



UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik



COMPUTERVISUALISTIK

Entwicklung eines augmentierten Montageszenarios mit Lego-Bausteinen für die Microsoft HoloLens

Bachelorarbeit

zur Erlangung des Grades Bachelor of Science (B.Sc.)
im Studiengang Computervisualistik

vorgelegt von
Darius Thies

Erstgutachter: Prof. Dr.-Ing. Stefan Müller
(Institut für Computervisualistik, AG Computergraphik)
Zweitgutachter: Anna Katharina Hebborn, M.Sc.
(Institut für Computervisualistik, AG Computergraphik)

Koblenz, im Dezember 2017

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.

Koblenz, 21. Dezember 2017

.....
(Ort, Datum) (Unterschrift)

Zusammenfassung

Augmented Reality ist schon seit einigen Jahren in verschiedenen Formen verfügbar. Durch Fortschritte in der Technik können nun auch kompakte Augmented Reality Brillen hergestellt werden, wodurch sich viele neue Möglichkeiten der Interaktion und Anwendung von Augmented Reality eröffnen.

Diese Arbeit beschäftigt sich mit der Microsoft HoloLens und den Möglichkeiten, welche die Brille für Verbraucher und Industrie bieten kann. Im Rahmen dieser Arbeit wurde eine interaktive und augmentierte Anwendung entwickelt, um die Grenzen und Möglichkeiten der Microsoft HoloLens zu bewerten. Dazu wurde exemplarisch ein Montageszenario nachgestellt, bei dem ein Modell aus Lego-Bausteinen zusammengesetzt wird. Die HoloLens zeigt dabei schrittweise an, wo die nächsten Bausteine platziert werden sollen. Dabei stellte sich heraus, dass die HoloLens schon heute Vorteile bei der Montage bietet und gut zur Unterstützung von Arbeitsschritten verwendet werden kann, es wurden allerdings auch einige Schwächen deutlich.

Abstract

Augmented reality is being present for many years. Through progress in technology smaller augmented reality glasses became possible. These new technologies allow many new ways of interaction and usage of augmented reality.

This thesis is about the Microsoft HoloLens and its possibilities for consumers and industry. In the context of this thesis a new interactive and augmented application to measure the possibilities and limitations of the Microsoft HoloLens has been developed. The scene is an assembly szenario with a step by step instruction of building with Lego bricks. The evaluation showed that the HoloLens can already be used to assist in assembling scenarios and offers some advantages over other methods, although the glasses still have some flaws.

Inhaltsverzeichnis

Abbildungsverzeichnis	1
Abkürzungsverzeichnis	2
1 Einleitung	3
2 Grundlagen	4
2.1 Augmented Reality	4
2.2 Augmented und Virtual Reality	5
2.3 Augmented-Reality Geräte	5
2.3.1 Microsoft HoloLens	8
2.4 Augmented Reality zur Unterstützung von Montage	10
2.4.1 Motivation	10
2.4.2 Nutzung in der Industrie	10
2.4.3 Nutzung im Verbrauchermarkt	12
2.4.4 Ähnliche Arbeiten	13
3 Implementierung	15
3.1 Verwendete Programme	15
3.2 Ablauf des Montageszenarios	15
3.3 Aufbau von HoloLego	16
3.4 Programmablauf	16
3.5 Vorstellung wichtiger Skripte	19
3.5.1 Construction Manager	20
3.5.2 Layout Renderer	22
3.5.3 HoloToolkit Skripte	23
3.6 Tracking	23
3.7 Benutzerinterface und Interaktion	24
3.7.1 Cursor	25
3.7.2 Menü	25
3.7.3 Sprachsteuerung	27
3.7.4 Darstellung der Platzierung der Legosteine	27
3.8 Verwendung eigener Legomodelle	28
3.9 Nicht implementierte Funktionen	28
4 Evaluation	30
4.1 Aufbau und Gestaltung der Umfrage	30
4.1.1 Szenario	30
4.1.2 Durchführung	31
4.2 Ergebnisse	31
4.2.1 Bearbeitungszeiten	31
4.2.2 Ergebnisse des Fragebogens	32
4.3 Auswertung der Ergebnisse	34

5 Ausblick	36
5.1 AR-unterstützte Montage	36
5.2 HoloLens	36
5.3 HoloLego Applikation	36
6 Fazit	37
Literatur	38

Abbildungsverzeichnis

1	Reality-Virtuality (RV) Kontinuum [1]	5
2	Optical see-through Head-Mounted-Display (HMD) — Schema [2]	6
3	Video see-through HMD — Schema [2]	6
4	Monitor basiertes System — Schema [2]	7
5	Beispiel eines optischen Head-Up Display (HUD) in einem BMW i8 [3]	8
6	Microsoft HoloLens [4]	9
7	Beispiele für AR-Anwendungen	11
8	Visualisierung der Anzahl an Publikationen für die Kombination aus Themenbereich und Hardwareplattform [5]	12
9	Darstellung der In-situ Anweisungen von Blattgerste et al. [6], weiße Einblendungen zeigen Box zum Greifen und Zielposition an	13
10	Darstellung der HoloLego Applikation durch die HoloLens, alle bereits platzierten Steine angezeigt	16
11	Szenenhierarchie von HoloLego	17
12	Programmablaufplan von HoloLego	18
13	Für das Tracking verwendete Marker	24
14	Überblick über die verschiedenen Cursor	25
15	Interaktives Menü, dargestellt mit der HoloLens	26
16	Überblick über die verschiedenen Hilfslinien, dargestellt mit der HoloLens	28
17	Darstellung der HoloLego Applikation durch die HoloLens, nur aktueller Stein wird angezeigt	29
18	Gesamtzeit für den Aufbau	32
19	Zeiten für die verschiedenen Schritte im Vergleich	33

Abkürzungsverzeichnis

2D	zweidimensional
3D	dreidimensional
AR	Augmented Reality
AV	Augmented Virtuality
fbx	Filmbox, ein Dateiformat von Autodesk
FPS	Bilder pro Sekunde
HMD	Head-Mounted-Display
HPU	Holographic Processing Unit
HUD	Head-Up Display
MR	Mixed Reality
OHMD	Optical Head-Mounted-Display
RGB	Rot, Grün und Blau Farbmodell
RV	Reality-Virtuality
VHMD	Video (based) Head-Mounted-Display
VR	Virtual Reality

1 Einleitung

Augmented Reality (AR) steht schon schon seit einigen Jahren in verschiedenen Formen zur Verfügung. Durch fortschrittlichere Technologien ist nun auch die Herstellung kompakter Augmented Reality Brillen möglich, die viele neue Möglichkeiten der Interaktion und Verwendung von Augmented Reality erlauben. Das Einblenden von Anweisungen und Beschreibungen in die reale Welt kann zum Beispiel die Effizienz steigern oder die Anzahl an Fehlern reduzieren.

Diese Arbeit beschäftigt sich hauptsächlich mit den Möglichkeiten, die AR-Brillen und insbesondere die Microsoft HoloLens für die Montage bieten können. Um die Grenzen und Möglichkeiten der Microsoft HoloLens zu bewerten wird dazu eine interaktive und augmentierte Anwendung entwickelt und implementiert.

Dabei soll exemplarisch ein Montageszenario nachgestellt werden, bei dem ein Modell aus Lego-Bausteinen zusammengesetzt wird. Die HoloLens zeigt dabei schrittweise an, wo die nächsten Bausteine platziert werden sollen. Dazu werden zunächst die relevanten Begriffe und Grundlagen, sowie bereits vorhandene Erkenntnisse anderer Arbeiten, vorgestellt. Anschließend wird der Aufbau und die Implementierung der Anwendung erklärt. Die fertige Applikation wird daraufhin in einem Nutzertest evaluiert und die Ergebnisse werden vorgestellt.

2 Grundlagen

Dieser Teil der Arbeit erklärt wichtige Begriffe und die theoretischen Grundlagen zum Thema AR, deren Abgrenzung zur Virtual Reality (VR) und Verwendung in der Industrie und zur Montage. Außerdem werden bisherige Erkenntnisse erläutert und ähnliche Publikationen vorgestellt.

2.1 Augmented Reality

Unter erweiterter Realität (englisch augmented reality) versteht man das Einblenden von computergenerierten Daten in die reale Umgebung, die an bestimmte Orte oder Aktivitäten gebunden sind. AR erlaubt es, digitale Inhalte nahtlos mit unserer Wahrnehmung der realen Welt zu überlagern und zu verbinden. Neben zweidimensionalen (2D) und dreidimensionalen (3D) Objekten können dabei auch Ton, Video oder Text verwendet werden, um das Wissen und Verständnis des Benutzers zu verbessern und die Realität zusammen mit den hinzugefügten Inhalten als Einheit zu sehen [7].

Manche Definitionen von AR limitieren den Begriff auf die Benutzung von Head-Mounted-Displays (HMDs). Dies sind auf dem Kopf getragene Geräte mit einem Bildschirm vor den Augen des Nutzers. Um AR allgemeiner zu definieren, muss ein AR-System laut Definition von Azuma folgende Eigenschaften erfüllen [7, 2]:

1. Kombination von virtuellen und realen Elementen
2. Interaktivität in Echtzeit
3. Registriert in 3D (virtuelle Objekte sind an Koordinaten der echten Welt gebunden)

Echtzeit bedeutet in diesem Fall, dass Veränderungen durch Interaktion ohne merkliche Verzögerung geschehen. Üblicherweise werden dafür 60 Bilder pro Sekunde (FPS) als Minimum angenommen. Diese Anforderung ist für stabile Hologramme notwendig.

Neben dem Hinzufügen von virtuellen Objekten kann erweiterte Realität auch dazu genutzt werden, um komplette reale Objekte oder Teile davon unsichtbar zu machen oder zu verdecken. So lässt sich zum Beispiel ein realer Tisch entfernen, indem im AR-Anzeigegerät die Stelle, an der der Tisch sichtbar wäre, mit der Hintergrundfarbe angezeigt wird [2]. Auch teilweise Verdeckung zwischen realen und virtuellen Objekten ist möglich und kann verwendet werden, um virtuelle Objekte wie erwartet im Raum darzustellen.

2.2 Augmented und Virtual Reality

Im Gegensatz zu AR, das die Welt digital erweitert, wird bei VR eine komplett computergenerierte virtuelle Welt dargestellt. Zwischen der realen Welt und der virtuellen Welt gibt es einen Übergangsbereich, in dem die realen und digitalen Umgebungen kombiniert sind. Milgram stellt dies zur Verdeutlichung in einem Reality-Virtuality (RV) Kontinuum (siehe [Abbildung 1](#)) dar [1].

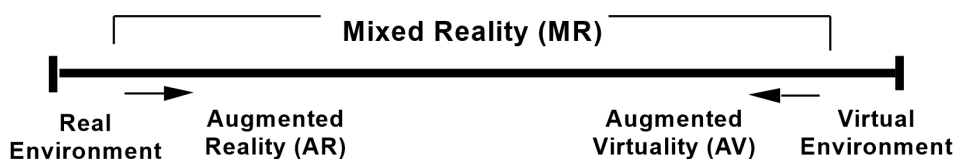


Abbildung 1: RV Kontinuum [1]

Die linke Seite des Kontinuums definiert eine Umgebung aus realen Objekten und allem was beim Betrachten der realen Szene gesehen werden kann, egal ob direkt von einer Person, durch ein Fenster oder auf einem Bildschirm. Die rechte Seite definiert Umgebungen, die nur aus virtuellen Objekten bestehen [1]. AR ist hier näher an der echten Umgebung, die durch digitale Daten erweitert wird. Augmented Virtuality (AV), auf deutsch auch erweiterte Virtualität genannt, ist ähnlich wie AR, enthält allerdings größtenteils computergenerierte Umgebungen mit nur wenigen realen Bildern [7].

Alle Bereiche, bei denen reale und virtuelle Umgebungen kombiniert werden, werden als Mixed Reality (MR), zu deutsch gemischte Realität, bezeichnet. Dazu gehören wie in [Abbildung 1](#) zu sehen sowohl AR als auch AV.

2.3 Augmented-Reality Geräte

Augmented-Reality-Geräte werden genutzt, um die digitalen Inhalte anzuzeigen und mit den realen zu verknüpfen. Beim Design von AR-Geräten unterscheidet man zwischen zwei Möglichkeiten, die realen und digitalen Inhalte zu kombinieren. Die Inhalte können optisch oder per Video kombiniert werden. Ein see-through HMD ist ein Gerät, das es erlaubt die reale Umgebung mit digitalen Inhalten zu kombinieren. Im Gegensatz zu typischen HMDs für virtuelle Realität, die keine Sicht auf die reale Welt erlauben.

