

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart  
Universitätsstraße 38  
D-70569 Stuttgart

Masterarbeit

**Tauglichkeit von  
Augmented-Reality-Brillen für die  
visuelle Analyse in Produktion  
und Fertigung**

Jan Reinhardt

<b>Studiengang:</b>	Informatik
<b>Prüfer/in:</b>	Prof. Dr. Thomas Ertl
<b>Betreuer/in:</b>	Dipl.-Inf. Dominik Herr, Dr. rer. nat. Guido Reina Dipl.-W. Inf. Nico Orschel (extern)
<b>Beginn am:</b>	1. Oktober 2016
<b>Beendet am:</b>	31. März 2017
<b>CR-Nummer:</b>	H.5.1, I.3.8, J.1



## **Kurzfassung**

Die Microsoft HoloLens ist die erste Augmented-Reality-Brille die autark arbeitet. Durch diese Kombination ergeben sich eine große Anzahl an Einsatzmöglichkeiten die zuvor nicht möglich waren. Das Ziel dieser Masterarbeit ist es herauszufinden, ob die Microsoft HoloLens für die visuelle Analyse von Produktions- und Fertigungsanlagen geeignet ist. Dazu werden im ersten Schritt die technischen Möglichkeiten der Microsoft HoloLens analysiert. Im zweiten Schritt wird auf Grundlage dieser Möglichkeiten ein Konzept entwickelt, das zeigt wie Augmented-Reality in Produktion und Fertigung eingesetzt werden kann. Dazu wird ein Fabriklayout-Simulators auf die HoloLens portiert. Neben der direkten Portierung sollen auch Anpassungen durchgeführt werden, um die vollen Möglichkeiten der Augmented-Reality-Brille zu nutzen. Ein besonderes Augenmerk liegt dabei auf der Interaktion und der Visualisierung. Dieses Konzept wird anschließend in Form eines Prototyps für die Microsoft HoloLens implementiert und auf Basis eines Use-Cases evaluiert.

## **Abstract**

The Microsoft HoloLens is the first fully untethered Augmented Reality device. This device opens possibilities for new application which were not possible before. The goal of this master thesis is to find out if the Microsoft HoloLens can be used for Visual Analytics in production and manufacturing. To achieve this goal, the first step is to analyze the technical possibilities of the device. The second step is to create a concept which shows the capabilities of Augmented Reality in production and manufacturing. To do this, a factory layout simulation is ported to the HoloLens. Instead of simply porting the concept, it is altered to make use of the full capabilities of the Augmented Reality technology. Special focus rests on the interaction and the visualization. This concept is then implemented as a prototype and evaluated with a use case.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>9</b>
1.1	Inhalt dieser Arbeit . . . . .	9
1.2	AIT GmbH & Co. KG . . . . .	10
<b>2</b>	<b>Grundlagen</b>	<b>11</b>
2.1	Virtual Reality und Augmented Reality . . . . .	11
2.1.1	Perzeption . . . . .	11
2.1.2	Virtual Reality . . . . .	12
2.1.3	Abgrenzung Augmented Reality zu Virtual Reality . . . . .	12
2.1.4	Binokulare Abbildungen . . . . .	13
2.2	Microsoft HoloLens . . . . .	13
2.2.1	Hardware Spezifikationen . . . . .	14
2.2.2	Designrichtlinien für 3D-Objekte in HoloLens-Anwendungen . . . . .	16
2.2.3	Interaktion mit der HoloLens . . . . .	16
2.3	Markerbasierte Objekterkennung . . . . .	19
2.3.1	Erkennung von Markern . . . . .	19
2.3.2	Natural Feature Tracking (NFT) . . . . .	20
2.4	Visualisierung . . . . .	20
2.4.1	Visualisierungs-Pipeline . . . . .	21
2.4.2	Visuelle Analytik . . . . .	22
2.5	Augmented Reality in Produktion und Fertigung . . . . .	23
2.5.1	Montage und Wartung . . . . .	23
2.5.2	Planung und Design . . . . .	25
<b>3</b>	<b>Technische Möglichkeiten der Microsoft HoloLens</b>	<b>29</b>
3.1	Evaluierung der Grafikleistung . . . . .	29
3.1.1	Testablauf . . . . .	29
3.1.2	Ergebnisse der Messungen . . . . .	31
3.1.3	Fazit . . . . .	34
3.2	Analyse der „Holographic Remoting“ Funktionen . . . . .	34
3.2.1	Qualität des holographischen Remoting . . . . .	35
3.2.2	Holographic Remoting in der Entwicklungsumgebung <i>Unity</i> . . . . .	37
3.2.3	Vorteile und Nachteile des Holographic Remoting . . . . .	37
3.3	Analyse des „Holographic Sharing“ . . . . .	38
3.3.1	Technologie . . . . .	39

3.4	Markerbasierte Objekterkennung mit der Microsoft HoloLens . . . . .	39
3.4.1	Test der HoloLens Kamera . . . . .	40
3.4.2	ARToolkit . . . . .	40
3.4.3	Vuforia . . . . .	43
3.4.4	Vergleich von Vuforia und dem ARToolkit . . . . .	44
3.5	Das dreidimensionale Umgebungsmodell . . . . .	45
3.5.1	Qualität des erstellten Umgebungsmodells . . . . .	46
3.5.2	Nutzung des Umgebungsmodells in Augmented Reality Anwendungen	48
<b>4</b>	<b>Layoutplanung von Produktionsstraßen mit der Microsoft HoloLens</b>	<b>51</b>
4.1	Fabriklayout-Simulator . . . . .	51
4.1.1	Analysieren eines Layouts . . . . .	53
4.1.2	Optimieren eines Layouts . . . . .	53
4.2	Anforderungen an die Produktionsstraßenplanung mit der Microsoft HoloLens	53
4.2.1	Anforderungen des Layout-Simulator . . . . .	54
4.2.2	Anforderungen der Augmented-Reality Anwendung . . . . .	55
4.3	Konzept . . . . .	56
4.3.1	Zustandsautomat . . . . .	56
4.3.2	Feedback beim Interagieren mit der HoloLens . . . . .	58
4.3.3	Platzieren einer Grundplatte . . . . .	59
4.3.4	Skalieren eines Layouts . . . . .	61
4.3.5	Editieren eines Layouts . . . . .	61
4.3.6	Visualisieren von Simulationsergebnissen . . . . .	64
4.3.7	Holographic Sharing in Verbindung mit dem Layout-Simulator . . . . .	68
4.4	Implementierung . . . . .	70
4.4.1	Verwendete Werkzeuge und Bibliotheken . . . . .	70
4.4.2	Verwendung von Unity-Szenen . . . . .	71
4.4.3	Probleme bei der Entwicklung für die HoloLens . . . . .	71
4.5	Evaluation . . . . .	75
4.5.1	Vergleich des Prototyps mit den gestellten Anforderungen . . . . .	75
4.5.2	Demonstration eines Analyseablaufs durch den Simulator . . . . .	76
4.5.3	Anwendung des Prototyps in einer realen Umgebung . . . . .	78
4.5.4	Benutzbarkeit . . . . .	79
4.5.5	Vergleich der HoloLens- und Desktop-Version des Layout-Simulators .	79
<b>5</b>	<b>Zusammenfassung &amp; Fazit</b>	<b>81</b>
5.1	Zusammenfassung . . . . .	81
5.2	Fazit . . . . .	81
<b>6</b>	<b>Ausblick</b>	<b>83</b>
	<b>Literaturverzeichnis</b>	<b>85</b>

# Abbildungsverzeichnis

2.1	Reality-Virtuality-Kontinuum . . . . .	12
2.2	Microsoft HoloLens . . . . .	14
2.3	Verschiedene binokulare Abbildungen . . . . .	15
2.4	Abstand 3D-Objekte . . . . .	16
2.5	Bloom Geste . . . . .	17
2.6	Air-Tap Geste . . . . .	18
2.7	Verschiedene Marker . . . . .	19
2.8	Beispiel für einen Marker . . . . .	21
2.9	Visualisierungs-Pipeline . . . . .	22
2.10	Modell der visuellen Analytik . . . . .	23
2.11	Montageaufgaben mit Hilfe von Augmented Reality . . . . .	24
2.12	Augmented Reality bei der Montage eines Geschützturms . . . . .	25
2.13	Augmented Reality Fabriklayoutplanung . . . . .	26
2.14	Virtuelle Maschine in Fabrikhalle . . . . .	26
2.15	Information an einer Maschine . . . . .	27
2.16	Reale und virtuelle Sortiermaschine . . . . .	28
3.1	Screenshot Grafiktest . . . . .	30
3.2	Ergebnisse des Benchmark-Tests Reine Visualisierung . . . . .	31
3.3	Ergebnisse des Benchmark-Tests Klassisches Einsatzszenario . . . . .	32
3.4	Ergebnisse des Benchmark-Tests Verdeckung durch reale Objekte . . . . .	32
3.5	Ergebnisse des Benchmark-Tests Anzeigen des Dreiecksnetzes . . . . .	33
3.6	Das Holographic Remoting Systems . . . . .	35
3.7	Ergebnisse Benchmark-Tests Holographic Remoting . . . . .	36
3.8	Holographic Sharing . . . . .	40
3.9	Testaufnahme Handy . . . . .	41
3.10	Testaufnahme HoloLens . . . . .	41
3.11	Markererkennung mit ARToolkit . . . . .	43
3.12	Vuforia Natural Marker Erkennung . . . . .	44
3.13	Vuforia VuMark Erkennung . . . . .	45
3.14	Umgebungsmodell eines Stuhls . . . . .	46
3.15	Aufbearbeitetes Umgebungsmodell eines Stuhls . . . . .	46
3.16	Umgebungsmodell eines Tisches . . . . .	47
3.17	Umgebungsmodell einer laufenden Person . . . . .	48

