möglichkeiten zur Weiterentwicklung und Potential in der Fortführung des Prototyps werden zum Abschluss dieser Arbeit dargestellt.

6.1 Zusammenfassung

In dieser Arbeit wurde der Prototyp eines Augmented Reality Tourist Guides mit dem Namen *HistoryTraveller* erstellt. Dabei hat man versucht, die Erfahrungen und Erkenntnisse aus bereits bestehenden Augmented City Guides zu nutzen und bestehende Problemfelder mit Hilfe der von *Metaio* zur Verfügung gestellten Technologien sinnvoll zu lösen. Wie auch schon die vorgestellten Applikationen beinhaltet der *HistoryTraveller* die grundlegenden Funktionen eines Tourist Guides. Zu diesen gehören die Navigations- und die Informationsfunktion. Zusätzlich bietet die Anwendung die Möglichkeit, die gewünschte Sehenswürdigkeit (in dieser Arbeit die *Alte Münchner Hauptsynagoge*) in München vor Ort an ihrer ehemaligen Position mittels Augmented Reality zu erleben.

Bei der Entwicklung dieses Prototypen lag der Fokus auf der Lösung von aktuellen Problemfeldern, die bei bisherigen Vertretern von Augmented City Guides analysiert wurden. So wurde für die Umgebungserkennung eine Kombination von Bilderkennungsverfahren verwendet, die sich aufgrund der Flexibilität und Stabilität von den Erkennungsverfahren der bisherigen Anwendungen abhebt. Ohne die Einschränkungen der sensor- oder markerbasierten Erkennungsverfahren, erlaubt die Kombination aus Kantentracking und 3D-Map-Tracking eine flexible, performante, markerlose und exakte Umgebungserkennung von zwei unterschiedlichen Positionen. Mit Hilfe des genutzten Verfahrens konnten auch wechselnde Tages- und Jahreszeiten berücksichtigt werden, sodass die Umgebungserkennung fast jederzeit ohne größere Einschränkungen funktionierte. Sowohl tagsüber bei Helligkeit oder nachts bei Dunkelheit als auch bei Sonnenschein oder bei Regenwetter ermöglicht der *HistoryTraveller* durch das Trackingverfahren eine Augmented Reality-Erfahrung. Des Weiteren hebt sich der *HistoryTraveller* durch die Implementierung des CAD Modells der *Alten Hauptsynagoge* von den Video- oder Bildaugmentationen der anderen Anwendungen ab. In Kombination mit der exakten Umgebungserkennung wurde darauf geachtet, dass die Positionierung des Modells in den städtischen Raum präzise und den Vorgaben entsprechend stattfand. Auch bei der Einbettung und Gestaltung des Modells wurden die Lichtverhältnisse und Tageszeiten berücksichtigt, um ein möglichst realitätsnahes Erlebnis der Synagoge zu ermöglichen. Die integrierte Verdeckung durch unveränderliche Objekte verstärkte diese Illusion. Diese Berücksichtigung von Verdeckung und Transparenz verbesserte zusätzlich die Erscheinung des 3D-Modells und zeigt einen weiteren Fortschritt bei der Augmentierung von Szenen im Außenraum.

6.2 Verbesserungsmöglichkeiten

Die Auswertung der Evaluierungen hat gezeigt, dass die Grundfunktionen bereits ausreichend im Prototyp implementiert sind. Allerdings gibt es Optimierungsmöglichkeiten, die sich während der Entwicklung herausstellten. So würde eine Verfeinerung des Kantentrackings die Erkennung noch performanter und weniger störanfällig machen. Zusätzlich kommt es aufgrund unterschiedlicher Brennweiten noch zu perspektivischen Abweichungen. Hier müssten die Brennweiten der Smartphone- und Creator-Kamera auf einen Nenner gebracht werden, damit keine Verzerrungen stattfinden.

Des Weiteren ist die Verdeckung durch ein virtuelles Verdeckungsmodell implementiert, dieses ist allerdings nur rudimentär umgesetzt. Möchte man die ideale Verdeckung verwirklichen, ist es notwendig, ein maßstabs- und detailgetreues Modell der unveränderlichen Objekte in der direkt angrenzenden Umgebung des eigentlichen Kernobjekts anzufertigen.