Optical see-through HMDs oder auch Optical Head-Mounted-Displays (OHMDs) genannt funktionieren, indem ein optischer halbdurchlässiger Kombiniierer vor den Augen platziert ist, durch den der Nutzer die Um-

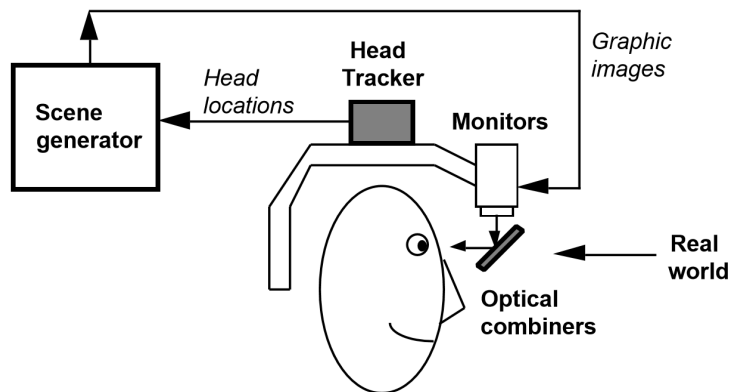


Abbildung 2: Optical see-through HMD — Schema [2]

gebung betrachten kann. Der Kombinierer ist halbreflektierend, sodass die digitalen Inhalte von Bildschirmen reflektiert werden und dem Benutzer angezeigt werden können. Alternativ werden auch Projektoren verwendet, um die digitalen Bilder direkt an die passende Position in der realen Umgebung zu projizieren. Ein schematischer Aufbau davon ist in [Abbildung 2](#) zu sehen. Beispiele für OHMDs sind die in dieser Arbeit verwendete Microsoft HoloLens (siehe Abschnitt [2.3.1](#)) oder Google Glass.

Die Linsen sind halbversilberte Spiegel, die nur einen Teil des Lichtes der Umgebung anzeigen und einen zweiten Teil des Lichtes von Monitoren reflektieren. Daher verringert der optische Kombinierer den Anteil an Licht, den der Benutzer von der Umgebung sieht. Ist der Monitor ausgeschaltet funktioniert das Gerät ähnlich einer Sonnenbrille. Das richtige Verhältnis zwischen durchgelassenem und reflektiertem Licht zu finden ist ein Design-Problem beim Entwickeln von OHMDs [2].

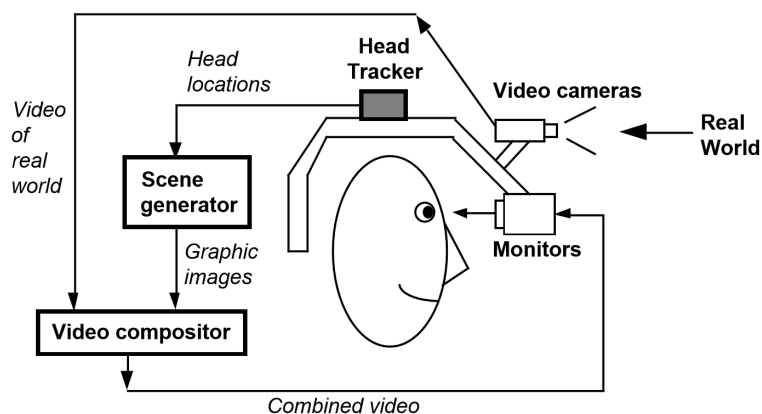


Abbildung 3: Video see-through HMD — Schema [2]

Eine andere Möglichkeit digitale Inhalte mit realen zu kombinieren bieten video see-through HMDs, im folgenden Video (based) Head-Mounted-Displays (VHMDs) genannt. Diese Geräte kombinieren die Videos der Kameras mit den computergenerierten Einblendungen und zeigen dieses auf einem Bildschirm vor den Augen des Nutzers an. Alternativ können die Bilder auch direkt auf die Netzhaut projiziert werden. Ein schematischer Aufbau eines VHMDs ist in [Abbildung 3](#) zu sehen.

Eine einfache Möglichkeit, die realen mit den generierten Bildern zu kombinieren, bietet die farbbasierte Bildfreistellung (engl. chroma keying), wie sie auch in der Filmindustrie unter dem Begriff Greenscreen verwendet wird. Dabei setzt man den Hintergrund für die erzeugten Bilder auf eine bestimmte Farbe. Beim Zusammenfügen ersetzt man alle Vorkommen dieser Farbe mit den Bildern der Kameras, also der echten Umgebung. Falls das System für jeden Pixel des Kamerabildes über Tiefenwerte verfügt, können diese auch verwendet werden, um für jeden Pixel zu entscheiden, ob der reale oder der computergenerierte Pixel im Vordergrund ist. Mittels dieser Methode lässt sich Verdeckung zwischen virtuellen und realen Gegenständen erreichen [2].

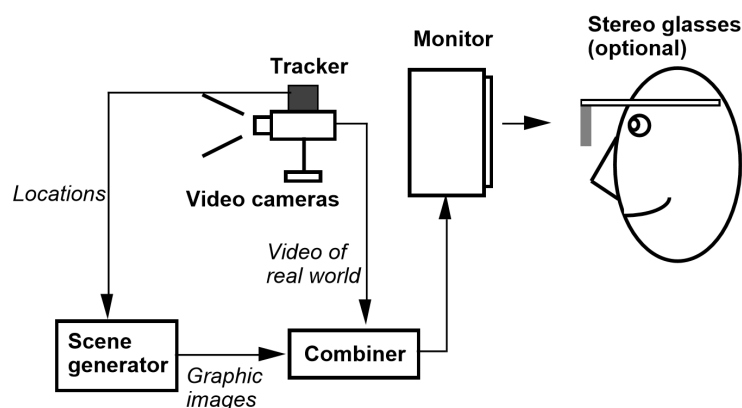


Abbildung 4: Monitor basiertes System — Schema [2]

AR-Systeme, die nicht am Kopf getragen werden, werden Monitor oder Projektor basierte Systeme genannt. Sie funktionieren ähnlich wie die Video basierten Brillen, nur dass der Bildschirm nicht direkt vor den Augen des Nutzers getragen wird, sondern zum Beispiel auf einem Tisch steht. Zur Bildausgabe können auch Projektoren verwendet werden. Eine oder mehrere Kameras filmen dabei die Umgebung, diese können statisch oder mobil sein. Mobile Kameras sind dabei zum Beispiel in einem Smartphone integriert oder am Arm eines Roboters befestigt. Ein schematischer Aufbau eines Monitor basierten Systems ist in [Abbildung 4](#) zu sehen.



Abbildung 5: Beispiel eines optischen HUD in einem BMW i8 [3]

Monitor basierte Systeme können auch optisch funktionieren. Dazu werden, wie bei den OHMDs, die computergenerierten Inhalte durch einen optischen Kombinierer zusammen angezeigt, nur dass dieser nicht vom Nutzer getragen wird. Der Nutzer schaut durch die Anzeige und kann die reale Welt sowie die eingeblendeten Inhalte sehen. Diese Technik wird zum Beispiel in Head-Up Display (HUD)–Anzeigen in Flugzeugen oder modernen Autos verwendet. Ein Beispiel dazu ist in [Abbildung 5](#) zu sehen.

Das am weitesten verbreitete Gerät, das für erweiterte Realität verwendet werden kann, ist das Smartphone. Dieses kann die Kamera und seine Sensoren verwenden, um die Umgebung wahrzunehmen und je nach Anwendung verschiedene Informationen einzublenden. Bekannt wurde Augmented Reality beim Smartphone vor Kurzem besonders durch die Applikation Pokémon Go.

2.3.1 Microsoft HoloLens

Das für diese Arbeit verwendete Anzeigegerät ist die HoloLens von Microsoft. Die HoloLens (siehe [Abbildung 6](#)) ist laut Microsoft eine MR–Brille, die digitale Inhalte optisch in die Umgebung einblenden kann und damit zur in [Unterabschnitt 2.3](#) vorgestellten Kategorie der OHMDs gehört. Seit März 2016 ist die Brille für Entwickler verfügbar [8].

Als Optik hat die HoloLens zwei halbdurchlässige Linsen (Lichtwellenleiter), die aus mehreren Glasschichten für die einzelnen Farbkanäle bestehen. Beleuchtet werden die Linsen jeweils von einem Projektor mit 2,3 Millionen Bildpunkten, der oberhalb der Linsen platziert ist. Das Sichtfeld (Field of View) beträgt ungefähr 30° in der Breite und $17,5^\circ$ in der Höhe [9].

Folgende Sensoren stehen der HoloLens zur Verfügung:

- 1 Inertiale Messeinheit (enthält Beschleunigungs– und Drehratensensoren)

sor)

- 4 Kameras zum Filmen und Verständnis der Umgebung
- 1 Tiefenbildkamera
- 1 2 Megapixel Kamera für Foto- und Videoaufnahmen
- 4 Mikrofone
- 1 Umgebungslichtsensor

Die Brille läuft unter einer speziellen Version von Windows 10 und nutzt einen 32-bit Prozessor von Intel, sowie einen zusätzlichen von Microsoft entwickelten Prozessor, der Holographic Processing Unit (HPU) genannt wird. Der HoloLens stehen 64GB Flashspeicher und 2GB Arbeitsspeicher zur Verfügung.

Die Microsoft HoloLens wiegt 579g und hat eine Akkulaufzeit von 2-3 Stunden während der Nutzung [10].



Abbildung 6: Microsoft HoloLens [4]

Die Sensoren der HoloLens werden zur Bestimmung des Raumes, der Position, der Orientierung, etc. verwendet. Um den aktuellen Raum zu erkennen, wird aus den Tiefenwerten ein Abbild des Raumes erstellt und die aktuellen Messdaten mit den bereits bekannten Räumen abgeglichen. Wird ein bekannter Raum gefunden, werden automatisch zu einem früheren Zeitpunkt in diesem Raum platzierte Hologramme geladen.

Tiefenwerte, Kameras und die inertielle Messeinheit werden verwendet, um zu jedem Zeitpunkt die genaue Lage und Orientierung der Brille zu bestimmen und Hologramme an eine Position der realen Umgebung zu binden. Für den Nutzer entsteht so der Eindruck, als ob ein an der Wand platzierter digitaler Inhalt wirklich fest an der Wand hängt. Die Kameras werden auch zur Erkennung der Hände für die Gestensteuerung verwendet. Alternativ kann die Brille durch Sprachbefehle oder per Controller bedient werden.

2.4 Augmented Reality zur Unterstützung von Montage

Augmented Reality wird immer häufiger zur Unterstützung von Montage Szenarien verwendet. In diesem Abschnitt wird erläutert, warum AR verwendet wird und es werden Beispiele für die Verwendung gezeigt.

2.4.1 Motivation

Ingenieure und Techniker verbringen viel Zeit mit dem Lesen von Anweisungen und Dokumentationen, um sich alle Prozeduren einzuprägen. Augmented Reality kann dabei helfen diese Zeit zu reduzieren, indem die jeweils benötigten Daten und Anweisungen direkt in das Sichtfeld eingeblendet werden. Dies ermöglicht ein effektiveres Training. Dadurch wird die benötigte Anlernzeit zur Fertigung neuer Produkte oder bei der Einarbeitung neuer Mitarbeiter stark reduziert. Darüber hinaus können noch nicht fertig ausgebildete Arbeiter Aufgaben schneller erledigen [11].

Bei der Verwendung von AR-Systemen für die Unterstützung in der Fertigung werden zum Beispiel wichtige Parameter oder die Reihenfolge der Montage angezeigt. Häufig führt die digitale Unterstützung auch zu einer verringerten Fehlerrate in Produktionsprozessen [12]. Einige Umfragen zeigen, dass es durch industrielle erweiterte Realität Innovationsmöglichkeiten für viele Aspekte bei der Herstellung gibt, z.B. durch verbessern der Prozesse oder verringern der physikalischen und kognitiven Belastung der Arbeiter [5, 13]. Außerdem werden durch die Digitalisierung der Produktion und Industrie, auch „Industrie 4.0“ genannt, neue, effizientere und flexiblere Wege zur Produktion benötigt. Daher sind AR-Geräte als neue Schnittstelle zwischen Maschinen und Arbeitern besonders geeignet und bieten Vorteile im Assistieren für komplexe Industrie 4.0-Szenarien [5].

2.4.2 Nutzung in der Industrie

Bereits 1992 war die Nutzung von AR zur Unterstützung manueller Montage Teil einer Forschungsarbeit. Caudell et al. [14, 15] bauten einen AR-Brillen Prototyp, der Arbeiten bei der Flugzeugproduktion die Position von Bohrlöchern anzeigen kann. Seit dem hat sich die Technologie stark weiterentwickelt und die Sensoren und Kameras sind genauer, die Darstellung besser und die verwendeten Geräte kleiner und günstiger geworden.