4.1	Realer Aufbau des iTFrame-Systems . . . . .	51
4.2	Fabriklayout-Simulator . . . . .	52
4.3	Zustandsautomat . . . . .	57
4.4	Cursor auf Objekt . . . . .	59
4.5	Cursor Rückmeldung . . . . .	59
4.6	Markieren von Elementen . . . . .	62
4.7	Feedback zur ausgewählten Manipulationsmöglichkeit . . . . .	63
4.8	Snapping beim Verschieben von Elementen . . . . .	64
4.9	Snapping beim Rotieren von Elementen . . . . .	64
4.10	Die Simulationsstatusbox . . . . .	65
4.11	Visualisierung der Auslastung . . . . .	66
4.12	Visualisierung der Wartezeit . . . . .	67
4.13	Detailinformationen in Form von Popups . . . . .	68
4.14	Unterschiedliche Kollisionsstrukturen der Roboterstation . . . . .	73
4.15	Skalierung eines Popups . . . . .	75
4.16	Beschreibung Arbeitsablauf, Bild 1 . . . . .	76
4.17	Beschreibung Arbeitsablauf, Bild 2 . . . . .	77
4.18	Beschreibung Arbeitsablauf, Bild 3 . . . . .	77
4.19	Beschreibung Arbeitsablauf, Bild 4 . . . . .	78



# 1 Einleitung

Nach dem Hype-Zyklus von Gartner [Gar17] wird Virtual Reality in der Zwischenzeit produktiv in der Unterhaltungsindustrie eingesetzt während Augmented Reality noch einige Jahre bis zur Reife benötigen soll. Trotzdem hat Microsoft mit der HoloLens ein verstärktes Interesse um die Augmented-Reality-Technologie ausgelöst. Die HoloLens ist die erste Augmented-Reality-Brille, die autark arbeitet. Es ist möglich, stereoskopische Abbildungen anzuzeigen und mit diesen zu interagieren. Während der klare Fokus der Virtual-Reality-Brillen auf die Unterhaltungsindustrie gerichtet ist, versucht Microsoft mit der HoloLens vor allem Firmen anzusprechen. Die meisten Showcases und Demonstrationen sind in Partnerschaft mit verschiedenen Firmen entstanden und zeigen wie die HoloLens den Arbeitsalltag erleichtern kann, ein Beispiel dafür ist die HoloLens-Demo von Thyssenkrupp [Thy17].

Durch die neue Technologie ergeben sich viele neue Einsatzgebiete in fast allen Bereichen. Auch im industriellen Umfeld ergeben sich Möglichkeiten, zum Beispiel für Montage, Wartung, Reparatur, Planung und Design. Durch das stereographische Display können Maschinen in Modellgröße oder auch in Originalgröße visualisiert werden. Raumbedürfnisse und Arbeitsabläufe lassen sich so ohne großen Kosten oder Zeitaufwand darstellen. Zusätzlich können unterschiedlichste Informationen zum Zustand der Anlage eingeblendet werden.

Das Ziel dieser Masterarbeit ist es herauszufinden, ob die Microsoft HoloLens für die visuelle Analyse von Produktions- und Fertigungsanlagen geeignet ist. Dazu soll ein Konzept entwickelt werden, das Produktionsanlagen darstellt und Interaktionen mit diesen ermöglicht. So soll es möglich sein, einzelne virtuelle Anlagenteile zu verschieben, zu erzeugen oder zu löschen. Dieses Konzept wird in Form eines Prototyps für die HoloLens entwickelt und evaluiert. Ein besonderes Augenmerk liegt dabei auf dem Informationstransport und die intuitive Bedienbarkeit der Anwendung.

Die Arbeit basiert auf dem Fabriklayout-Simulator von Wörner et al. [WE11; WE13].

## 1.1 Inhalt dieser Arbeit

Im ersten Kapitel werden die nötigen Grundlagen der im späteren Verlauf verwendeten Techniken vermittelt. Außerdem wird auf verwandte Arbeiten im Bereich Produktion und Fertigung eingegangen. Im Kapitel *Technische Möglichkeiten der Microsoft HoloLens* werden Informationen

über die Augmented-Reality-Brille gesammelt, um in den folgenden Kapiteln eine Diskussionsgrundlage für die Ausarbeitung eines Konzeptes zu bieten. Das folgende Kapitel bildet den Hauptteil dieser Arbeit. Im Kapitel *Layoutplanung von Produktionsstraßen mit Microsoft HoloLens* werden zuerst verschiedene Konzepte der Interaktion und Visualisierung diskutiert und im zweiten Teil eines dieser Konzepte implementiert. Im dritten Teil soll die prototypische Anwendung auf Informationstransport und Bedienbarkeit evaluiert werden. Die Kapitel Zusammenfassung, Fazit und Ausblick schließen die Arbeit ab.

## 1.2 AIT GmbH & Co. KG

Diese Arbeit ist in Zusammenarbeit mit der Firma AIT GmbH & Co. KG entstanden, welche die Hardware in Form der Microsoft HoloLens gestellt hat. Zusätzlich hat die Firma durch ihr Fachwissen zum Gelingen dieser Arbeit beigetragen. Die Firma AIT aus Stuttgart versteht sich als innovativer Full-Service Lösungspartner, der im Bereich Softwareentwicklung Unternehmen ganzheitlich unterstützt. Kunden sind dabei hauptsächlich mittelständische Maschinenbauer in ganz Deutschland, für die AIT Visualisierungslösungen entwickelt. Als Technologieunternehmen engagiert sich die AIT in wichtigen Zukunftsthemen wie Industrie 4.0, Internet of Things (IoT) und Digitale Transformation. Seit der Veröffentlichung der Microsoft HoloLens gehört auch diese zum Portfolio des Unternehmens [AIT16].

## 2 Grundlagen

In diesem Kapitel werden die zum Verständnis der Arbeit nötigen Grundlagen erklärt. Zu Beginn wird allgemein auf die Virtual- und Augmented-Reality-Technologie eingegangen und im Anschluss die Microsoft HoloLens vorgestellt. Anschließend folgen Abschnitte zur markerbasierten Objekterkennung und zu Visualisierung. Abschließend wird auf verwandte Arbeiten eingegangen.

### 2.1 Virtual Reality und Augmented Reality

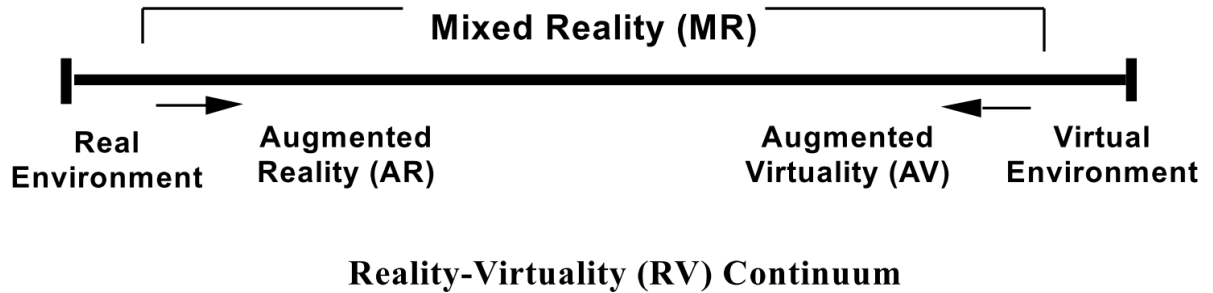
Nach einer Erklärung zur Perzeption der menschlichen Sinne werden in diesem Abschnitt die beiden Schlagworte Virtual Reality und Augmented Reality erklärt und die beiden Begriffe gegeneinander abgegrenzt.

#### 2.1.1 Perzeption

Unser Körper nimmt seine Umgebung über eine Vielzahl von Sinnen auf. Es entsteht daraus ein Eindruck unserer Umgebung. Um eine perfekte virtuelle Realität zu erreichen, müssten alle Sinne getäuscht werden. Dies ist jedoch mit der aktuellen Technik nicht möglich. Einer Einschätzung von Mazuryk und Gervautz[MG96] folgend sind jedoch nicht alle Sinne gleich wichtig:

Sehen	70%
Hören	20%
Riechen	5%
Fühlen	4%
Schmecken	1%

Aus diesem Grund müssen nicht alle Sinne getäuscht werden, um eine glaubhafte virtuelle Realität zu erschaffen. Es wird versucht, nur die wichtigsten Sinne zu täuschen. Dazu gehören vor allem das Sehen und Hören. Das Riechen und Schmecken spielt nur eine geringe Rolle in der Wahrnehmung und kann deshalb meistens ignoriert werden. Neben diesen traditionellen Sinnen existieren auch noch weitere, wie zum Beispiel die Balance, Beschleunigung, Temperatur und Propriozeption (Wahrnehmung von Körperbewegungen) [How15]. Diese sind alle schwer bis unmöglich zu täuschen.



**Abbildung 2.1:** Reality-virtuality-Kontinuum von Milgram et al. [MTUK95]. Augmented Reality befindet sich eher links während Virtual Reality eher im rechten Teil angesiedelt ist.

### 2.1.2 Virtual Reality

Virtual-Reality-Anwendungen versuchen den Benutzer durch eine Täuschung der Sinne in eine virtuelle Welt zu versetzen. Da das Sehen die wichtigste Rolle in der Wahrnehmung spielt, liegt der Hauptfokus von Virtual Reality auf den verwendeten Display-Technologien und Render-Technologien. Um eine möglichst reale Erfahrung zu erreichen, muss ein hochauflösendes Display mit weitem Blickwinkel verwendet werden. Dieses Display sollte eine hohe Bildwiederholungsrate besitzen, da sonst bei Kopfbewegungen die virtuelle Welt zu langsam auf Änderungen der realen Kopfposition reagiert. Für die Sinne wie Balance, Beschleunigung und Propriozeption sollten die Anwendungen so gewählt werden, dass sie nicht getäuscht werden müssen oder nur eine untergeordnete Rolle spielen.