Weiterhin würden weitere an die äußeren Umstände angepasste Modell-Versionen der Synagoge das Erscheinungsbild noch realistischer gestalten. Mit unterschiedlichem Level-Of-Detail könnten verschiedene Detailstufen in die Anwendung integriert werden, die je nach Internetverbindung oder Rechenleistung angezeigt werden. Zusätzlich wäre es dann auch möglich, mittels Heranzoomens an die Synagoge dortige Details genauer zu betrachten.

6.3 Anwendungen und Chancen

Sollten die angesprochenen Verbesserungen realisierbar sein, entstünde eine marktreife Applikation für Smartphones und Tablets. Diese Anwendung könnte abhängig vom Konzept zum Beispiel auf weitere, bereits zerstörte Gebäude erweitert werden und letztendlich als Tourist Guide für eine komplette Stadtrundtour dienen. Der Erweiterung auf weitere Städte oder sogar Länder sind dabei keine Grenzen gesetzt.

Des Weiteren würde die Implementierung von Zusatzfunktionen wie etwa einer virtuellen Innenraumtour durch das digitale Gebäude den Nutzen solch eines Augmented City Guides erweitern. Weiterhin würde sich eine Mehrsprachigkeit der Anwendung und die Einbettung einer Sprachausgabe zu den interessanten Informationen anbieten, um den Tourist Guide vor allem auch für ausländische Touristen interessant zu machen.

Durch die bereits in der Einleitung angesprochenen technischen Fortschritte und durch die Verbreitung von mobilen Endgeräten oder Datenbrillen haben solche Applikationen vor allem in der Tourismusbranche eine hohe Relevanz. Über unterschiedliche Kanäle wäre es möglich, solch eine Tourismus-App zu vermarkten.

Mit den Erkenntnissen aus der Entwicklung dieses Prototyps und weiteren, zukünftigen Verbesserungen könnte man diese Arbeit als den Anfang der Entwicklung eines marktreifen Konzepts sehen. Hat man sich bei der Umsetzung dieses Konzepts anfangs nur auf ein einzelnes Gebäude einer einzigen Stadt beschränkt, so kann es mit der Weiterentwicklung der Applikation das Ziel sein, überall auf der Welt mit Hilfe eines On-Site 3D Augmented Reality Tourist Guides die Vergangenheit aufleben zu lassen oder die Geschichte unserer heutigen Zeit virtuell zu erleben.

Inhalt der beigelegten CD

1. **Masterarbeit-SimonPfaab** (Elektronische Version der Arbeit im Originalformat)
2. **HistoryTraveller** (Im Rahmen der Arbeit erstellte Software)
3. **Screenshots, Bilder, Videos** (Screenshots, Bilder und Videos zur erstellten Software)
4. **Literatur** (In der Arbeit zitierte Quellen, in elektronischer Form)
5. **Antrittsvortrag** (Folien vom Antrittsvortrag zu der Arbeit)
6. **Evaluierung Benutzeroberfläche** (Umfragebögen der ersten Nutzerstudie zur Benutzeroberfläche)
7. **Evaluierung finaler Prototyp** (Umfragebögen der zweiten Nutzerstudie zum finalen Prototypen)
8. **Inhaltsverzeichnis und Erläuterung.txt** (Inhaltsverzeichnis der CD)

Literatur

- [1] Timetraveler augmented Ltd. Timetraveler. <http://timetraveler.berlin//>, 2016. accessed 19.01.2016.
- [2] R. Azuma. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.
- [3] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, . Julier, and B. MacIntyre. Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, 21(6):34–47, 2001.
- [4] M. Bajura, H. Fuchs, and R. Ohbuchi. Merging virtual objects with the real world: Seeing ultrasound imagery within the patient. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 203–210. ACM, 1992.
- [5] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1):341–377, 2011.
- [6] T. Caudell and D. Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on*, volume 2, pages 659–669. IEEE, 1992.
- [7] Y. Cho, J. Lee, and U. Neumann. A multi-ring color fiducial system and an intensity-invariant detection method for scalable fiducial-tracking augmented reality. In *In IWAR*. Citeseer, 1998.
- [8] Colorbitor. Chronovisor. http://www.colorbitor.com/portfolio_page/cronovisor/, 2013. accessed on 03.11.2015.
- [9] A. Comport, E. Marchand, and F. Chaumette. A real-time tracker for markerless augmented reality. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, page 36. IEEE Computer Society, 2003.
- [10] A. Comport, E. Marchand, M. Pressigout, F. Chaumette, et al. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *Visualization and Computer Graphics, IEEE Transactions on*, 12(4):615–628, 2006.
- [11] Microsoft Corporation. Microsoft hololens - transform your world with holograms. <http://www.microsoft.com/microsoft-hololens/en-us>, 2014. accessed on 03.11.2015.
- [12] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410. IEEE, 2003.
- [13] D. DeMenthon and L. Davis. Model-based object pose in 25 lines of code. In *Computer Vision ECCV 1992*, pages 335–343. Springer, 1992.
- [14] M. Dhome, J. Lapresté, G. Rives, and M. Richetin. Spatial localization of modelled objects of revolution in monocular perspective vision. In *Computer Vision CCV 90*, pages 475–485. Springer, 1990.
- [15] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):932–946, 2002.
- [16] P. Dunston et al. Identification of application areas for augmented reality in industrial construction based on technology suitability. *Automation in Construction*, 17(7):882–894, 2008.