Ein moderneres Beispiel zeigt [Abbildung 7a](#). Die dargestellte Anwendung hilft beim Auseinanderbauen einer Kaffeemaschine. Dabei wird Augmented Reality außerdem genutzt, um das Innere der Kaffeemaschine mit einer Art Röntgensicht darzustellen. Diese Ansicht kann dabei helfen zu planen, wie man die inneren Teile erreichen will und bei exakter Darstellung kann so auch einfacher an normalerweise nicht sichtbaren Bereichen gearbeitet werden.



(a) Röntgensicht zeigt die Brüheinheit einer Kaffeemaschine [16]

(b) Autoreparatur assistiert durch Experten per Telepräsenz [17]

Abbildung 7: Beispiele für AR-Anwendungen

Falls menschliche Hilfe benötigt wird, bietet erweiterte Realität auch die Möglichkeiten, Arbeiter oder auch Verbraucher bei Tätigkeiten wie Montage oder Reparatur per Telepräsenz zu unterstützen. [Abbildung 7b](#) zeigt zum Beispiel eine Person, die ein Auto repariert und dabei eine AR-Applikation verwendet. Ein Experte kann nun Beschreibungen und Anweisungen direkt mit dem 3D-Modell verknüpfen, indem er diese auf dem Tablet einzeichnet. Der Anwender sieht die Anweisungen in der Applikation dann an der passenden Position und kann die Hilfe einfacher umsetzen [17, 11].

Die aktuellen Interessen und Forschungsbereiche der Industrie in Bezug auf erweiterte Realität lassen sich in [Abbildung 8](#) erkennen. Im Diagramm sind die Anzahl der Publikationen für verschiedene Themenbereiche und Hardwareplattformen dargestellt ¹.

Dimension eins beschreibt das Anwendungsgebiet.

Manufacturing enthält unterstützte Montage, betreiben von Maschinen und ähnliches.

Logistics enthält manuelle Arbeiten wie aufheben von Paketen, Navigation oder Datenmanagement.

Maintenance enthält Anwendungen zur Wartung, z.B. Remote-Support via AR.

Training enthält Anwendungen, die beim Lernen bestimmter Szenarios helfen sollen.

Die zweite Dimension unterscheidet zwischen den verschiedenen Hardwareplattformen. Im Diagramm wird zwischen "AR mobile devices", also mobile Geräte wie Tablets und Smartphones, "AR projection", also dem

¹Unter <http://www.smartfactory-owl.de/designspace/designspace.html> lässt sich ein interaktives aktuelles Diagramm aufrufen.

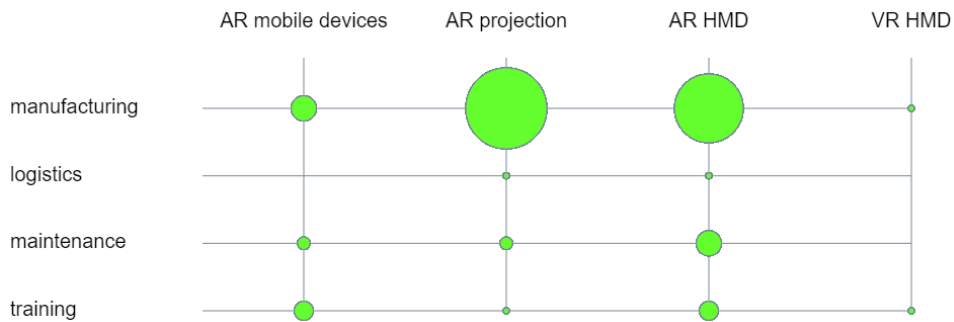


Abbildung 8: Visualisierung der Anzahl an Publikationen für die Kombination aus Themenbereich und Hardwareplattform [5]

Darstellen von AR-Inhalten durch einen Projektor, "AR-HMDs" und "VR-HMDs" unterschieden. Die letzte Dimension wird durch die Größe der Kreise dargestellt und bezieht sich auf die Anzahl an Publikationen zu dem jeweiligen Anwendungsgebiet mit der entsprechenden Hardware [5].

Zur Zeit ist die Verwendung von AR zur Unterstützung für die Produktion durch Projektion von AR-Inhalten oder HMDs also besonders interessant. Diese Arbeit würde sich unter "Manufacturing" und "AR-HMDs" einordnen lassen, dem zweit wichtigsten Anwendungsbereich.

2.4.3 Nutzung im Verbrauchermarkt

Für Verbraucher könnte es als Anleitung für gekaufte Produkte hilfreich sein, zum Beispiel wie die in dieser Arbeit entwickelte Anwendung für Lego, für den Zusammenbau von Möbeln oder die Reparatur von Autos. Bei Bedarf können Experten auch hier die Anwender per Telepräsenz unterstützen.

Momentan wird erweiterte Realität zur Montage allerdings hauptsächlich in der Industrie verwendet, da Verbraucher neben dem Smartphone nur selten andere AR-fähige Geräte besitzen. Smartphones eignen sich nur bedingt zur Unterstützung beim Zusammenbauen, da die Kamera immer wieder auf das Objekt gerichtet werden muss. AR-Brillen gibt es für nicht Privatkunden kaum und die vorhandenen Brillen sind meistens teuer und noch nicht komplett ausgereift. Die HoloLens kostet für Verbraucher zum Beispiel 5.000 €.

Wenn in Zukunft Geräte weiter verbessert und günstiger werden und der Markt größer wird, ist davon auszugehen, dass auch immer mehr Privatpersonen Augmented Reality zur Montage verwenden werden.



Abbildung 9: Darstellung der In-situ Anweisungen von Blattgerste et al. [6], weiße Einblendungen zeigen Box zum Greifen und Zielposition an

2.4.4 Ähnliche Arbeiten

Es gibt bereits einige andere Arbeiten, die sich mit dem Thema von unterstützter Montage von Lego durch erweiterte Realität beschäftigen haben. Funk et al. [15] entwickelten einen Benchmark, um augmentierte Montageszenarien besser vergleichbar zu machen. Das Ziel war außerdem, dass der Test einfach aufzubauen, zu wiederholen und skalierbar sein soll. Der Nutzertest soll auch alle wichtigen Aktionen wie Teile aufheben, platzieren und anbringen enthalten.

Der Benchmark wurde später verwendet, um den Aufbau eines Legomodells mit Papieranleitung, Epson Moverio BT-200 HMD, Tablet und in-situ² Projektion zu vergleichen. Das Epson Moverio zeigte die Instruktionen dabei nicht dreidimensional an, sondern verwendete die Bilder der Papieranleitung und stellte diese in der Mitte des Sichtfeldes dar. Dabei schnitten die Papieranleitung und die in-situ Projektion am Besten ab, während mit dem HMD viele Fehler gemacht wurden und der Aufbau die meiste Zeit benötigte [18].

Blattgerste et al. [6] führten einen Nutzertest nach dem Prinzip von Funk et al. durch und verglichen dabei die Papieranleitung mit dem Epson Moverio und in-situ Anweisungen auf Smartphone und HoloLens. Die AR-Anleitung ist in [Abbildung 9](#) zu sehen, dabei zeigt das Fadenkreuz die Box mit dem benötigten Stein an und der weiße Quader die Position zum Platzieren. Dabei benötigte der Aufbau mit der Papieranweisung die wenigste

²latenisch für am Ort, Objekte werden am Ort angezeigt, an dem sie platziert werden sollen

Zeit. Die HoloLens konnte allerdings die Anzahl an Fehlern im Vergleich zu den anderen digitalen Anweisungen stark reduzieren. Die Schlussfolgerung war, dass eine Kombination aus in-situ Anweisungen zum Anzeigen der korrekten Position und Piktogramme für eine genaue Montage die besten Ergebnisse erzielen würde [6].

3 Implementierung

Dieses Kapitel beschreibt den Aufbau und die Implementierung der Anwendung. Dabei wird zuerst ein grober Überblick über den Aufbau gegeben und anschließend werden wichtige Skripte und Funktionen genauer vorgestellt.

Die Applikation wird im Folgenden auch HoloLego genannt. Außerdem werden *Skripte* und `Funktionen()` wie gezeigt dargestellt.

3.1 Verwendete Programme

Zur Entwicklung der Applikation wurde Unity³ in der Version 2017.1.0f3 verwendet. Unity ist momentan die einzige von Microsoft unterstützte Engine, um Applikationen für die HoloLens zu entwickeln. Als Erweiterungen wurden Vuforia⁴ für das Tracking und HoloToolkit⁵ (seit kurzem in MixedRealityToolkit umbenannt) für die einfache Verwendung und Einrichtung der HoloLens mit Unity genutzt.

Zur Erstellung der Lego-Objekte wurde LDCad von LDraw⁶ verwendet, das eine komplette Datenbank aller Legosteine enthält. Um diese Dateien mit Unity zu nutzen, wurde zur Konvertierung in das fbx-Format Blender⁷ mit der LDraw-Importer Erweiterung⁸ verwendet.

3.2 Ablauf des Montageszenarios

Die für diese Arbeit entwickelte Anwendung ist eine Montageanleitung für die HoloLens. Wie die Anwendung durch die HoloLens aussieht ist im Screenshot in [Abbildung 10](#) zu sehen. Um diese Aufnahmen zu erstellen wurde Mixed Reality Capture verwendet. Dieses Feature erlaubt es, die Hologramme mit dem Kamerabild kombiniert aufzunehmen. Dabei kann es allerdings zu Verschiebungen der Hologramme kommen, so dass die Aufnahmen nicht immer genau so aussehen, wie beim Blick durch die HoloLens.

Der Anwender sieht neben dem zu platzierenden Legomodell ein Menü zur Bedienung. Es wird schrittweise jeweils ein Legosteine angezeigt, der platziert werden soll. Dieser bewegt sich dabei auf- und abwärts, um zu zeigen, wo und wie der Stein angebracht werden soll. Nachdem die Anweisung vom Nutzer befolgt wurde, kann dieser zum nächsten Schritt wech-

³<https://unity3d.com/de/>

⁴<https://www.vuforia.com/>

⁵<https://github.com/Microsoft/MixedRealityToolkit-Unity>

⁶<http://www.ldraw.org/>

⁷<https://www.blender.org/>

⁸https://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Import-Export/LDRAW_Importer

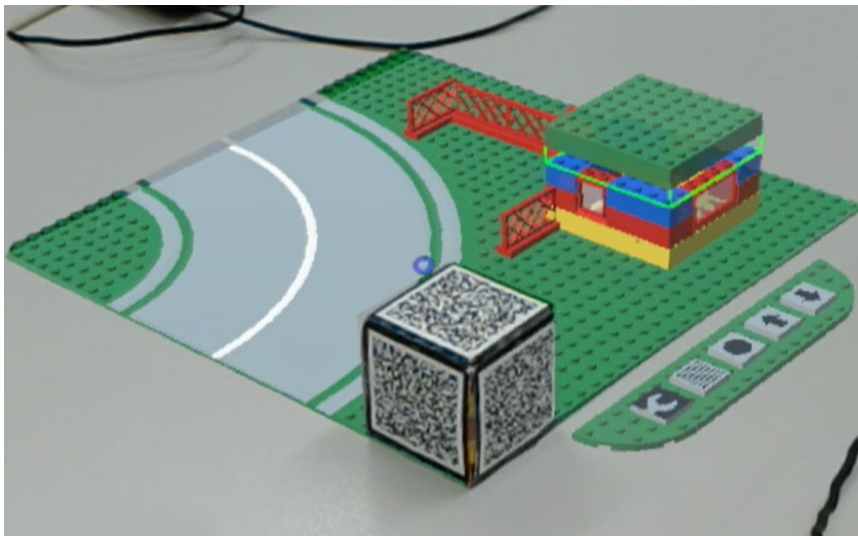


Abbildung 10: Darstellung der HoloLego Applikation durch die HoloLens, alle bereits platzierten Steine angezeigt

seln, um die entsprechende Anweisung zu bekommen. Im Menü kann zwischen verschiedenen Darstellungsoptionen gewählt werden, diese werden in [Unterabschnitt 3.7](#) vorgestellt.

3.3 Aufbau von HoloLego

HoloLego ist in verschiedene Skripte oder Klassen aufgeteilt, die in Unity an Objekte (GameObjects) angehängt sind. Die meisten Skripte besitzen eine `Awake()` oder `Start()` Funktion, sowie eine `Update()` Methode. `Awake()` wird direkt zur Initialisierung aufgerufen, falls das Skript aktiv ist, wird danach einmalig `Start()` ausgeführt. `Update()` wird solange das Skript aktiv ist jeden Frame aufgerufen.

In Unity ist die Szenehierarchie wie in [Abbildung 11](#) aufgebaut. Dabei sind die Namen der Objekte fett gedruckt und die Namen der angehängten Skripte kursiv dargestellt.