Vor wenigen Jahren waren Virtual-Reality-Technologien nur in Laboren anzufinden. Heute existieren bereits mehrere kommerziell verfügbare Systeme, welche vor allem für die Unterhaltungsindustrie verwendet werden. Anwendungen sind bisher vor allem Spiele oder sogenannte Erfahrungen, bei denen man sich virtuell in sein zukünftiges Auto, an seinen Urlaubsort, usw. versetzen lassen kann.

### 2.1.3 Abgrenzung Augmented Reality zu Virtual Reality

Zum Vergleich der beiden Technologien hilft das Kontinuum von Milgram et al. [MTUK95]. Dies beschreibt ein Spektrum von Anwendungen. Ganz rechts in diesem Spektrum befinden sich rein virtuelle Anwendungen. Ganz links in diesem Spektrum sind rein reale Anwendungen. Da bei Virtual Reality die reale Umgebung komplett ausgeblendet wird und durch eine virtuelle Realität ersetzt wird befinden sich diese Anwendungen weit rechts im Spektrum.

Statt den Benutzer in eine virtuelle Welt zu versetzen wird bei Augmented Reality die reale Welt um virtuelle Elemente angereichert. Diese Anwendungen befinden sich weiter links im Spek-

trum. Je nach Anwendung kann sich die Technologie nach links oder rechts verschieben. Durch die Kombination von Realität und virtueller Welt werden viele Probleme gelöst, die aktuelle Virtual Reality Brillen haben. Ein Vorteil ist, dass viele Sinne bei Augmented Reality nicht mehr getäuscht werden müssen, wie zum Beispiel Balance, Beschleunigung und Propriozeption. Dies sorgt für ein immersiveres Erlebnis und verringert die sogenannte „simulator sickness“ [RB75], da das Gefühlte und das Gesehene übereinstimmen und der eigene Körper gesehen werden kann. Ein weiterer Vorteil ist, dass die Umgebung und damit auch die Mitmenschen gesehen werden können. Somit eignet sich Augmented Reality besonders in Anwendungen, bei denen mit anderen Personen zusammengearbeitet wird oder kommuniziert werden muss.

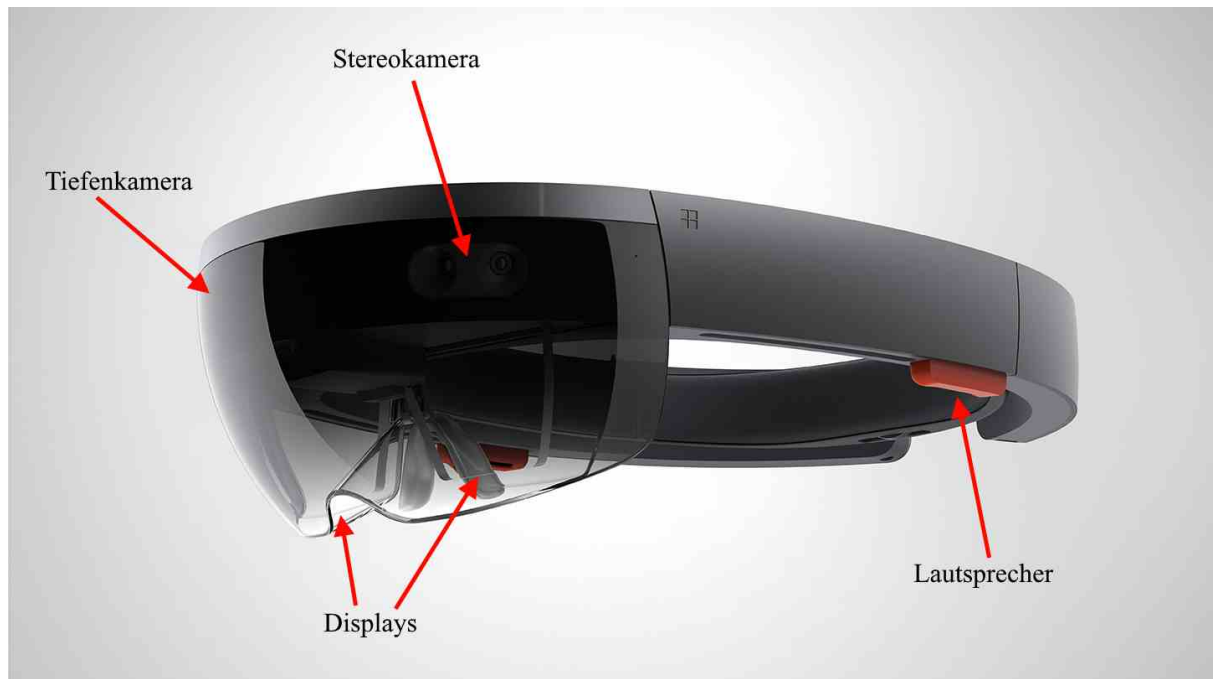
Es existieren dennoch viele Anwendungen, in denen es gewünscht ist, die reale Welt komplett auszublenden. Da bei aktuellen Augmented Reality Brillen aber immer ein Teil der realen Welt gesehen werden kann, ist diese Technologie dafür nicht geeignet. Augmented Reality ist deshalb nicht die Weiterentwicklung von Virtual Reality, sondern eine andere Technologie.

### 2.1.4 Binokulare Abbildungen

Das Ziel der Augmented-Reality-Technologie ist es binokulare Abbildungen, also dreidimensionale virtuelle Objekte, die an einer definierten Position im Raum stehen darzustellen. So können sie von unterschiedlichen Seiten betrachtet werden. Im Gegensatz zu klassischen Displays erzeugen die binokularen Abbildung eine Tiefenwahrnehmung. Eine binokulare Abbildung kann hinter sich liegende Objekte verdecken und von vor ihr liegenden Objekten verdeckt werden. Neben der grafischen Darstellung gehören auch Geräusche und Klänge zu einer solchen Abbildung [Mic16c].

## 2.2 Microsoft HoloLens

Die Microsoft HoloLens (Abbildung 2.2) ist die erste Augmented-Reality-Brille die autark arbeitet [Mic16b]. Mit Hilfe von zwei lichtdurchlässigen Displays auf den Brillengläsern können Objekte durch binokulare Abbildungen so dargestellt werden, als würden sie sich in der realen Welt befinden, siehe Abbildung 2.3. Der Benutzer kann über die Blickrichtung, verschiedene vordefinierte Handgesten und Spracheingabe die HoloLens bedienen. Für eine komfortablere Bedienung kann eine Bluetooth-Tastatur verbunden werden. Um die binokularen Abbildungen an der richtigen Stelle zu platzieren, muss die HoloLens immer wissen, wo sie sich in der Welt befindet. Dazu verwendet die HoloLens eine Vielzahl von Sensoren und Kameras. Der nötige Rechenaufwand, um alle diese Informationen in Echtzeit zu verarbeiten, wird von einer sogenannten Holographic Processing Unit (HPU) bereitgestellt. Es handelt sich dabei um einen speziell von Microsoft angepassten digitalen Signal Prozessor (DSP) mit 24 Kernen und einer Rechengeschwindigkeit von einer Billion Berechnungen pro Sekunde (eine Billion FLOPS). Microsoft gibt an, Berechnungen damit 200-mal schneller ausführen zu können als in reinen Software-Implementierungen [MSP16].



**Abbildung 2.2:** Microsoft HoloLens, eine Augmented Reality Brille [Mic16b]. Zu sehen sind die zwei lichtdurchlässigen Displays, die Stereo Lautsprecher, die Infrarot Tiefenkamera und die Stereo Tiefenkameras.

### 2.2.1 Hardware Spezifikationen

Welche Komponenten in der HoloLens verbaut sind, ist nicht genau bekannt. Die folgende Aufzählung soll einen groben Überblick über frei verfügbare Infos geben [Mic16b]:

- Anzeige
  - Zur Anzeige der binokularen Abbildungen verwendet die HoloLens zwei durchsichtige Linsen mit einer Auflösung von 2,3 Millionen Lichtpunkten.
- Sensoren
  - Neben einer Vielzahl von Kameras (Zwei Paare Stereo Tiefenkameras, eine Infrarot Tiefenkamera und eine 2MP-Foto- und Video-Kamera) sind in der HoloLens vier Mikrophone, eine Inertiale Messeinheit (IMU) und ein Lichtsensor verbaut.
- Rechenleistung
  - Als Hauptprozessor kommt ein Intel Atom x5-Z8100 (1.04 Ghz, 2 Kerne) zum Einsatz. Dieser teilt sich mit der Microsoft Holographic Processing Unit (HPU 1.0) die zwei Gigabyte Arbeitsspeicher



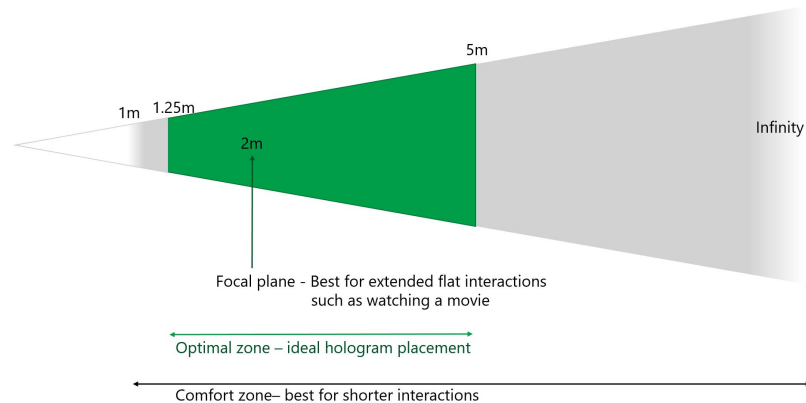
**Abbildung 2.3:** Verschiedene binokulare Abbildungen wie sie ein Benutzer mit Hilfe der HoloLens sehen würde [Mic16c].