- [17] S. Geerkens. Google glass. <http://www.wearvision.de/googleglass/>, 2015. accessed on 09.08.2015.
- [18] Bibliographisches Institut GmbH. Fremdenführer, der. <http://www.duden.de/rechtschreibung/Fremdenfuehrer>, 2015. accessed on 29.10.2015.
- [19] R. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, and M. Kim. Pose estimation from corresponding point data. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(6):1426–1446, 1989.
- [20] W. Höhl. Landauerwalk. augmented reality app. <http://www.scienceviz.com/?p=1350>, 2015. accessed on 29.10.2015.
- [21] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.
- [22] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85–94. IEEE, 1999.
- [23] D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tuceryan. Real-time vision-based camera tracking for augmented reality applications. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 87–94. ACM, 1997.
- [24] M. Koob and M. Grellert. Synagogen in deutschland - eine virtuelle rekonstruktion. <http://www.cad.architektur.tu-darmstadt.de/synagogen/inter/menu.html>, 2015. accessed on 10.11.2015.
- [25] R. Kumar and A. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP: Image Understanding*, 60(3):313–342, 1994.
- [26] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial intelligence*, 31(3):355–395, 1987.
- [27] P. Milgram. Some human factors considerations for designing mixed reality interfaces. Technical report, DTIC Document, 2006.
- [28] P. Milgram and H. Colquhoun. A taxonomy of real and virtual world display integration. *Mixed reality: Merging real and virtual worlds*, pages 5–30, 1999.
- [29] Metaio GmbH Munich. Metaio developer portal. <https://my.metaio.com/dev/>, 2015. accessed 15.12.2015.
- [30] W. Narzt, G. Pomberger, A. Ferscha, D. Kolb, R. Müller, J. Wiegardt, H. Hörtnner, and C. Lindinger. Augmented reality navigation systems. *Universal Access in the Information Society*, 4(3):177–187, 2006.
- [31] D. Nistér. Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329, 2005.
- [32] J. P. Rolland, L. Davis, and Y. Baillet. A survey of tracking technology for virtual environments. *Fundamentals of wearable computers and augmented reality*, 1:67–112, 2001.

- [33] E. Rose, D. Breen, K. H. Ahlers, C. Crampton, M. Tuceryan, R. Whitaker, and D.s Greer. Annotating real-world objects using augmented reality. In *Computer Graphics: Developments in Virtual Environments (Proceedings of CG International 1995 Conference)*, pages 357–370, 1995.
- [34] B. Schaufler, P. Burkert, and S. Straubinger. Zeitfenster app. <http://www.zeitfenster-app.de>, 2013. accessed on 02.11.2015.
- [35] R. Schöffel and U. Köppen. Das making-of zur augmented-reality-app. <http://www.br.de/fernsehen/das-erste/sendungen/kurt-landauer-der-film/die-app-making-of-100.html>, 2014. accessed on 02.11.2015.
- [36] H. Schumacher. Wahre sensibelchen: ein überblick über wirklich alle handy-sensoren, auch über kaum bekannte. <http://www.androidmag.de/report/ein-wahres-sensibelchen-handy-sensoren/>, 2014. accessed on 29.10.2015.
- [37] I. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.
- [38] S. You, U. Neumann, and R. Azuma. Hybrid inertial and vision tracking for augmented reality registration. In *Virtual Reality, 1999. Proceedings., IEEE*, pages 260–267. IEEE, 1999.
- [39] F. Zhou, H. Duh, and M. Billinghurst. 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*, pages 193–202. IEEE Computer Society, 2008.