3.4 Programmablauf

Der Programmablauf ist in [Abbildung 12](#) zu sehen, dabei wird hauptsächlich auf die für diese Anwendung wichtigen Funktionen eingegangen und weniger auf generelle Funktionen von Unity für das Rendern, verwalten von Events, etc.. Die einzelnen Schritte werden in diesem Abschnitt noch einmal genauer erläutert.

Der **Programmstart** wird einmalig zu Beginn ausgeführt.

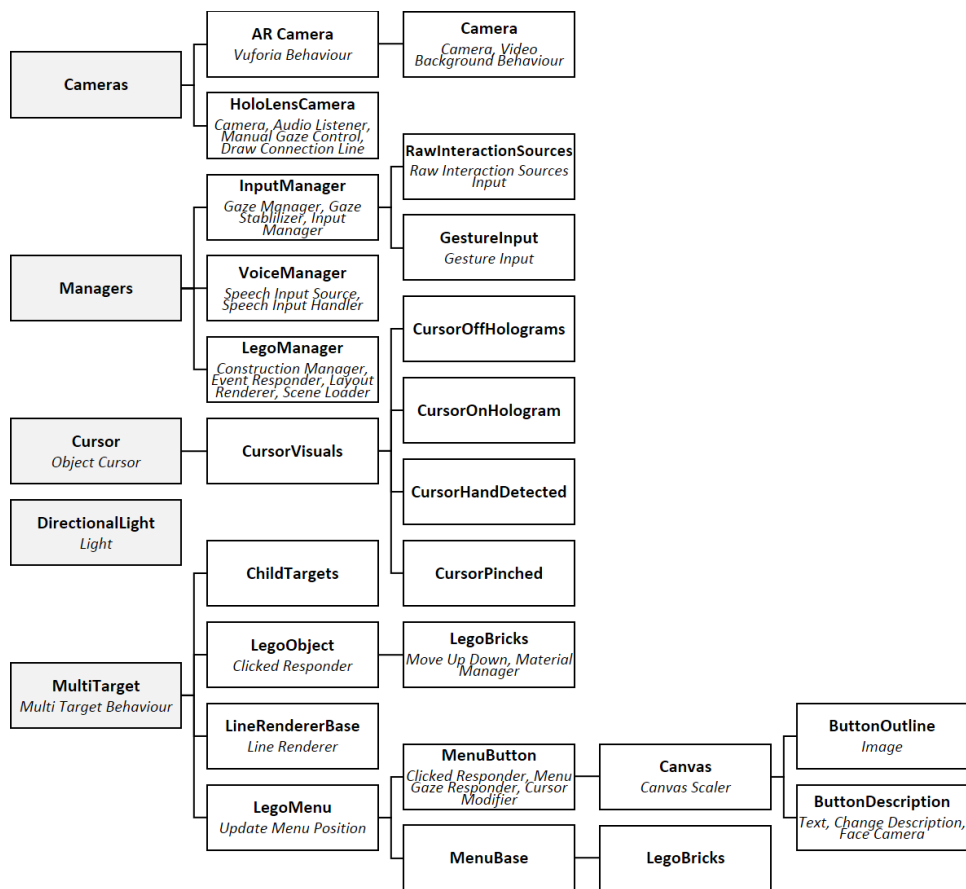


Abbildung 11: Szenenhierarchie von HoloLego

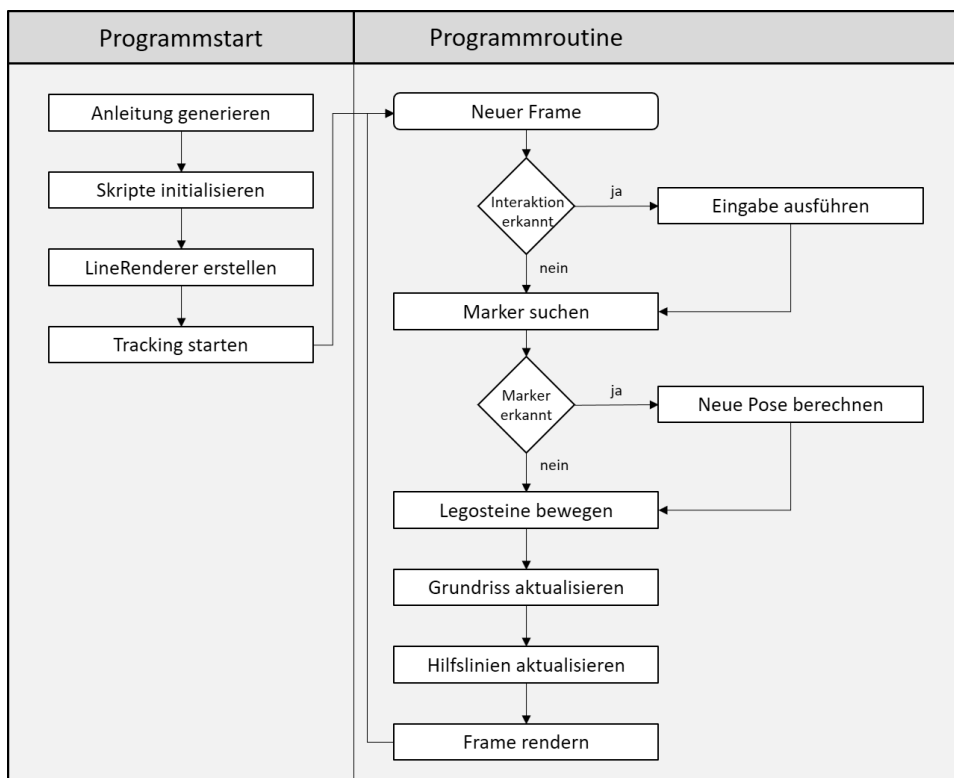


Abbildung 12: Programmablaufplan von HoloLego

- **Anleitung generieren:** Die Reihenfolge für die Anleitung wird generiert. Allen Lego-Objekten werden die benötigten Skripte *Move Up Down* und *Material Manager* zur Bewegung und zum Ändern des Materials angehängt.
- **Skripte initialisieren:** Die HoloToolkit Skripte werden initialisiert, diese sind hauptsächlich für die Interaktion zuständig (siehe [Unterabschnitt 3.5.3](#)).
- **LineRenderer erstellen:** Es wird ein neuer LineRenderer erstellt und initialisiert, dieser zeigt später den auf die Grundfläche projizierten Grundriss des aktuellen Steins an.
- **Tracking starten:** Vuforia Tracking wird initialisiert und die Kamera gestartet.

Die **Programmroutine** wird nach dem Starten und Initialisieren der Anwendung in jedem Frame wiederholt.

- **Eingabe ausführen:** Der InputManager verarbeitet alle aktuellen Interaktionen. Falls eine Eingabe vorliegt, wird diese an das aktuelle Objekt gesendet und ruft dort eine entsprechende Funktion (Callback) auf.
- **Marker suchen:** Vuforia verarbeitet das aktuelle Kamerabild und sucht nach Features, die mit dem ImageTarget übereinstimmen und bestimmt, falls gefunden, die neue Pose des virtuellen Markers.
- **Legosteine bewegen:** *Move Up Down* aktualisiert die Position der Legosteine abhängig von ihrer aktuellen Bewegung.
- **Grundriss aktualisieren:** *Line Renderer* berechnet die aktuelle Position des Grundrisses und stellt diesen dar.
- **Hilfslinien aktualisieren:** *Draw Connection Lines* berechnet und rendert, falls aktiviert, zusätzliche Hilfslinien.
- **Frame rendern:** Der aktuelle Frame wird mithilfe von Unity gerendert und auf der HoloLens dargestellt.

3.5 Vorstellung wichtiger Skripte

Das wichtigste selbstgeschriebene Skript ist der *Construction Manager*, der mit den meisten anderen Skripten interagiert oder von diesen verwendet wird. Andere wichtige, aber kleinere Skripte für die verschiedenen Darstellungsoptionen sind z.B. der *Material Manager*, *Layout Renderer* oder *Draw Connection Lines*. Ansonsten sind Skripte aus HoloToolkit wichtig, wie zum Beispiel *Input Manager* und generell Skripte für Blick- und Cursorsteuerung, sowie die Spracherkennung.

3.5.1 Construction Manager

Dieses Skript wird an das Objekt `LegoManager` angehängt. Damit man die Skripte nicht an das Legomodell anhängen muss und dieses bequemer austauschen kann, enthält der `LegoManager` alle wichtigen Skripte für die Verwaltung und Steuerung der Montage.

Construction Manager enthält Funktionen, um die Steine des Lego-Objektes zu sortieren, durch die Anleitung zu navigieren und generelle Funktionen, die auf alle Objekte angewendet werden sollen, wie zum Beispiel die Materialien neu zu setzen oder die Anwendung zurückzusetzen.

Beim Start der Anwendung wird `Awake()` aufgerufen und es wird eine Liste mit Referenzen zu allen Legosteinen erstellt, um später einfach auf diese zugreifen zu können. Die Steine werden dann in eine für die Anleitung sinnvolle Reihenfolge gebracht und dazu zuerst nach den `y`-Werten ihrer Position in Ebenen gruppiert.

```
1 void PutInLayer(GameObject brick) {
2     foreach (Layer l in layers) {
3         double yDiff = Mathf.Abs((float)((l.y /
4             brick.transform.localPosition.y) - 1));
5         if (yDiff < layerThreshold) {
6             l.bricks.Add(brick);
7             return;
8         }
9     }
10 }
```

Algorithmus 1: Sortieren der Objekte in Ebenen

Für jeden Stein wird dabei `PutInLayer()` aufgerufen (siehe [Algorithmus 1](#)). Es wird für alle bereits existierenden Ebenen überprüft, ob die `y`-Werte (Höhenwerte) ähnlich zu denen einer schon existierenden Ebene sind, falls ja wird dieser Stein der Ebene hinzugefügt. Falls keine Ebene existiert, die ähnlich genug ist, wird eine neue Ebene mit diesem Stein erstellt.

Anschließend werden die Steine in den Ebenen nach ihren `x`-Werten und die Ebenen nach ihren `y`-Werten sortiert. Dadurch wird eine Reihenfolge der Steine von unten nach oben und von rechts nach links bestimmt, die für die Anleitung verwendet wird.

Während über alle Objekte iteriert wird, werden den Objekten auch die Skripte *Material Manager* und *Move Up Down* angehängt.

Move Up Down steuert dabei die Bewegung des Steines der platziert werden soll. Die ursprüngliche Position wird gespeichert und kann bei

Bedarf wiederhergestellt werden. Um eine weiche Bewegung zu erzielen, wird der Sinus zum Berechnen der jeweils neuen Position verwendet.

$$\vec{pos} = \begin{pmatrix} x_{start} \\ y_{start} \\ z_{start} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} * [0,5 * amplitude * (1 + \sin angle)]$$

Der Start-Vektor ist dabei die Ursprungsposition des Steines. Die Amplitude lässt sich einstellen und bestimmt die Höhe der Schwingung. Der Winkel für den Sinus wird jeden Frame abhängig von der Framedauer erhöht.

Material Manager speichert die Originalmaterialien, setzt neue und stellt bei Bedarf die ursprünglichen Materialien wieder her. Die Materialien werden verwendet, um für den alternativen Anzeigemodus Objekte unsichtbar darzustellen. Dazu wird die Tiefenmaske als Material verwendet.

Eine weitere wichtige Funktion ist `nextStep()`, diese wird aufgerufen um zum nächsten Schritt in der Anleitung zu schalten (siehe [Algorithmus 2](#)). Die Aufgabe ist es, den nächsten Stein zu aktivieren und den vorher aktiven zu deaktivieren. Dabei soll auf die passenden Materialien zur Darstellung geachtet werden. Die Hilfs- und Grundrisslinien werden automatisch immer für den jeweils aktiven Stein bestimmt und müssen nicht separat aktualisiert werden.

```

1 public void NextStep() {
2     if (step >= 0 && step < GetBrickCount()) {
3         bricks[step].GetComponent<MoveUpDown>().TurnMovementOff();
4         if (onlyShowCurrent) {
5             bricks[step].GetComponent<MaterialManager>(
6                 ).Set(occlusionMaterial);
7         }
8         step++;
9         bricks[step].SetActive(true);
10        bricks[step].GetComponent<MoveUpDown>().TurnMovementOn();
11        bricks[step].GetComponent<MaterialManager>().Restore();
12    }

```

Algorithmus 2: Zum nächsten Schritt wechseln

Falls der aktuelle Schritt gültig ist, also zwischen null und der Anzahl an vorhandenen Steinen, wird die Funktion ausgeführt. Dazu wird durch `TurnMovementOff()` die Bewegung des aktuellen Steins gestoppt und der Stein auf seine ursprüngliche Position zurückgesetzt. Falls im Menü eingestellt wurde, dass nur der aktuelle Stein angezeigt werden soll, wird dessen Material mit `Set()` durch ein Neues ersetzt. Das Material „occlusionMaterial“ nutzt die Tiefenmaske (`DepthMask`) und zeigt den Hintergrund, also

in diesem Fall den bereits platzierten echten Stein, anstelle des Objektes an. Für den neuen Stein werden Renderer und die Animation des Steins aktiviert und dessen Materialien auf die Originalmaterialien zurückgesetzt.