- Die HoloLens verfügt außerdem über ein WLAN- und ein Bluetooth-Modul. Zur Soundausgabe sind über den Ohren Lautsprecher angebracht. Der Akku versorgt die HoloLens circa zwei bis drei Stunden mit Energie. Die Brille wiegt 580 Gramm.

### Displaytechnologie

Die beiden durchsichtigen Displays bestehen aus jeweils drei Lagen von Wellenleitern (Waveguides) für die Farben Rot, Grün und Blau. Diese transportieren das Licht von den über den Linsen liegenden Projektoren direkt in die Augen.

Die Displays arbeiten dabei additiv, das heißt es kann kein Licht blockiert werden, sondern nur zusätzliches Licht in die Augen projiziert werden. Schwarze Farbe ist somit nicht wirklich schwarz, sondern transparent und hängt damit vom realen Hintergrund ab. Abhängig von den Farben und dem Hintergrund können kleine schwarze Flecken trotzdem als Schwarz empfunden werden. Große schwarze Flächen sind aber nicht zu realisieren, siehe [Mic16a].



**Abbildung 2.4:** Der empfohlene Abstand zwischen Augen und 3D-Objekten beträgt zwischen 1,25 Metern und fünf Metern, akzeptabel ist ein Abstand von größer als einem Meter, Quelle: [Mic17a]

### 2.2.2 Designrichtlinien für 3D-Objekte in HoloLens-Anwendungen

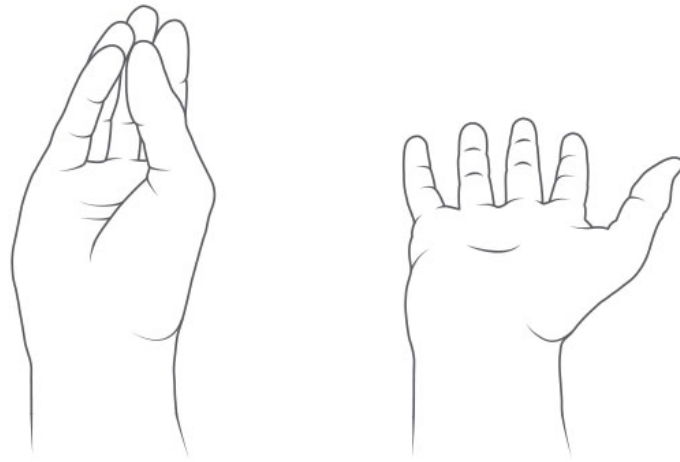
Das Design von 3D-Objekten für die HoloLens unterscheidet sich in einigen Punkten von klassischen 3D-Anwendungen. Aufgrund des additiven Lichts des HoloLens-Displays sind bei der Farbwahl Regeln zu beachten. Da die Displays nur Licht zur Umgebung hinzufügen, werden dunkle Farben transparent dargestellt. Die Farbe Schwarz ist komplett transparent und damit nicht sichtbar. Aus diesem Grund sollten helle Farben verwendet werden. Grautöne und dunkle Farben wirken wie ein Transparenz-Effekt. Hologramme können die Umgebung nicht verschatten.

Neben der Farbe ist auch die Positionierung und Größe der 3D-Objekte wichtig. Da das Display der HoloLens nur einen Sichtbereich von circa 40 Grad und nicht das komplette Sichtfeld eines Menschen abdeckt, sollten die 3D-Objekte so platziert werden, dass sich das komplette 3D-Objekt im Sichtfeld befindet. Ist das 3D-Objekt zu groß, werden seitlich Teile davon abgeschnitten. Zusätzlich sollte die Entfernung zwischen dem Auge und den 3D-Objekten zwischen 1,25 Metern und fünf Metern betragen, siehe Abbildung 2.4. Objekte, die näher als einen Meter sind können nichtmehr richtig fokussiert werden und der Benutzer muss schießen, was unangenehm für die Augen ist [Mic17a].

### 2.2.3 Interaktion mit der HoloLens

Zum Interagieren mit der HoloLens stehen die Blickrichtung, verschiedene Handgesten und Spracheingabe zur Verfügung. Diese unterscheiden sich damit grundsätzlich von Desktop-Anwendungen, die mit einer Maus bedient werden und Anwendungen für mobile Geräte, die über einen Touchscreen bedient werden. In den folgenden Abschnitten wird auf die einzelnen Interaktionsmöglichkeiten genauer eingegangen.





**Abbildung 2.5:** Skizze zur Durchführung der *Bloom* Geste der HoloLens [Mic17d]. Erster Schritt: Alle Finger werden zusammengeführt. Zweiter Schritt: Die Hand wird geöffnet. Während der Geste muss die Handfläche nach oben zeigen.

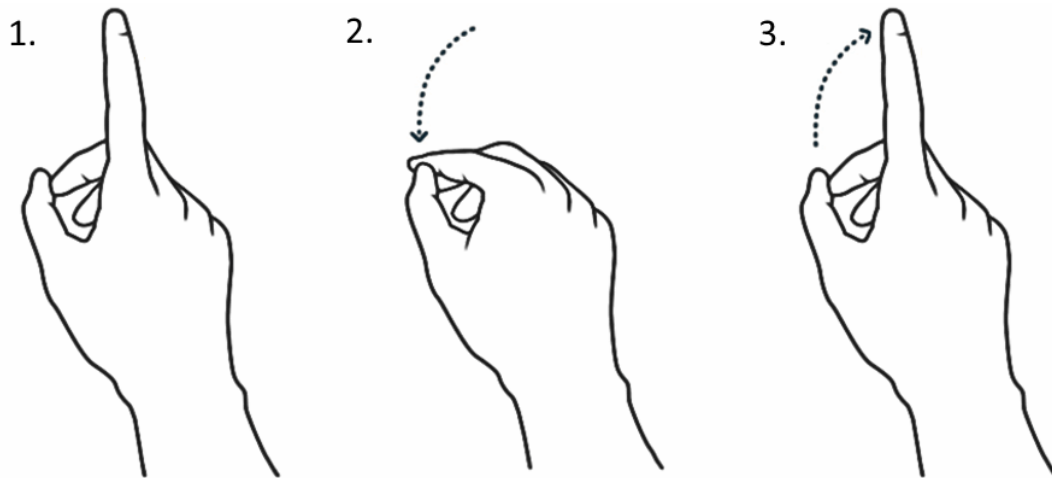
### Blickrichtung

Die Interaktion mit Hilfe der Blickrichtung dient zum Markieren von Elementen. Die Blickrichtung ist vergleichbar mit dem Bewegen einer Maus bei Desktop-PCs. Der aktuelle Fokus der Blickrichtung wird in vielen Anwendungen durch einen Cursor visualisiert. Die Blickrichtung der HoloLens hängt nur von der Kopfrichtung ab. Die Bewegung der Augen wird nicht gemessen und spielt deswegen auch bei der Blickrichtung keine Rolle.

### Handgesten

Die HoloLens erkennt mit Hilfe der Infrarot-Tiefenkamera Handgesten. Aktuell kann die HoloLens zwei verschiedene Arten von Gesten erfassen. Die erste Geste ist die *Bloom* Geste, siehe Abbildung 2.5. Diese dient zum Zurückkehren zum Hauptmenü und öffnen und schließen des Hauptmenüs. Sie ist vergleichbar mit dem Drücken der Windowstaste bei einem Windows-Desktop-PC oder der *Home*-Taste bei Smartphones. Zur Interaktion mit einer Anwendung kann diese Geste nicht verwendet werden.

Die zweite Geste ist die *Air-Tap* Geste, siehe Abbildung 2.6. Diese Geste kann auf verschiedene Arten verwendet werden. Zum Auswählen von Objekten werden die drei Schritte in Abbildung 2.6 direkt hintereinander ausgeführt. Diese Aktion ist vergleichbar mit einem Mausklick. Bei dieser *Air-Tap*-Geste ist die Position der Hand bedeutungslos. Auf welchem Objekt, diese Geste ausgeführt wird, entscheidet die Blickrichtung.



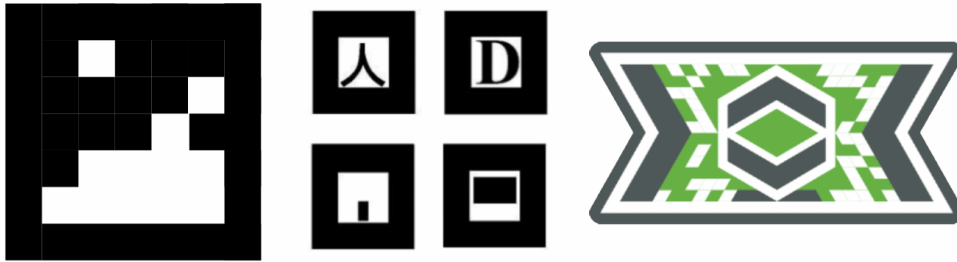
**Abbildung 2.6:** Skizze zur Durchführung der *Air-Tap* Geste der HoloLens [Mic17d]. Dabei wird der Zeigefinger zuerst nach oben gestreckt, dann auf die anderen Finger abgelegt und anschließend wieder gestreckt.

Eine zweite Möglichkeit ist es, die Geste nur halb auszuführen, also nur Schritt Eins und Zwei und anschließend die Hand in dieser Fingerstellung zu bewegen. Diese relative Bewegung der Hand wird von der HoloLens erkannt und kann zum Beispiel zum Verschieben oder Rotieren von Objekten oder zum Scrollen durch Listen verwendet werden. Diese Geste wird Manipulationsgeste genannt.

### Sprachbefehle

Sprachbefehle sind eine weitere Form der Interaktion mit der HoloLens. Durch vorkonfigurierte Sprachbefehle sind Kommandos wie zum Beispiel „Erstelle neues Objekt“ möglich. Die Sprachbefehle sollten dabei mehrsilbig sein, da einsilbige Befehle nicht zuverlässig erkannt werden. Bei mehreren Sprachbefehlen sollte darauf geachtet werden, dass diese sich phonetisch nicht ähneln. Im Fabriklayout-Simulator für die HoloLens ersetzen die Sprachbefehle große Teile der Oberflächen und Menüs. Das dauerhafte Einblenden von Schaltflächen und Menüs in HoloLens Anwendungen stört die Betrachtung der 3D-Objekte.