3.5.2 Layout Renderer

Das Skript *Layout Renderer* liefert die Positionsdaten für den Grundriss des jeweils aktuellen Steins, zu sehen in [Abbildung 16](#). Dazu wird bei Programmstart ein neues Objekt mit dem *Line Renderer* Skript von Unity erstellt und initialisiert. Das Objekt stellt zwischen gegebenen Punkten eine Linie dar.

```
1 void Update () {
2     GameObject obj = _constManager.GetCurrent();
3     if (obj == null) {
4         _lineRenderer.enabled = false;
5         return;
6     }
7     Bounds bound =
8         obj.GetComponent<MeshFilter>().mesh.bounds;
9     Vector3 c = bound.center;
10    Vector3 e = bound.extents;
11    Vector3[] points = new Vector3[5];
12    points[0] = new Vector3(c.x + e.x, c.y + e.y, c.z - e.z);
13    points[1] = new Vector3(c.x + e.x, c.y - e.y, c.z - e.z);
14    points[2] = new Vector3(c.x - e.x, c.y - e.y, c.z - e.z);
15    points[3] = new Vector3(c.x - e.x, c.y + e.y, c.z - e.z);
16    obj.transform.localPosition =
17        obj.GetComponent<MoveUpDown>().getStartingPos();
18    for (int i = 0; i < 4; i++) {
19        points[i] = obj.transform.TransformPoint(points[i]);
20    }
21    points[4] = points[0];
22    _lineRenderer.enabled = true;
23    _lineRenderer.SetPositions(points);
24 }
```

Algorithmus 3: Eckpunkte für den Grundriss bestimmen

Jeden Frame wird im *Layout Renderer* `Update()` aufgerufen um die neuen Eckpunkte des Grundrisses zu berechnen (siehe [Algorithmus 3](#)).

Falls kein aktueller Stein existiert wird der Algorithmus nicht ausgeführt und *Line Renderer* deaktiviert. Wenn ein Stein existiert, wird die Bounding Box verwendet, um die Eckpunkte zu bestimmen. Gegeben ist das Zentrum der Bounding Box im lokalen Koordinatensystem, sowie die Ausdehnung der Box in jede Dimension. Daraus werden die Koordinaten aller unteren Ecken berechnet und anschließend in das Weltkoordinatensystem transfor-

miert. Da die Linie geschlossen sein soll, wird der letzte Punkt gleich dem ersten gesetzt. Anschließend wird das Skript aktiviert und bekommt die Punkte übergeben.

3.5.3 HoloToolkit Skripte

Das HoloToolkit ist eine Sammlung an Skripten und Komponenten, um die Entwicklung von Applikationen für die HoloLens zu unterstützen. Die wichtigsten verwendeten Skripte sind für die Interaktion zuständig. Dazu gehören unter anderem *Gaze Manager*, *Input Manager* und *Voice Manager*. Der *Gaze Manager* ermittelt durch Raycasts Objekte, die vom Benutzer angeschaut werden. Die Daten werden vom Zeiger (Cursor) benutzt, um ihn an entsprechender Stelle anzuzeigen und außerdem an den *Input Manager* weitergeleitet.

Der *Input Manager* überwacht alle Eingabequellen (z.B. verschiedene Gesten, Sprachbefehle, Controller) und sendet Eingabe-Events an das gerade fokussierte Objekt, das nun auf die entsprechende Eingabe reagieren kann. Der *Voice Manager* verknüpft Sprachbefehle mit Events. Wenn ein erkannter gesprochener Text mit gespeicherten Sprachbefehlen übereinstimmt, wird der entsprechende Befehl ausgeführt (z.B. "next", um zum nächsten Schritt zu schalten).

3.6 Tracking

Zum Tracking wird Vuforia verwendet. Zu Beginn wurde als Marker eine zweidimensionale Textur in Form der Straße auf der Legoplatte verwendet, was allerdings häufig zu einer falschen Positionierung führte. Der Marker wurde dabei oft ein bis zwei Zentimeter höher oder niedriger als dessen eigentliche Position erkannt, wodurch das Hologramm nur aus einem bestimmten Blickwinkel übereinstimmte. Aus anderen Richtungen betrachtet verschob sich das Hologramm relativ zum echten Objekt und das Aufbauen wurde so erschwert.

Daher wird anstelle eines einzelnen Markers ein Würfel⁹ (Abbildung 13a) verwendet, der an allen Seiten ein Tracking-Bild bietet und so zu einem stabileren und genaueren Tracking führen soll. Dazu wurde für jede Seite des Würfels ein Marker mit ARMaker¹⁰ generiert, der für Vuforia genug Featurepunkte (siehe Abbildung 13b) bietet. Der Würfel hat die Maße 4,8cm x 5cm x 4,8cm (Breite, Höhe, Länge). Die genauen Algorithmen, die Vuforia für das Tracking verwendet sind nicht veröffentlicht. Die Marker Tracking Pipeline ist aber nach folgendem Prinzip aufgebaut [11]:

- Kamerabild aufnehmen

⁹Marker Download unter <https://goo.gl/9taLzN>

¹⁰<https://github.com/shawnlehner/ARMaker>

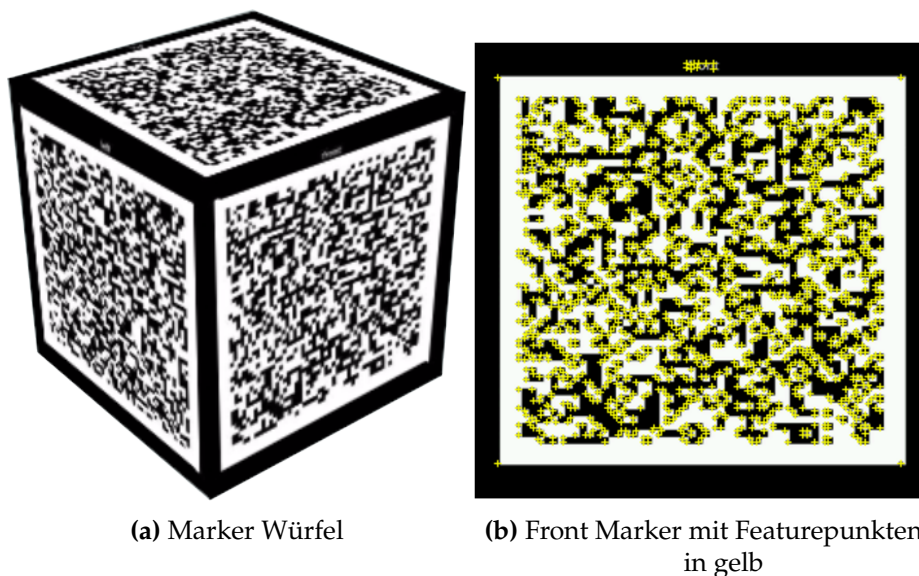


Abbildung 13: Für das Tracking verwendete Marker

- Marker detektieren, indem nach bekannten Formen und Features gesucht wird
- Pose schätzen durch Homographie
- Pose verbessern durch Optimierung und Fehlerminimierung
- Rendern

In Unity sind die wichtigen Skripte *Vuforia Behaviour*, das dem Game-Object ARCamera angehängt ist, und *Multi Target Behaviour* des MultiTarget Objektes. Dem Objekt sind das Legomodell und das Menü als Kinder zugeordnet, wodurch diese bei Positionsänderungen des Markers mit transformiert werden. Die Position und Orientierung der Kamera der HoloLens wird als Weltmittelpunkt zur Berechnung der Pose des Markers genutzt.

3.7 Benutzerinterface und Interaktion

Dargestellt wird das Legomodell, ein Cursor und das Menü. Das Menü ist grafisch auch aus Lego aufgebaut, allerdings nur digital vorhanden und orientiert sich an der Pose des Legomodells. Das Interface und die verschiedenen Darstellungsoptionen für das Legomodell werden in diesem Abschnitt behandelt.

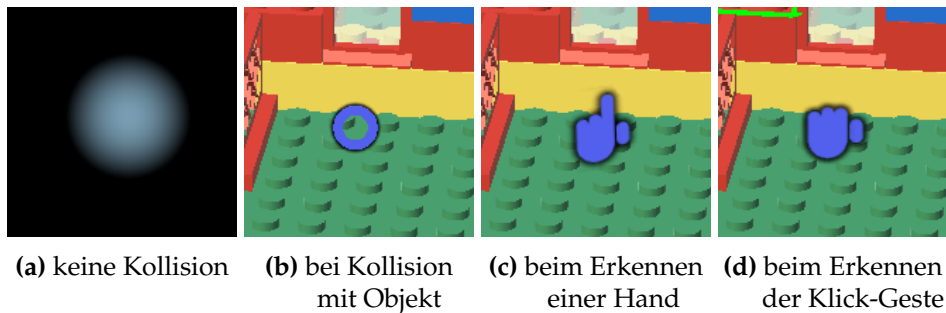


Abbildung 14: Überblick über die verschiedenen Cursor

3.7.1 Cursor

Zur Interaktion wird ein Cursor verwendet, der den Kopfbewegungen folgt und mittig im Sichtfeld positioniert ist. Der Cursor gibt dem Nutzer durch vier verschiedene Modelle (siehe [Abbildung 14](#)) Feedback zu seinen Aktionen. Durch das Skript *GazeManager* werden in jedem Frame Raycasts in Blickrichtung gesendet, um zu prüfen ob aktuell ein Objekt angeschaut wird. Bei Kollision mit einem Objekt wird diesem ein Event gesendet und Skripte können bei Bedarf darauf reagieren, wie es zum Beispiel bei den Buttons vom Menü der Fall ist. Außerdem wird der Cursor an dem aktiven Objekt platziert.

Die folgenden unterschiedlichen Zeiger können dargestellt werden:

Der Cursor in [Abbildung 14a](#) wird in einer festen Entfernung angezeigt falls kein Objekt angeschaut wird, also der Raycast nicht kollidiert. Falls ein Objekt getroffen wird, wird der Kreis aus [Abbildung 14b](#) auf dem entsprechenden Objekt und in Orientierung zur Kamera angezeigt. Die Hand aus [Abbildung 14c](#) ist zu sehen, falls die Hand des Anwenders vor der HoloLens erkannt wird. Wenn die Klick-Geste ausgeführt wird, wird als Feedback kurz die Hand mit angelegten Fingern aus [Abbildung 14d](#) angezeigt. Für den Cursor wurde ein Prefab aus HoloToolkit verwendet und um die beiden letzten Zustände erweitert.

3.7.2 Menü

Das Menü wird zum Start automatisch am Legomodell ausgerichtet. Es ist in der Mitte der rechten Seite neben dem Modell platziert und kann sich auf Wunsch mit dem Anwender um das virtuelle Modell bewegen. Das Menü bietet folgende Buttons zur Interaktion:

- nächster Schritt
- vorheriger Schritt
- nur aktuellen Stein / alle Steine anzeigen

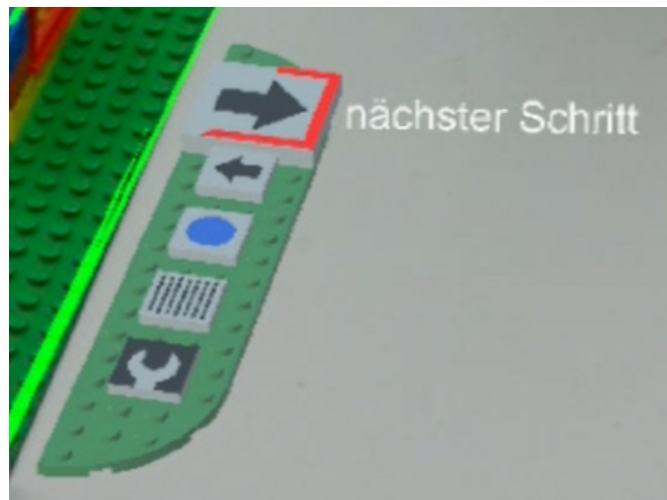


Abbildung 15: Interaktives Menü, dargestellt mit der HoloLens

- Hilfslinien ein-/ausblenden
- erweitertes Menü
 - Zurücksetzen
 - Szene wechseln
 - Tracking ein-/ausschalten
 - von Hand platzieren

Das Menü ist in [Abbildung 15](#) zu sehen. Dort ist das erweiterte Menü eingeklappt und der Button "nächster Schritt" fokussiert. Durch Anvisieren des gewünschten Buttons mit dem Cursor wird `OnFocusEnter()` von *Menu Gaze Responder* aufgerufen, wodurch der Button skaliert und dessen Beschriftung zum besseren Verständnis angezeigt wird. Die Normale der Beschriftung zeigt dabei zur Kamera, damit der Text immer gut lesbar ist. Nach einer kurzen Verzögerung füllt sich ein roter Balken entlang des Randes, ist dieser komplett gefüllt wird das Klick-Event des jeweiligen Menüpunktes ausgelöst und die entsprechenden Funktionen ausgeführt. Alternativ kann das Event per Geste auch direkt gestartet werden. Durch weiteres Anvisieren nach dem ersten Auslösen werden die nachfolgenden Events schneller ausgelöst, bis eine maximale Geschwindigkeit erreicht ist. Um zum nächsten Schritt zu wechseln, kann auch das komplette Legomodell als Button verwendet und per Geste ausgelöst werden. Als Geste wird eine der von Microsoft eingebauten Standardgesten verwendet. Für die Klick-Geste hält der Anwender seine Hand in etwa 20-30 cm Entfernung vor sich und tippt Zeigefinger und Daumen zusammen.

3.7.3 Sprachsteuerung

Alternativ zum Menü kann die Anwendung größtenteils auch mit Sprachbefehlen gesteuert werden. Hier werden Skripte von Microsofts HoloToolkit verwendet. Das GameObject VoiceManager enthält die dazu nötigen Skripte *Speech Input Source* und *Speech Input Handler*. Das Skript *Dictation Input Manager* konvertiert die vom Mikrofon aufgenommene Sprache in Text, der mit definierten Schlüsselwörtern verglichen wird. Stimmt der erkannte Text mit einem definierten Wort oder einer Sequenz überein, werden die entsprechenden Funktionen ausgeführt.

Definiert sind folgende Wörter und Funktionen:

- **next**: nächster Schritt
- **back**: vorheriger Schritt
- **toggle material**: nur aktuellen Stein / alle Steine anzeigen
- **toggle line**: Hilfslinien ein-/ausblenden
- **reset**: Szene zurücksetzen
- **load main/demo scene**: lädt die entsprechende Szene

3.7.4 Darstellung der Platzierung der Legosteine

HoloLego bietet mehrere verschiedene Anzeigemöglichkeiten, die dabei helfen sollen, die korrekte Position zum Anbringen der Steine zu finden. Viele der Einblendungen sind optional, so dass jeder Nutzer das Programm nach seinen Wünschen einstellen kann.

Der Stein des aktuellen Schrittes wird durch *Move Up Down* animiert dargestellt und bewegt sich an der Stelle wo er platziert werden soll auf und ab. An der richtigen Position wird der Grundriss des Steins durch das Skript *Layout Renderer* (siehe [Unterunterabschnitt 3.5.2](#)) angezeigt.

Auf Wunsch können durch *Draw Connection Lines* zusätzliche Linien zur Hilfe eingeblendet werden. Dabei besteht die Auswahl zwischen einer Hilfslinie im Zentrum des aktuellen Steins ([Abbildung 16b](#)) oder in den vier Ecken ([Abbildung 16c](#)).

Je nach Einstellung werden neben dem aktuellen Stein auch alle vorher schon platzierten Steine angezeigt ([Abbildung 10](#)) oder nur der momentan aktive Schritt ([Abbildung 17](#)). Auch wenn nur der aktive Stein angezeigt wird, werden virtuelle Objekte dennoch von bereits platzierten realen Steinen perspektivisch verdeckt. Um das zu erreichen, wird die Tiefenmaske als Material verwendet, wodurch die Objekte unsichtbar werden und den Hintergrund anzeigen.

In [Abbildung 16](#) ist das zum Beispiel am Grundriss zu erkennen, der durch das reale Lego-Fenster verdeckt wird. Dadurch soll verhindert werden,

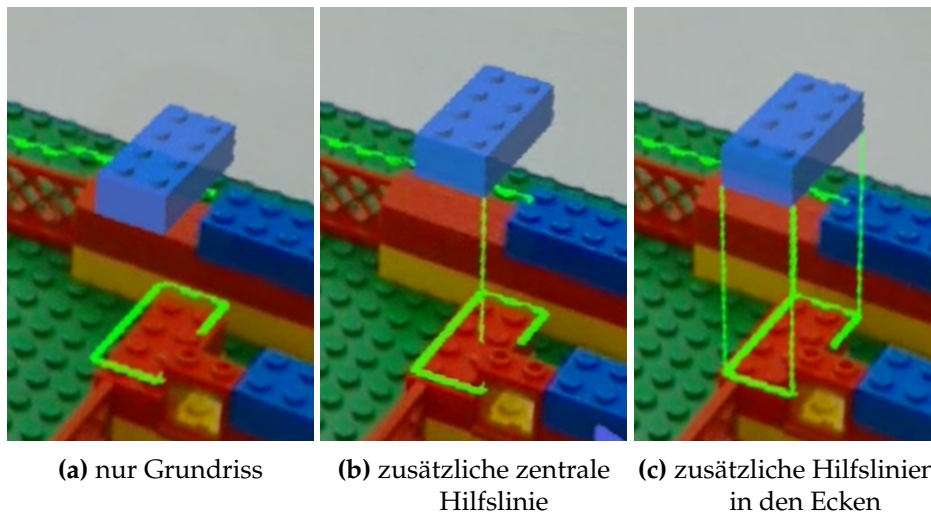


Abbildung 16: Überblick über die verschiedenen Hilfslinien, dargestellt mit der HoloLens

dass Anwender durch fehlende Verdeckung verwirrt werden, wie es bei Nutzertests von Blattgerste et al. [6] bei einigen Personen vorkam. Außerdem wird die Umrandung der Grundplatte angezeigt damit erkennbar ist, falls das Hologramm nicht mehr synchron mit dem realen Objekt ist.

3.8 Verwendung eigener Legomodelle

Die Anwendung ist dynamisch und funktioniert mit jedem in Ebenen aufgebauten Legomodell. Das bedeutet, dass das Legomodell von unten nach oben aufgebaut sein muss. Es können keine Teile von der Seite platziert werden oder Teilmodelle verwendet werden.

Ein passendes Objekt kann in Unity importiert werden, z.B. im fbx-Format. Es muss als Kind des MultiTargets gesetzt werden und so platziert werden, dass die virtuelle Position des Trackers mit der echten übereinstimmt. Anschließend muss im LegoManager als Legomodell das neu hinzugefügte Objekt ausgewählt werden.

Falls noch kein Modell im passenden Format zur Verfügung steht, kann es z.B. mit LDCad gebaut und anschließend mit dem in [Unterabschnitt 3.1](#) angegebenen Blender Plugin in eine fbx-Datei konvertiert werden. Der verwendete Marker ist in [Unterabschnitt 3.6](#) beschrieben.

3.9 Nicht implementierte Funktionen

Die Funktionen in diesem Kapitel waren zur Implementierung geplant, wurden allerdings nicht umgesetzt.

Es sollte erkannt werden, wenn der aktuelle Stein korrekt platziert wurde

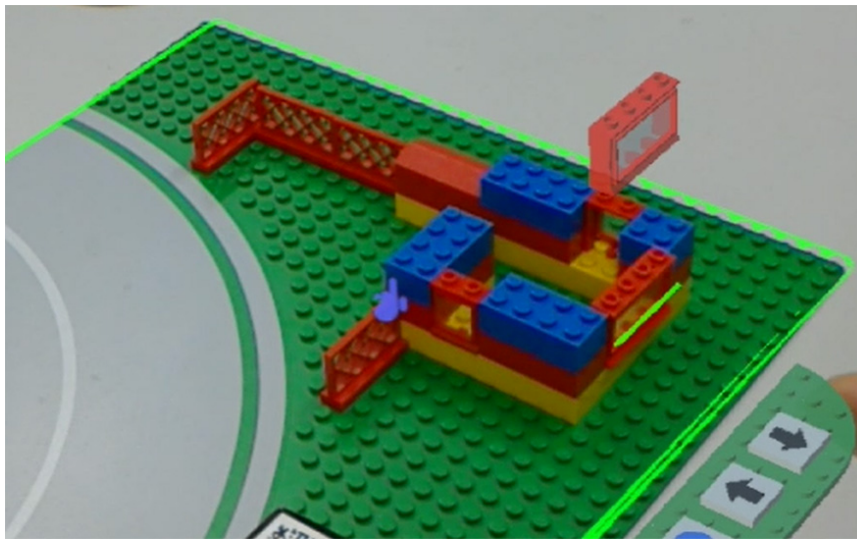


Abbildung 17: Darstellung der HoloLego Applikation durch die HoloLens, nur aktueller Stein wird angezeigt

und daraufhin automatisch zum nächsten Schritt gewechselt werden. Falls der Stein falsch platziert wurde oder generell ein falscher Stein ausgewählt wurde (z.B. falsche Farbe oder Größe), sollte eine Rückmeldung gegeben werden. Um dies umzusetzen, sollte unter anderem der Tiefensensor der HoloLens verwendet werden, weil die Legosteine meist einfarbig sind, nur wenige Featurepunkte bieten und sich dadurch ausschließlich mit RGB-Bildern schwer erkennen lassen. Der Tiefensensor der HoloLens ist für diesen Anwendungsfall allerdings nicht geeignet, da dessen Genauigkeit mit ca. 10cm zu gering ist und auch erst ab einer Distanz von 85cm gemessen werden kann. Die Legosteine sind also zu klein, um im Tiefenbild erkannt zu werden und die Minimaldistanz für die Montage von Lego zu groß. Außerdem bietet Microsoft keine Möglichkeit, direkt auf die Tiefenwertbilder zuzugreifen, sondern nur auf ein bereits weiterverarbeitetes Mesh der Umgebung.

Ähnliche Szenarien verwendeten die Microsoft Kinect zur Unterstützung der HoloLens, wodurch allerdings auch viele Vorteile der HoloLens wegfallen. Es muss ein spezieller Arbeitsplatz existieren, an dem die Kinect aufgebaut und mit einem Computer verbunden ist. Außerdem muss die Kinect mit der HoloLens zusammen kalibriert werden, während man ansonsten nur die HoloLens aufsetzen muss und mit dem Zusammenbau starten kann.

4 Evaluation

Um zu überprüfen, wie gut die HoloLens und die entwickelte Anwendung zur Montage von Lego geeignet ist, wurde ein Nutzertest durchgeführt. Dieser Test sollte klären, wie die augmentierten Anweisungen der HoloLens im Vergleich zu der Papieranleitung abschneiden und wie hilfreich die HoloLens dabei ist. Außerdem sollte überprüft werden, wie gut die Darstellung in der Applikation ist und ob diese dabei hilft, die Aufgabe zu erfüllen. In den folgenden Abschnitten wird der Aufbau, die Durchführung und die Ergebnisse dieser Umfrage vorgestellt und besprochen.

4.1 Aufbau und Gestaltung der Umfrage

4.1.1 Szenario

Der Ablauf des Nutzertests orientiert sich grob am von Funk et al. [15] entwickelten Benchmark.

Nach einer kurzen Vorstellung und Erklärung des Themas dieser Arbeit und des Nutzertests wird das Lego Haus einmal mit Hilfe der HoloLens und einmal mit Papieranleitung zusammengebaut. Bei beiden Tests wird das gleiche Modell eines Hauses verwendet und die einzelnen Legosteine liegen unsortiert neben der Grundplatte. Da bei mehrfachem Aufbau desselben Modells ein Lerneffekt zu erwarten ist, verwendet die Hälfte der Teilnehmer die HoloLens im ersten Durchlauf und die andere Hälfte die Papieranleitung. Die jeweils andere Aufbaumethode wird im zweiten Durchlauf verwendet.

Die Verwendung und Bedienung der HoloLens und der Applikation, sowie die verschiedenen Darstellungsoptionen werden anhand einer Demo-Szene erklärt und von den Teilnehmern ausprobiert. Nachdem die Funktion und Bedienung der Anwendung verstanden ist, beginnt der Test mit der Haupt-Szene. Beim Aufbau wird für beide Methoden die Zeit gestoppt. Dabei werden für jeden Schritt drei Zeiten erfasst:

- t_{locate_pic} : Zeit den passenden Stein zu finden und zu greifen
- $t_{locate_assemble}$: Zeit die richtige Stelle zu finden und den Stein zu platzieren
- t_{switch} : Zeit um zum nächsten Schritt zu schalten

Die Gesamtzeit pro Schritt t_{total} ergibt sich durch Addition der drei Teilzeiten.

$$t_{total} = t_{locate_pic} + t_{locate_assemble} + t_{switch}$$

Nach Aufbau des Legomodells mit beiden Methoden wird ein Fragebogen ausgefüllt, der die Teilnehmer neben generellen Fragen zur Person und den

Erfahrungen mit AR hauptsächlich nach Bewertungen für den Aufbau mit der HoloLens und mit Papier, sowie nach den verschiedenen Darstellungsoptionen befragt. Bei vielen Fragen soll eine Bewertung anhand einer Skala gegeben werden. Es wird immer eine Skala von eins bis sechs verwendet, was die Tendenz die mittlere Antwort zu wählen verhindern soll.