Neben einfachen Sprachkommandos sind auch komplexere Befehle basierend auf der „Speech Recognition Grammar Specification“ (SRGS) [Wor17] möglich. Diese Grammatik kann mit Hilfe von SRGS Dateien konfiguriert werden und erlaubt Sprachbefehle wie zum Beispiel: „Erstelle einen kleinen/mittleren/großen Würfel/Zylinder/Quader“. So müssen nicht alle Permutationen als einzelne Sprachbefehle konfiguriert werden.



**Abbildung 2.7:** Verschiedene Marker zur Objekterkennung von links nach rechts: Matrix Marker[Rek98], Vier ARToolkit Marker[Fia05], Vuforia VuMark Marker[PTC17b]

## 2.3 Markerbasierte Objekterkennung

Um die vollen Möglichkeiten der Augmented-Reality-Technologie ausschöpfen zu können, müssen Objekte in der realen Welt erkannt und lokalisiert werden, sodass kontextbezogenen Informationen an der korrekten Stelle angezeigt werden können. Im Folgenden Kapitel werden Verfahren der Objekterkennung vorgestellt und erklärt.

Ziel der markerbasierten Objekterkennung ist es, mithilfe einer Kamera ein bekanntes Muster in der Welt zu finden und anschließend aus der Kameraposition und der Position des Musters im Kamerabild die reale Position und Orientierung des Musters zu bestimmen. Solche Muster werden in der Objekterkennung Marker genannt. In Abbildung 2.7 sind verschiedene Marker aus verschiedenen Bibliotheken dargestellt.

### 2.3.1 Erkennung von Markern

Um die Marker zuverlässig zu erkennen, sollten sie bestimmte Eigenschaften aufweisen. Dazu gehört ein hoher Kontrast (schwarz/weiß) im Bild. Die Marker müssen unabhängig von Rotationen und Projektion erkennbar sein. Ein klassisches Beispiel dafür sind die Matrix-Marker von Rekimoto [Rek98], siehe Abbildung 2.7 ganz links. Durch den hohen Kontrast und die langen Kanten können Schwellwertverfahren oder Kantenerkennung dieses Muster zuverlässig erkennen. In einigen Markern werden Matrix Codes verwendet, um die einzelnen Marker unterscheiden zu können.

Die Objekterkennung und Lokalisierung läuft dabei in drei Schritten ab:

1. Erkennung des Markers im aufgenommenen Bild. Dazu werden Verfahren der Bildverarbeitung verwendet wie Schwellwertverfahren oder Kantenerkennung.
2. Falls der Marker eine ID besitzt, muss diese dekodiert werden. Um fehlerhafte Erkennungen zu vermindern, werden Fehlererkennungscode oder Fehlerkorrekturcodes eingesetzt. Zum Beispiel durch Einsatz von Paritätsbits oder einer zyklischen Redundanzprüfung (CRC).

3. Berechnen der Markerposition und Orientierung. Durch die Kenntnis der Größe und Form der Marker kann aus dem Kamerabild die Position und Orientierung im Raum bestimmt werden. Zusätzlich wird dazu die Position und Orientierung der Kamera zum Aufnahmezeitpunkt und Kameraparameter wie Blickwinkel und Verzerrung benötigt. Zur Weiterverarbeitung können Filter zum Einsatz kommen, die einzelne fehlerhafte Erkennungen filtern und die Stabilität der Positionen verbessern.

Vorteile der markerbasierten Objekterkennung sind die Geschwindigkeit und Robustheit der Erkennung. Durch die vordefinierten Strukturen der Marker kommt die Objekterkennung mit wenigen Operationen aus und ist schneller als wenn beliebige Eigenschaften eines Bildes erkannt werden müssen. Die Marker sind günstig herzustellen und einfach zu verwenden. Ein Nachteil ist, dass die Umgebung vorbereitet werden muss, also die Marker an den richtigen Stellen angebracht werden müssen.

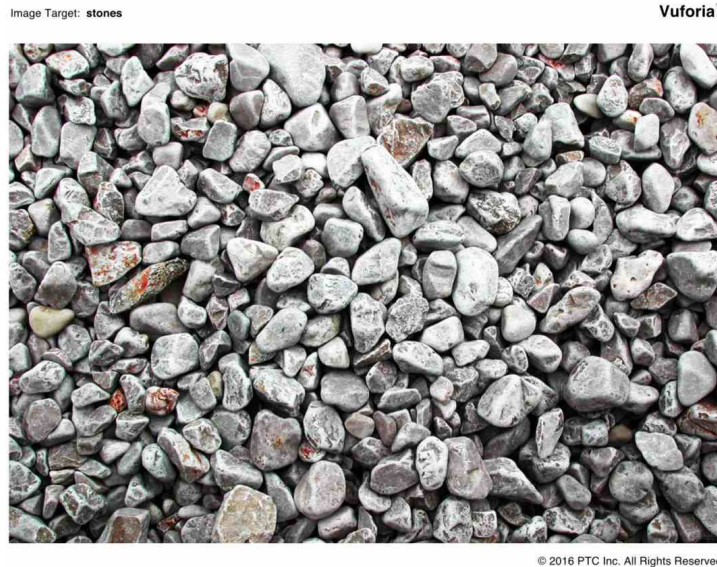
### 2.3.2 Natural Feature Tracking (NFT)

Natural Feature Tracking (deutsch: Tracking von natürlichen Eigenschaften) ist eine besondere Form der Markererkennung. Anstatt eines speziell entwickelten Markers kann jedes beliebige Bild verwendet werden. Dieses Bild wird im Voraus der Objekterkennung bekannt gemacht und kann dann später von dieser wieder im Bild detektiert werden. Nicht jedes Bild eignet sich für eine zuverlässige Erkennung. Es sollte auf viele harte Kanten und einen hohen Kontrast geachtet werden. In Abbildung 2.8 ist ein natürlicher Marker dargestellt, der sich gut zur Erkennung eignet.

Vorteil dieser Art von Erkennung ist es, dass existierende Elemente wie Etiketten oder Bilder in Bedienungsanleitungen genutzt werden können, um Objekterkennung durchzuführen. Im Vergleich zu einfachen quadratischen Markern ist die Erkennung zuverlässiger, da eine größere Anzahl an Features im Bild verwendet werden. Nachteilig ist der höhere Rechenaufwand, der nötig ist, um die Marker zu erkennen.

## 2.4 Visualisierung

Unter Visualisierung versteht sich die Methode, Daten visuell darzustellen. Die Ergebnisse helfen Simulation und Berechnungen zu analysieren [DB91]. Visualisierungen helfen in Gebieten, in denen große Datenmengen vorkommen und das klassische Betrachten von Zahlen oder Tabellen nicht ausreicht. Dazu werden grafische Repräsentationen der Daten definiert. Durch die grafische Repräsentation wird versucht, die kognitiven Fähigkeiten des Menschen auszunutzen, um Sachverhalte besser zu vermitteln. Die Ziele der Visualisierung können unterschiedlich sein. Beispiele für solche Ziele sind zum Beispiel Erkenntnisgewinn, Kommunikation oder die Steuerung einer Anwendung. Je nach Einsatzgebiet muss auch die Visualisierung angepasst werden.



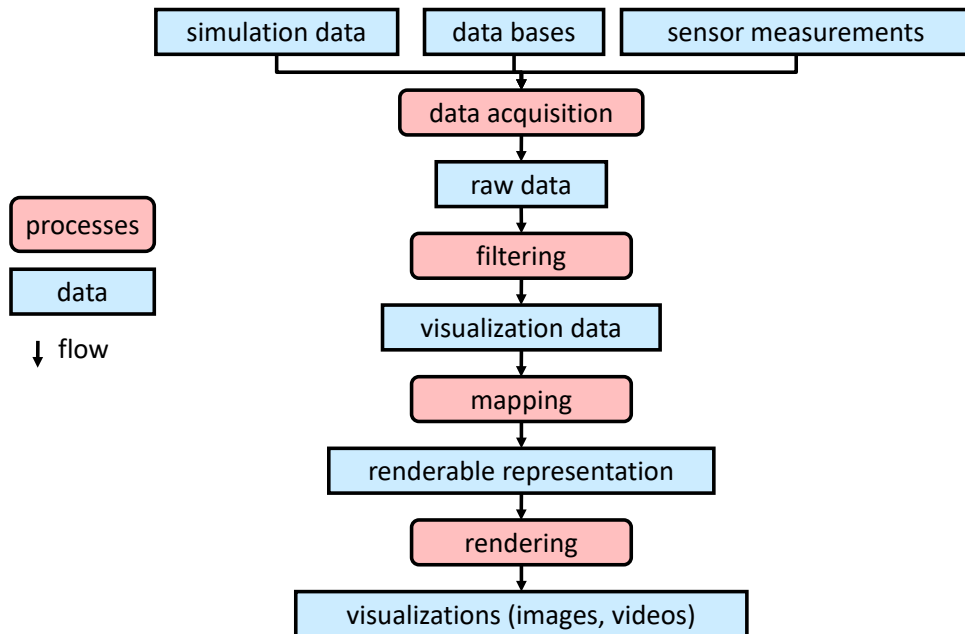
**Abbildung 2.8:** Ein Beispiel für einen natürlichen Marker [PTC17a]. Durch die vielen Kontraste und ungleichmäßigen Strukturen eignet sich dieser Marker gut zur Erkennung.

In der Visualisierung werden zwei Arten unterschieden: Die Informationsvisualisierung und die wissenschaftliche Visualisierung. In der wissenschaftlichen Visualisierung werden Daten dargestellt, die einen örtlichen Bezug haben. Diese Daten sind kontinuierlich, werden aber periodisch abgetastet. Beispiele dafür sind die Druckverteilung in einem Körper. In der Informationsvisualisierung sind die Daten abstrakt und besitzen keinen örtlichen Bezug. Die Daten sind diskret. Beispiele dafür sind Graphen oder Datenstrukturen.