4.1.2 Durchführung

Die Applikation wurde insgesamt an zehn Personen getestet. Die Teilnehmer waren zwischen 22 und 30 Jahre alt ($\bar{x} = 24,7$; $\sigma = 2,7$) und haben zu 60% schon vorher zumindest einmal eine AR-Brille ausprobiert.

Fünf Teilnehmer bekamen zuerst die HoloLens als Anleitung zum Zusammenbau, die anderen Fünf zuerst die Papieranleitung. Die jeweils andere Art der Anleitung wurde als zweites getestet. Anschließend füllten die Teilnehmer einen Fragebogen aus.

4.2 Ergebnisse

In diesem Abschnitt werden die Ergebnisse für die verschiedenen Teilbereiche des Tests und Fragebogens vorgestellt. Dabei geht es zum einen um die Frage, wie die Anwendung im Vergleich zum Papier abschneidet und wie man die Interaktion und Darstellung in der Anwendung am Besten realisiert.

4.2.1 Bearbeitungszeiten

Für den gesamten Aufbau des Legomodells mit der HoloLens benötigten die Testpersonen durchschnittlich 5 Minuten und 38 Sekunden ($\bar{x} = 5,6$; $\sigma = 0,9$). Mit der Papieranleitung wurde das Haus etwas schneller in durchschnittlich 5 Minuten und 12 Sekunden ($\bar{x} = 5,2$; $\sigma = 1,1$) aufgebaut. Die Zeiten wurden per t-Test miteinander verglichen, ein statistisch signifikanter Unterschied in der Aufbautdauer konnte allerdings nicht festgestellt werden. Dies liegt zum einen an den ähnlichen Zeiten, zum anderen an der geringen Zahl an Tests. In [Abbildung 18](#) ist eine Boxplot-Darstellung der Gesamtzeit zu sehen. Daran lässt sich erkennen, dass die Streuung der Zeiten beim Papier größer ist, der Mittelwert und Median allerdings ähnlich sind.

Die folgenden Zeiten für die (Teil-)Schritte sind als α -getrimmte Mittelwerte $\bar{x}_{i;\alpha}$ mit $\alpha = 0,2$ angegeben, das heißt, dass die kleinsten und größten 20% der Daten nicht mit eingerechnet werden. Dies soll den Einfluss von beim Messen entstandenen Ausreißern verhindern.

Um nach dem Wechsel zum nächsten Schritt den passenden Stein zu finden

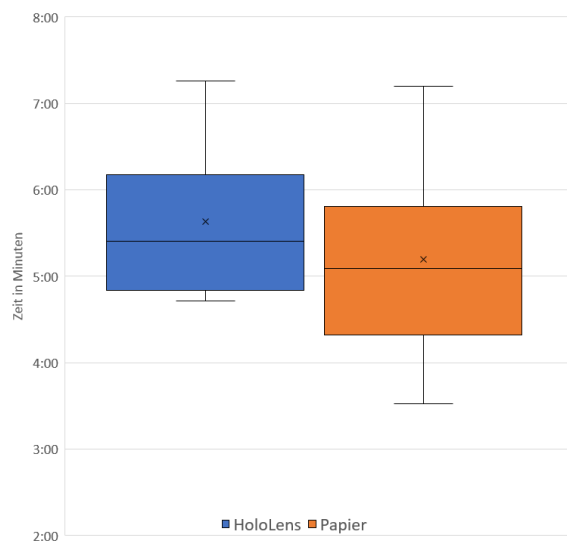


Abbildung 18: Gesamtzeit für den Aufbau

und zu greifen (t_{locate_pic}) benötigten die Teilnehmer am kürzesten mit der HoloLens mit durchschnittlich $2,8s$ ($\sigma = 1,0$) und mit der Papieranleitung mit $2,9s$ ($\sigma = 1,1$) etwas länger.

Das Erkennen der Position und das Platzieren des Legosteins ($t_{locate_assemble}$) dauerte mit der HoloLens $2,4s$ ($\sigma = 0,8$) und damit etwas länger als die $2,3$ ($\sigma = 0,7$) Sekunden, die Teilnehmer mit der Papieranleitung brauchten.

Für das Umschalten zum nächsten Schritt (t_{switch}), bzw. Umblättern der Seite, wurden mit der HoloLens $2,2s$ ($\sigma = 1,0$) und mit Papier $1,9s$ ($\sigma = 0,6$) benötigt.

Insgesamt war der Aufbau mit Hilfe der Papieranleitung etwas schneller und dauerte pro Schritt durchschnittlich $8,3s$ ($\sigma = 2,6$), mit der HoloLens benötigten Teilnehmer $8,6s$ ($\sigma = 2,3$). In [Abbildung 19](#) sind die Zeiten pro Teilschritt zum einfachen Vergleich in einem Kastendiagramm dargestellt. Dabei steht blau für die HoloLens und orange für die Papieranleitung.

Alle Zeiten wurden mittels Zweistichproben-t-Test miteinander verglichen. Ein signifikanter Unterschied zwischen den beiden Aufbaumethoden ist nur bei t_{switch} gegeben, das Wechseln zum nächsten Schritt benötigt mit der HoloLens also mehr Zeit. Um eine relevante Aussage für die anderen Teilschritte machen zu können, sind mehr Tests notwendig.

4.2.2 Ergebnisse des Fragebogens

Der von den Teilnehmern ausgefüllte Fragebogen lieferte die folgenden Ergebnisse.

Die Teilnehmer wurden befragt, ob bei jedem Schritt eindeutig erkennbar war, welchen Stein (welche Farbe, Größe, etc.) man platzieren soll. Die

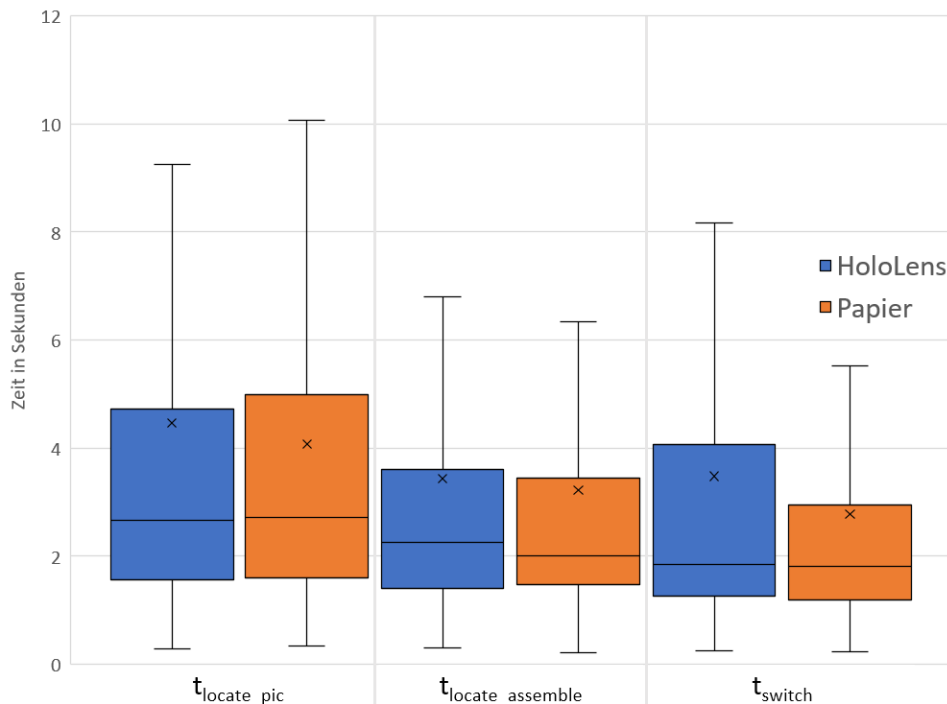


Abbildung 19: Zeiten für die verschiedenen Schritte im Vergleich

Antwort wurde auf einer Skala von eins bis sechs (nie bis immer) angegeben. Die HoloLens wurde mit durchschnittlich 5,6 ($\sigma = 0,5$) bewertet, die Papieranleitung etwas besser mit 5,7 ($\sigma = 0,5$). In dieser Frage schnitten beide Methoden sehr gut ab.

Die Frage, ob es eindeutig war, wo man den jeweiligen Stein platzieren soll, wurde mit der gleichen Skala gestellt und für die HoloLens mit $\bar{x} = 4,5$ ($\sigma = 0,7$) und für Papier mit $\bar{x} = 5,1$ ($\sigma = 0,8$) bewertet. Hier schnitt die HoloLens signifikant schlechter ab als die Papieranleitung.

Die Teilnehmer wurden außerdem gebeten zu bewerten, wie (geistig) anstrengend der Aufbau mit der jeweiligen Methode war. Die Antwort wurde wieder auf einer Skala von eins bis sechs (gar nicht bis sehr anstrengend) gegeben. Die HoloLens wurde dabei mit $\bar{x} = 1,6$ ($\sigma = 0,9$) besser bewertet als die Papieranleitung mit $\bar{x} = 2,4$ ($\sigma = 1,5$).

Um herauszufinden, wie gut Anwender mit der Bedienung und Interaktion der Anwendung klar kommen, sollte auf einer Skala (gar nicht bis sehr gut) bewertet werden, wie der Teilnehmer mit der Bedienung der HoloLens insgesamt klar kommt. Die Frage wurde mit $\bar{x} = 4,6$ ($\sigma = 0,8$) bewertet. Die Bedienung der HoloLego Applikation wurde dagegen mit $\bar{x} = 5,1$ ($\sigma = 0,8$) besser bewertet.

Um zum nächsten Schritt zu wechseln wurde hauptsächlich die Sprach-

steuerung verwendet (60%), etwas weniger Teilnehmer bevorzugten die Navigation durch eine Geste auf das Legomodell (30%). Das Menü am Rand wurde für diesen Zweck nur von 10% verwendet.

Als Darstellungsmodus (siehe [Abbildung 10](#) und [Abbildung 17](#)) wurde von 70% bevorzugt, wenn nur der aktuelle Stein eingeblendet wird. 30% bevorzugten es, alle Steine anzeigen zu lassen.

Für die Darstellung der Hilfslinien (siehe [Abbildung 16](#)) bevorzugten 70% vier Linien in den Ecken, 20% verzichteten auf die Einblendung der Linien und 10% ließen eine zentrale Hilfslinie anzeigen.

70% der Testpersonen würden die HoloLens Anwendung beim Aufbau von Lego der Papieranleitung vorziehen, sofern beide Optionen verfügbar sind.

Alle Testpersonen gaben an, dass die Verwendung der Applikation ihnen Spaß gemacht hat. Als Schulnote wurde die Anwendung mit 1,8 mit gut bewertet.

4.3 Auswertung der Ergebnisse

Insgesamt schneidet HoloLego bei dem Nutzertest gut ab.

Zum Aufbauen wird im Vergleich zur Papieranleitung nicht signifikant mehr Zeit benötigt. Bei einem ähnlichen Nutzertests von Blattgerste et al. [6] benötigten Teilnehmer für den Aufbau mit AR-Brillen länger als mit Papieranweisungen.

Nur das Umschalten zum nächsten Schritt benötigte im Test mehr Zeit mit der HoloLens. Die Gründe könnten sein, dass Gesten und Sprachbefehle nicht immer direkt erkannt wurden und die Aktivierung des Menüs durch Anschauen eine bestimmte Zeit zum Auslösen benötigt. Es ist außerdem zu erwarten, dass sich t_{switch} bei wiederholtem Verwenden der HoloLens verkürzen würde, da die Teilnehmer die Verwendung der Brille weniger gewohnt sind als das Umblättern von Papier.

Die meisten Teilnehmer hatten keine Probleme zu erkennen, welcher Stein aktuell platziert werden sollte. Wo der Legosteine platziert werden soll, war dagegen nicht immer eindeutig, wurde aber noch gut bewertet. Das schlechtere Ergebnis in dieser Frage lässt sich auf die teilweise einige Millimeter versetzte Überlagerung der Hologramme mit den echten Objekten zurückführen, dies sollte in Zukunft korrigiert werden. Trotz dieser Einschränkung wurden beim Aufbau mit der HoloLens nur wenige Fehler gemacht. Hauptsächlich der erste Schritt bereitete Schwierigkeiten, da noch keine Orientierung an bereits platzierten Steinen möglich war. Der Stein wurde teilweise um eine Reihe versetzt platziert. Einige Teilnehmer hatten auch Schwierigkeiten, den Überhang der Eckdachsteine korrekt zu erkennen.