### 2.4.1 Visualisierungs-Pipeline

Die Visualisierungs-Pipeline ist eine Standard-Vorgehensweise zur Erstellung von Visualisierungen von Daten, siehe Abbildung 2.9. Im Folgenden soll der Ablauf dieser Pipeline erklärt werden. Als Beispiel dient dabei die Auslastung einer Produktionsstraße.

Im ersten Schritt der Pipeline steht die Beschaffung der zu visualisierenden Daten. Dazu wird während der Simulation die Zeit, in der einzelne Module ausgelastet sind, gemessen. Der zweite Schritt der Pipeline filtert die aufgenommenen Daten. Im Beispiel wird aus den Daten die durchschnittliche Auslastung oder die Auslastung während der letzten Minute berechnet. Im dritten Schritt werden die zu visualisierenden Daten darstellbaren Repräsentationen zugeordnet. Dazu wird um die einzelnen Elemente der Produktionsstraße ein halbtransparenter Quader gelegt und anhand der Auslastung die Farbe des Quaders festgelegt. Dabei wird ein Farbverlauf von Grün nach Rot verwendet. Grün bedeutet voll ausgelastet, rot bedeutet unausgelastet. Im letzten Schritt werden die darstellbaren Repräsentationen gerendert. Im Beispiel wird die



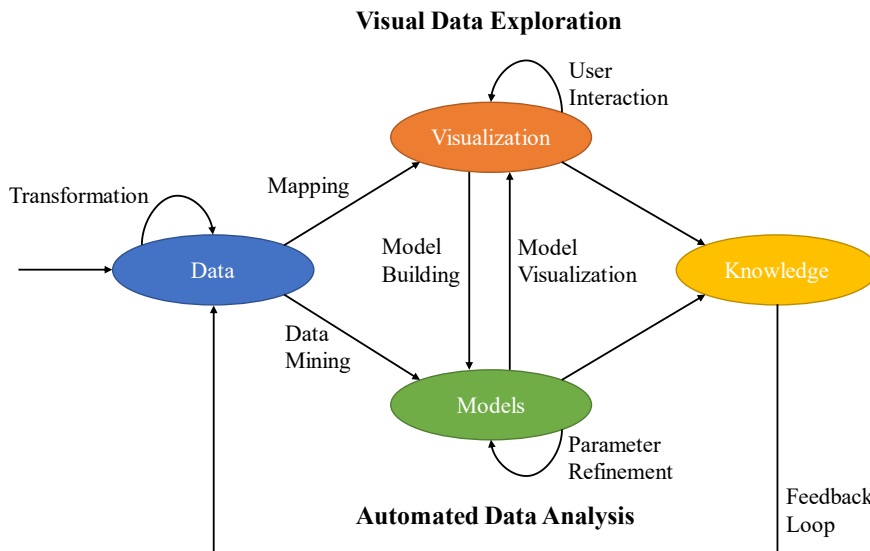
**Abbildung 2.9:** Aufbau der Visualisierungs-Pipeline nach Haber et al. [HM90]. Jeder Datensatz durchläuft die einzelnen Visualisierungsschritte bis dieser im letzten Schritt angezeigt wird.

Produktionsstraße und die zugehörigen Quader dargestellt. So kann auf einen Blick erfasst werden, welche Module der Produktionsstraße wie ausgelastet sind.

### 2.4.2 Visuelle Analytik

Die Visuelle Analytik besteht aus zwei Hauptfeldern, der visuellen Datenexploration und der automatischen Datenanalyse, siehe Abbildung 2.10. Zur automatischen Datenanalyse kommen Methoden aus den Bereichen Maschinelles Lernen und *Data Mining* zum Einsatz. Bei der Exploration der Daten spielt die Interaktion eine wichtige Rolle. In der Visuellen Analyse werden grafische Repräsentationen der Daten verwendet, um Entscheidungen zu treffen. In Abbildung 2.10 sind die wichtigsten Abläufe der Visuellen Analytik skizziert. Im ersten Schritt werden die Daten so transformiert, dass diese sich zur weiterverarbeitung eignen. Diese Daten werden anschließend auf grafische Repräsentationen gemappt. Gleichzeitig wird aus den Daten mit Hilfe von *Data Mining* ein Modell erstellt. Dieses Modell wird unterstützend bei der Visualisierung verwendet. Durch Interaktion kann die Visualisierung angepasst werden. Die Interaktion mit der Visualisierung wird außerdem zum Verfeinern des Modells der Daten eingesetzt. Durch Analyse des Modells und der Visualisierungen wird Wissen generiert was im nächsten Schritt bei der Erfassung der Daten wieder eingebracht werden kann.

Einige Kriterien für den Einsatz der Visuellen Analytik sind große und dynamische Datensätze die unterschiedliche Informationen beinhalten und eine hohe Anzahl an Dimensionen



**Abbildung 2.10:** Das Modell der visuellen Analytik nach Keim et al. [KKEM10]. Die Daten werden transformiert und zur Visualisierung und zum Erstellen eines Modells verwendet. Das Modell unterstützt die Visualisierung. Die Interaktion mit der Visualisierung verfeinert das Modell. Aus Modell und Visualisierung wird Wissen generiert welches bei der Datenerfassung wieder eingesetzt werden kann.

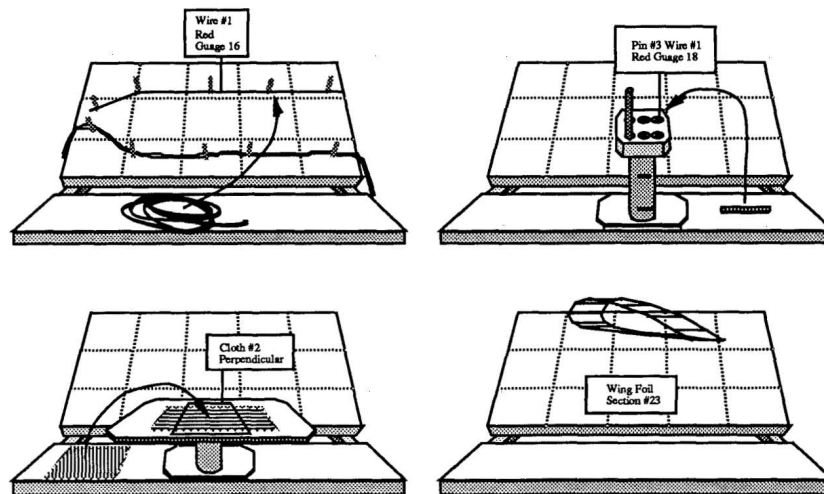
aufweisen. In Problemen die durch die Visuelle Analytik gelöst werden, wird meist explorativ nach Lösungen gesucht. Das heißt, das Ziel ist nur abstrakt formuliert. Es werden ständig neue Hypothesen aufgestellt und geprüft. Einsatzfelder sind unter anderem die Sicherheitsbranche, das Gesundheitswesen, die Stromversorgung, der Finanzsektor, der Journalismus und die Industrie.

## 2.5 Augmented Reality in Produktion und Fertigung

In diesem Abschnitt werden Arbeiten vorgestellt, die eine ähnliche Zielsetzung wie diese Masterarbeit haben. Die Arbeiten sind chronologisch sortiert, um die Entwicklung der Technologien aufzuzeigen.

### 2.5.1 Montage und Wartung

Ein Einsatzgebiet von Augmented Reality in Produktion und Fertigung ist zurzeit die Montage und Wartung. Die Technik ermöglicht es Montageinformationen im Sichtfeld des Arbeiters einzublenden. Ein Beispiel dafür ist die Arbeit von Caudell und Mizell [CM92], die schon 1992 ein Augmented-Reality-Display benutzten. Das virtuelle Bild war dabei nur mit einem



**Abbildung 2.11:** Montageaufgaben mit Hilfe von Augmented Reality. Links oben: Verkabelungs-Aufgabe, rechts oben: Verkabelung der Anschlussbuchse, links unten: Platzieren der Verbundstruktur-Materialien, rechts unten: 3D-Drahtmodell eines Montageschritts [CM92].

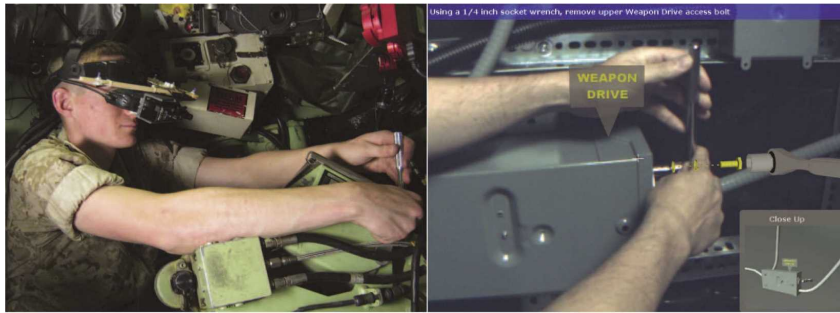
Auge zu sehen, hatte eine große Verzögerung und ein kleines Sichtfeld. Trotzdem wurden vier Anwendungen entwickelt, siehe Abbildung 2.11.

Eine aktuellere Arbeit von Henderson et al. [HF11] aus dem Jahr 2011 beschäftigt sich mit dem Einsatz von Augmented Reality in Wartung und Reparatur. Eingesetzt wurde der Prototyp bei Wartungsarbeiten in einem Geschützturm eines Armeefahrzeugs, siehe Abbildung 2.12. Die Mechaniker werden durch verschiedene Informationen bei Ihrer Arbeit unterstützt:

- Verschiedene Pfeile, um die Aufmerksamkeit auf eine bestimmte Stelle zu lenken
- Anweisungen und Warnungen in Textform
- Eine Zoomfunktion, die das bearbeitete Teil größer darstellt
- 3D-Modelle von Werkzeugen und Bauteilen an ihrer gewünschten Position

In einer Benutzerstudie wurde das AR-System mit Informationen auf einem LC-Display verglichen. Bei intuitiver Bedienung und allgemeiner Zufriedenheit wurde das AR-System besser eingestuft als das Display. Jedoch fanden die Mechaniker den Bedienkomfort beim Display besser und würden es einem AR-System bevorzugen. Probleme waren vor allem die behinderte Sicht und das schwere AR-System auf dem Kopf.





**Abbildung 2.12:** Augmented Reality bei der Montage eines Geschützturms. Links: Ein Mechaniker mit dem verwendeten auf dem Kopf getragenen Display, rechts: Der Blick durch das Display [HF11].

### 2.5.2 Planung und Design

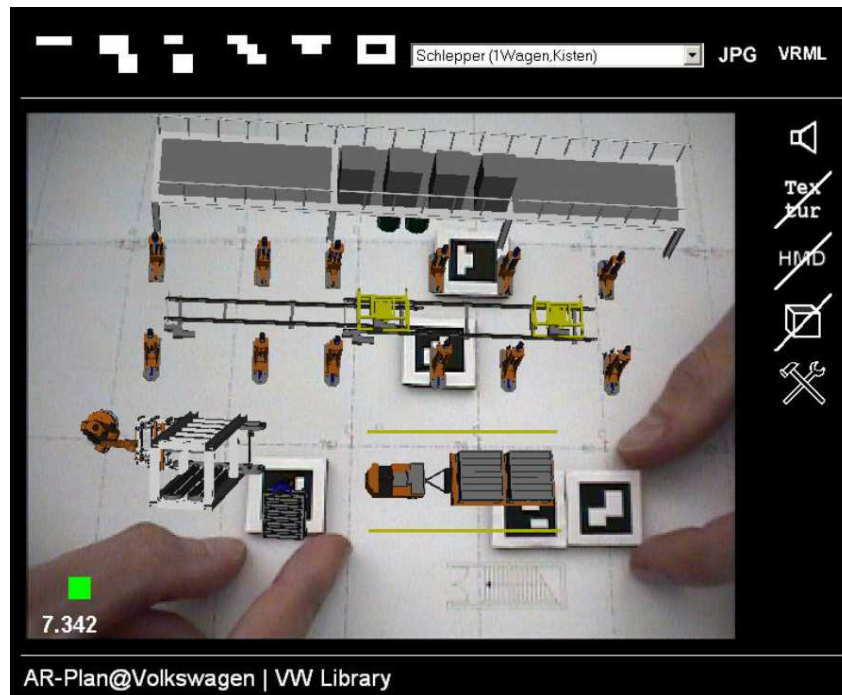
Ein weiteres Einsatzgebiet ist die Planung von Fabrikanlagen. So können beispielsweise neue Maschinen virtuell in eine Halle gestellt werden und eventuelle Probleme früher erkannt werden.

In ihrem Artikel „Designing an immersive virtual reality interface for layout planning“ aus dem Jahr 2000 evaluieren Korves et al. ein neues Werkzeug zur Planung von Fabriklayouts [KL00]. Ziel war es, herauszufinden, ob sich mit Hilfe von Virtual-Reality-Technologie im Vergleich zu normalen Desktop-PCs Designfehler an einer Produktionsstraße leichter finden lassen. Die Ergebnisse zeigen, dass mit Hilfe des verwendeten Head Mounted Displays (HMD) deutlich bessere Ergebnisse erzielt werden. Während Benutzer am Bildschirm nur im Durchschnitt 26% der Fehler gefunden haben, beträgt dieser Wert beim HMD 79%.

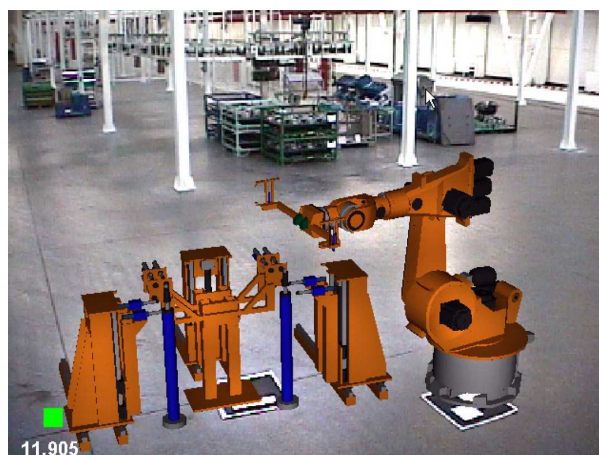
Der Artikel „Augmented Reality for manufacturing planning“ aus dem Jahr 2003 von Doil et al. [DSAP03] stellt zuerst Anforderungen an eine Augmented-Reality-Anwendung für Produktionsplanung auf. Im Anschluss wird das Programm AR-Plan vorgestellt, welches diese Anforderungen implementiert. Es ist als Client-Server Anwendung realisiert, um die Recheneinheit des mobilen Anzeigegegeräts zu entlasten. Ein Teil der Anwendung ist eine interaktive Fabriklayoutplanung, bei der die virtuellen Anlagenteile auf real existierende Marker projiziert werden. Durch verschieben der Marker kann das Layout interaktiv angepasst werden (siehe Abbildung. 2.13).

Eine andere Anwendung in AR-Plan erlaubt es, virtuelle Objekte in der späteren realen Umgebung anzuzeigen, siehe Abbildung 2.14. Durch ein Webinterface werden dynamisch verschiedene Module nachgeladen und angezeigt. Ein virtuelles Messwerkzeug erlaubt es, Abstände zu messen. Zusätzlich können virtuelle Arbeiter eingefügt werden um so Arbeitsplätze auf Ergonomie zu überprüfen.

In ihrem Artikel aus dem Jahr 2016 designen Yew et al. [YON16] eine neue Architektur, bestehend aus verschiedenen intelligenten Maschinen und Objekten, die über ein Netzwerk



**Abbildung 2.13:** Interaktive Augmented Reality Fabriklayoutplanung in AR-Plan [DSAP03]. Die Marker werden mit Hilfe einer Kamera erkannt. Anschließend werden an den erkannten Positionen Fabrikelemente dargestellt.



**Abbildung 2.14:** Integrieren einer virtuellen Maschine in eine Fabrikhalle [DSAP03]. Durch die Visualisierung am tatsächlichen Einsatzort können Platzverhältnisse besser eingeschätzt werden.



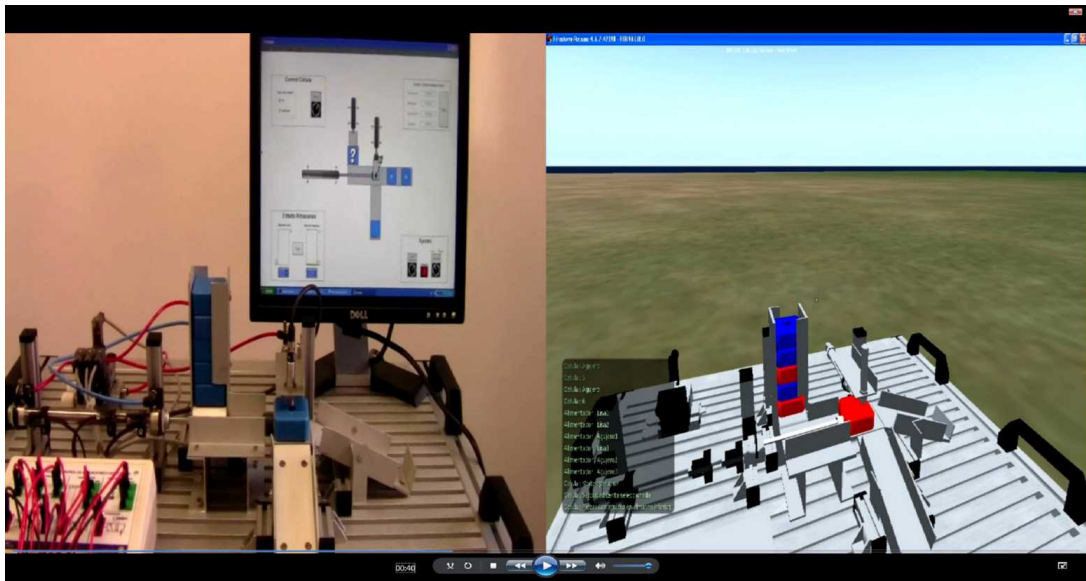
**Abbildung 2.15:** Einblendung von Informationen an einer Werkzeugmaschine. Links: Angezeigte Daten während des Normalbetriebs, rechts: Angezeigte Daten während der Wartung [YON16].

verbunden sind. Dies ermöglicht es, Live-Daten aus der Maschine auszulesen die dann über ein HMD angezeigt werden können. Für verschiedene Objekte sind dabei verschiedene Modi möglich. So werden während dem Betrieb andere Informationen angezeigt als während der Wartung, siehe Abbildung 2.15. Neben intelligenten Maschinen existieren auch intelligente CAD Modelle. Sie können mit Hilfe eines Tools aus dem CAD Programm exportiert werden und anschließend im AR-System zum Betrachten und Interagieren bereitgestellt werden.

Calvo et al. [CLZG16] verfolgen mit Ihrer Arbeit das Ziel, ein Softwaresystem für Produktion und Virtual Reality zu entwickeln, welches nur auf offenen Software Lösungen aufbaut. Ihre Ziele sind:

- Die Simulation von Prozessen für Lehre, Marketing oder Prototypendesign.
- Die Fernbedienung von realen Prozessen, besonders für gefährliche Umgebungen.