Aber auch bei der Montage mit Papieranleitung wurden Fehler gemacht. In der Anleitung wurde die Ansicht rotiert, so dass der jeweils aktuelle Stein nicht verdeckt wird. Mit dieser Rotation in der Anleitung hatten einige

Testpersonen Schwierigkeiten.

Zusätzlich wurde von den Teilnehmern des Tests erwähnt, dass man bei der HoloLens auf den Aufbau fokussiert bleiben kann. Es muss nicht für jeden Schritt der Kopf bewegt werden um auf die Anleitung zu schauen, da die Anweisung direkt in das Sichtfeld eingeblendet wird.

Diese Gründe führten vermutlich dazu, dass die Verwendung der HoloLens im Vergleich zur Papieranleitung als weniger anstrengend empfunden wurde.

Von den meisten Teilnehmern wurde die Anwendung per Sprachbefehl navigiert. Es fiel auf, dass die Bedienung von Personen, die die Sprachsteuerung verwendeten insgesamt besser bewertet wurde, als bei Verwendung der anderen Interaktionsmöglichkeiten.

Zur Darstellung wurde hauptsächlich das Anzeigen des aktuellen Steins mit vier Hilfslinien in den Ecken ausgewählt. Anhand der Ergebnisse lässt sich erkennen, dass Teilnehmer, die sich keine Hilfslinien anzeigen ließen, es schwerer hatten, die korrekten Positionen zu erkennen. Diese Teilnehmer bewerteten die Frage nach der Genauigkeit der angezeigten Steine schlechter. Zwischen den anderen Darstellungsmöglichkeiten fiel kein besonderer Zusammenhang auf. Für genauere Aussagen wäre ein umfangreicherer Test mit mehr Teilnehmern notwendig.

In den Freitextfeldern des Fragebogens und im Gespräch wurde außerdem folgendes Feedback gegeben: Gelobt wurde die Visualisierung, einfache Bedienung und die starke Überlagerung der Hologramme. Diese empfanden einige Nutzer auch als zu stark, da sie die eigentlichen Objekte nicht mehr richtig erkennen konnten. An der HoloLens lässt sich die Helligkeit des Bildes und damit die Überlagerungsstärke allerdings einstellen.

Kritisiert wurde dagegen von vielen Teilnehmern das geringe Sichtfeld der Brille und von einigen, dass die Brille relativ schwer und mit der Zeit unbequem sei.

Positiv ist allerdings, dass ein Großteil der Nutzer die HoloLens trotz dieser Einschränkungen bevorzugt verwenden würden und alle Teilnehmer Spaß am Aufbau mit Hilfe der HoloLens hatten.

Insgesamt hat die HoloLego Anwendung im Nutzertest trotz der Schwächen im Tracking gut abgeschnitten. HoloLego kann in der Aufbaudauer mit der Papieranleitung mithalten, bietet dafür aber neue Vorteile und Möglichkeiten. Die von den Teilnehmern gegebene Schulnote von 1,8 spiegelt dies wieder.

5 Ausblick

Obwohl die Unterstützung von Montageszenarien durch Augmented Reality schon heute Verwendung findet und funktioniert, gibt es immer noch viele Verbesserungsmöglichkeiten.

5.1 AR-unterstützte Montage

Momentan wird digital unterstützte Montage hauptsächlich in der Industrie verwendet, da die Anwendungen besser für ein bestimmtes Szenario optimiert werden können und die Geräte noch relativ teuer sind. Dabei werden oft Projektor-basierte Systeme genutzt, da diese an einem fest installierten Arbeitsplatz bereits gut funktionieren und HMDs noch einige Einschränkungen haben. Durch Verbessern der AR-Brillen könnten diese in Zukunft aber auch vermehrt genutzt werden. Die Verbreitung würde auch für den Endverbrauchermarkt neue Möglichkeiten bieten, digital erweiterte Montage zu verwenden, um zum Beispiel beim Zusammenbau von Lego und Möbeln, als alternative Bedienungsanleitung oder bei der Reparatur (z.B. von Fahrzeugen) zu helfen.

5.2 HoloLens

Die Microsoft HoloLens und AR-Brillen im Allgemeinen haben neben dem hohen Preis momentan noch mehrere Schwächen, unter anderem ein kleines Sichtfeld, kurze Akkulaufzeit oder hohes Gewicht. Dies verhindert momentan noch eine längere Verwendung der Brille. Die Einschränkungen sollten in neueren Modellen verbessert werden und können im Gegenzug zu einer größeren Verbreitung der HoloLens führen. Auch der Tiefenwertsensor sollte verbessert werden, da er momentan nur grobe Daten zum Erkennen von Räumen und großen Objekten liefert.

5.3 HoloLego Applikation

Die für diese Arbeit implementierte Anwendung generiert die Reihenfolge der Schritte automatisch und kann Modelle daher nur von unten nach oben aufbauen. Das seitliche Anbringen von Steinen oder das Zusammen- und Einbauen von Teilmodellen ist momentan nicht möglich. Dies könnte verbessert werden, indem die Daten für genauere Schritte und Teilschritte in einer Datei gespeichert werden, die zum Beispiel als XML aufgebaut ist. Auch das Verwenden von Anleitungen anderer Nutzer würde so vereinfacht werden.

Bereits für diese Arbeit war optional angedacht, automatisch zum nächsten Schritt weiter zuschalten und Fehler in der Montage zu erkennen. Dies war aufgrund von Hardwareeinschränkungen noch nicht möglich, wäre

für die Zukunft aber eine wichtige Funktion. Außerdem könnte der jeweils benötigte Stein in der Umgebung erkannt und hervorgehoben werden, so würde auch die Phase, den richtigen Stein zu finden (t_{locate_pic}) beschleunigt werden.

Eine weitere praktische Ergänzung wäre das Laden von neuen Objekten während der Laufzeit, indem diese als fbx-Datei und alternativ auch im LDraw-Format geöffnet werden können.

6 Fazit

Das Ziel der Arbeit war, es eine interaktive und augmentierte Anwendung für die HoloLens zu implementieren und zu evaluieren. Dabei sollten die Möglichkeiten und Grenzen der HoloLens in Bezug auf Montageszenarien bewertet werden, um zu evaluieren, wie hilfreich die HoloLens dabei sein kann.

Es stellte sich heraus, dass die HoloLens schon heute gute Möglichkeiten bietet, bei der Montage zu unterstützen und gegenüber anderen Methoden einige Vorteile bietet. Die HoloLens erlaubt es zum Beispiel, auf die Montage fokussiert zu bleiben.

Im Nutzertest wurde der Aufbau mit der HoloLego Applikation ähnlich schnell erledigt wie mit Papieranleitung. Die Darstellung und Bedienung der Applikation wurde als gut bewertet und der Aufbau war für die Teilnehmer weniger anstrengend als mit der Papieranleitung. Es wurden jedoch auch einige noch existierende Grenzen und Schwächen festgestellt, sowohl software- als auch hardwareseitig. Für die Software muss hauptsächlich das Tracking verbessert werden, das teilweise noch nicht die nötige Genauigkeit erzielen konnte und zu leicht versetzten Hologrammen führte. Weitere Möglichkeiten, die Anwendung zu erweitern, wurden in [Unterabschnitt 5.3](#) vorgestellt. An der HoloLens fiel hauptsächlich das geringe Sichtfeld negativ auf. Für eine Erweiterung der Software wäre ein verbesserter Tiefenwertsensor, bei dem auch direkt auf die Bilder zugegriffen werden kann hilfreich.

Optional sollte die Anwendung automatisch erkennen und verifizieren, ob ein Stein korrekt platziert wurde. Durch die bestehenden technischen Grenzen wurde das optionale Ziel leider nicht erreicht.

Literatur

- [1] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, "<title>Augmented reality: a class of displays on the reality-virtuality continuum</title>," vol. 2351, pp. 282–292, 1995. [Online]. Available: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=981543>
- [2] M. Billinghurst, A. Clark, and G. Lee, "A Survey of Augmented Reality," *Foundations and Trends® in Human–Computer Interaction*, vol. 8, no. 2-3, pp. 73–272, 2015. [Online]. Available: <http://www.nowpublishers.com/article/Details/HCI-049>
- [3] P. Dobrindt, S. Martin, and J. Flemming, "BMW i8 Head-Up Display · FHP," 2015. [Online]. Available: <https://fhp.incom.org/project/5579> (Accessed 2017-10-11).
- [4] Microsoft, "Microsoft HoloLens | The leader in mixed reality technology," 2017. [Online]. Available: <https://www.microsoft.com/en-us/hololens> (Accessed 2017-10-12).
- [5] S. Büttner, H. Mucha, M. Funk, T. Kosch, M. Aehnelt, S. Robert, and C. Röcker, "The Design Space of Augmented and Virtual Reality Applications for Assistive Environments in Manufacturing," *Proceedings of the 10th International Conference on Pervasive Technologies Related to Assistive Environments - PETRA '17*, pp. 433–440, 2017. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3056540.3076193>
- [6] J. Blattgerste, B. Strenge, P. Renner, T. Pfeiffer, and K. Essig, "Comparing Conventional and Augmented Reality Instructions for Manual Assembly Tasks," *Proceedings of the 10th International Conference on Pervasive Technologies Related to Assistive Environments - PETRA '17*, pp. 75–82, 2017. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3056540.3056547>
- [7] S. C.-Y. Yuen, G. Yaoyuneyong, and E. Johnson, "Augmented Reality: An Overview and Five Directions for AR in Education," *Journal of Educational Technology Development and Exchange*, vol. 4, no. 1, pp. 119–140, 2011. [Online]. Available: <http://aquila.usm.edu/jetde/vol4/iss1/11>
- [8] A. Kipman, "Announcing Microsoft HoloLens Development Edition open for pre-order, shipping March 30," 2016. [Online]. Available: <https://blogs.windows.com/devices/2016/02/29/announcing-microsoft-hololens-development-edition-open-for-pre-order-shipping-march-30/> (Accessed 2017-10-12).

- [9] O. Kreylos, "On the road for VR: Microsoft HoloLens at Build 2015, San Francisco," 2015. [Online]. Available: <http://doc-ok.org/?p=1223> (Accessed 2017-12-11).
- [10] Microsoft Dev Center, "Hardware details," p. 1, 2016. [Online]. Available: <https://developer.microsoft.com/en-us/windows/holographic/hardware{ }details{#}power> (Accessed 2017-10-12).
- [11] D. Schmalstieg and T. Hollerer, "Augmented reality: Principles and practice," in *2017 IEEE Virtual Reality (VR)*, 2017, pp. 425–426. [Online]. Available: <http://ieeexplore.ieee.org/document/7892358/>
- [12] inIT - Institut für industrielle Informationstechnik and Hochschule Ostwestfalen-Lippe, "inIT: Bequemer produzieren – durch mobile Projektionen in der industriellen Fertigung," 2015. [Online]. Available: <https://www.hs-owl.de/init/aktuelles/news/news-detail/news/bequemer-produzieren-durch-mobile-projektionen-in-der-industriellen-fertigung.html> (Accessed 2017-10-12).
- [13] A. Nee, S. Ong, G. Chryssolouris, and D. Mourtzis, "Augmented reality applications in design and manufacturing," *CIRP Annals - Manufacturing Technology*, vol. 61, no. 2, pp. 657–679, 2012. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0007850612002090>
- [14] T. Caudell and D. Mizell, "Augmented reality: an application of heads-up display technology to manual manufacturing processes," in *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, 1992, pp. 659–669 vol.2. [Online]. Available: <http://ieeexplore.ieee.org/document/183317/>
- [15] M. Funk, T. Kosch, S. W. Greenwald, and A. Schmidt, "A benchmark for interactive augmented reality instructions for assembly tasks," *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia - MUM '15*, pp. 253–257, 2015. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2836041.2836067>
- [16] P. Mohr, B. Kerbl, M. Donoser, D. Schmalstieg, and D. Kalkofen, "Retargeting Technical Documentation to Augmented Reality," *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, pp. 3337–3346, 2015. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2702123.2702490>
- [17] S. Gauglitz, B. Nuernberger, M. Turk, and T. Höllerer, "In touch with the remote world," *Proceedings of the 20th ACM Symposium on Virtual*

Reality Software and Technology - VRST '14, pp. 197–205, 2014. [Online].
Available: <http://dl.acm.org/citation.cfm?doi=2671015.2671016>

- [18] M. Funk, T. Kosch, and A. Schmidt, “Interactive worker assistance,” *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '16*, pp. 934–939, 2016. [Online].
Available: <http://dl.acm.org/citation.cfm?doi=2971648.2971706>