Die Anwendung besteht aus einer MVC-Architektur. Das Model ist ein OpenSimulator-Server. Auf diesen Server können sich die Views verbinden. Die Controller sind entweder die Maschinensimulationen oder die realen industriellen Controller in den Anlagen. Getestet wurde dieser Ansatz anhand einer Sortiermaschine, siehe Abbildung 2.16.



**Abbildung 2.16:** Links: Eine Sortiermaschine, rechts: Das virtuelle Abbild der Sortiermaschine in OpenSimulator [CLZG16].

# 3 Technische Möglichkeiten der Microsoft HoloLens

In diesem Kapitel werden die technischen Möglichkeiten der HoloLens überprüft. Dabei sollen Stärken und Schwächen der Brille identifiziert werden. Neben der Evaluierung der Grafikleistung werden auch die Technologien *Holographic Remoting* und *Holographic Sharing* analysiert. Außerdem wird auf markerbasierte Objekterkennung und das dreidimensionale Umgebungsmodell eingegangen. Dieses Wissen über die Brille soll im nächsten Kapitel helfen, eine HoloLens Anwendung zu konzipieren.

## 3.1 Evaluierung der Grafikleistung

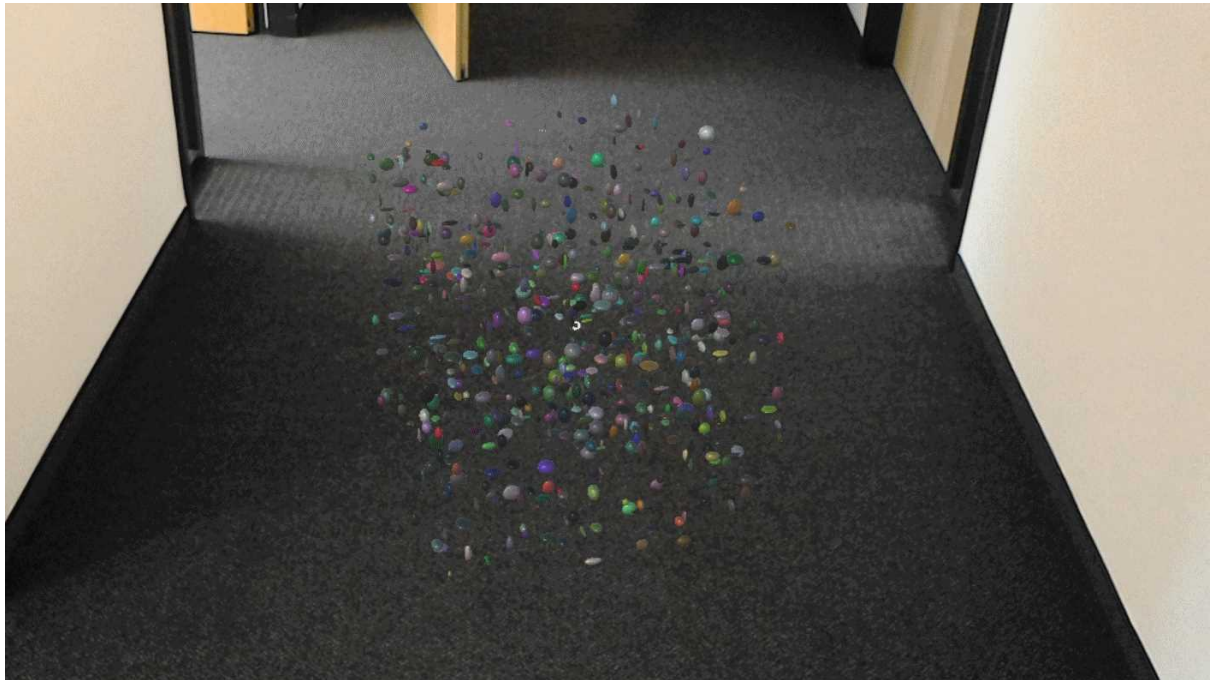
Im Gegensatz zu vielen Desktopanwendungen ist die Bildwiederholungsrate bei Virtual und Augmented Reality Anwendungen kritisch. Je geringer die Bildwiederholungsrate ist, desto stärker weicht das virtuelle Bild von der Realität ab. Dieser Effekt führt zur sogenannten „Simulator Sickness“ [RB75]. Um virtuelle Anwendungen über längere Zeitspannen anstatt nur für kurze Tests zu verwenden, ist eine konstant hohe Framerate deshalb von hoher Wichtigkeit. In diesem Abschnitt wird die Grafikleistung der HoloLens evaluiert. Es wird überprüft wie sich die Bildwiederholungsrate bei steigender grafischer Komplexität verhält [Rei17].

### 3.1.1 Testablauf

Zur Evaluation der Grafikleistung der HoloLens werden verschiedene Einsatzszenarien mit jeweils unterschiedlicher Komplexität getestet. Ziel ist es, herauszufinden, ab welcher grafischen Komplexität die Bildwiederholungsrate der HoloLens unter die Grenze von 60 Hertz fällt.

In HoloLens-Anwendungen treten immer wieder ähnliche Einsatzszenarien auf. Um ein realistischeres Ergebnis zu erhalten, werden verschiedene dieser Einsatzszenarien getestet. Die getesteten Einsatzszenarien bauen aufeinander auf:

- **Reine Visualisierung:** Dieses Szenario deckt Anwendungen ab, welche keine Interaktion zulassen und nur über die Fähigkeit zum Betrachten eines virtuellen Objekts verfügen. Zum Vergleichen ist dieses Szenario am besten geeignet, da es unabhängig von äußeren Einflüssen ist.

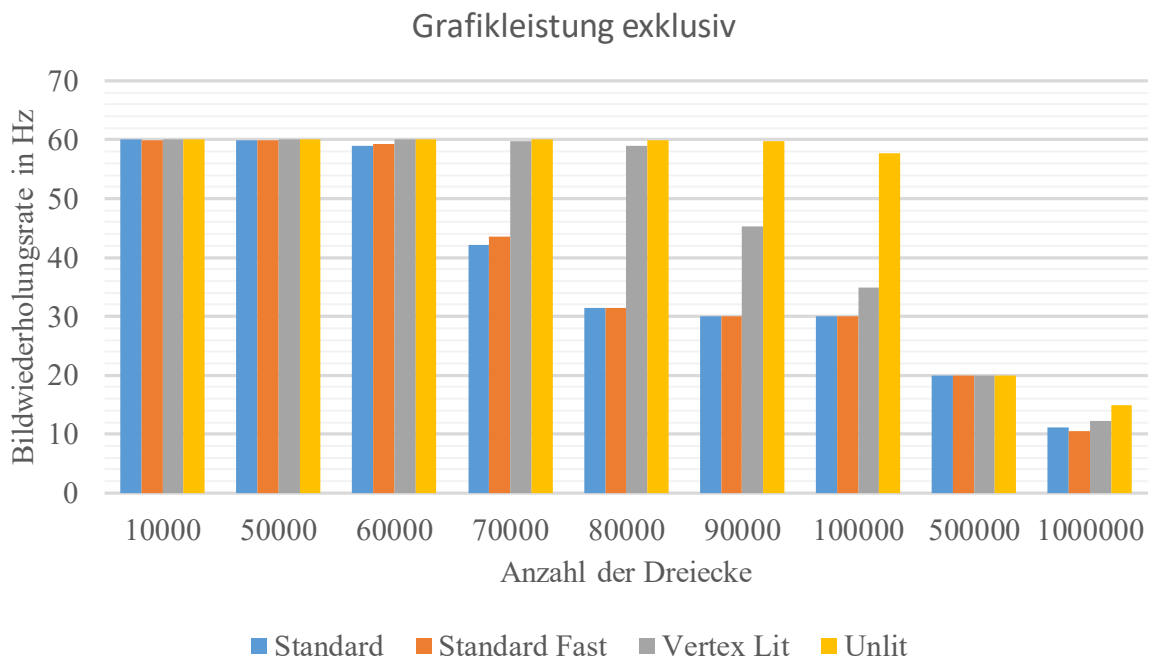


**Abbildung 3.1:** Screenshot während der Durchführung eines Grafiktests. Die Kugeln sind über das komplette Sichtfeld der HoloLens verteilt.

- **Klassisches (AR/VR-) Einsatzszenario:** Dieses Szenario ist die Grundlage für einen Großteil der HoloLens Anwendungen. Neben dem Betrachten von 3D-Objekten sind Interaktionen über Handgesten und Spracherkennung möglich. Zusätzlich wird ein interaktiver Zeiger angezeigt.
- **Verdeckung durch reale Objekte:** Dieses Szenario verfügt über alle Eigenschaften des vorherigen Szenarios, es werden aber zusätzlich die von der HoloLens generierten Informationen der Umgebung eingelesen und zur Verdeckung von eigenen Objekten verwendet. Es wird so ermöglicht, dass virtuelle Objekte, die sich hinter einem realen Objekt befinden, ausgeblendet werden.
- **Anzeigen des Dreiecksnetzes:** Ähnlich wie das vorherige Szenario. Hier wird das Dreiecksnetz jedoch nicht zur Verdeckung verwendet, sondern als Drahtgitter visualisiert. So werden Interaktion mit der Umgebung erleichtert.

Jedes dieser Szenarien wird mit verschiedenen Dreiecksanzahlen und Shadern getestet. Neben dem Unity Standard Shader werden auch drei Shader aus dem HoloToolkit [Mic17b] getestet. Der Standard Fast Shader ist eine für die HoloLens optimierte Variante des Unity Standard Shaders. Der Vertex Lit Shader zeichnet sich durch einen schnellen Pixel Shader aus. Der Unlit Shader nutzt keine Beleuchtung.

Die Anwendung zum Evaluieren der Grafikleistung besteht aus einer sich bewegendem Punktlichtquelle und texturierten Kugeln, siehe Abbildung 3.1. Die Kugeln werden zufällig im



**Abbildung 3.2:** Ergebnisse des Benchmark-Tests für das Szenario *Reine Visualisierung* bei verschiedenen Dreiecksanzahlen und Shadern

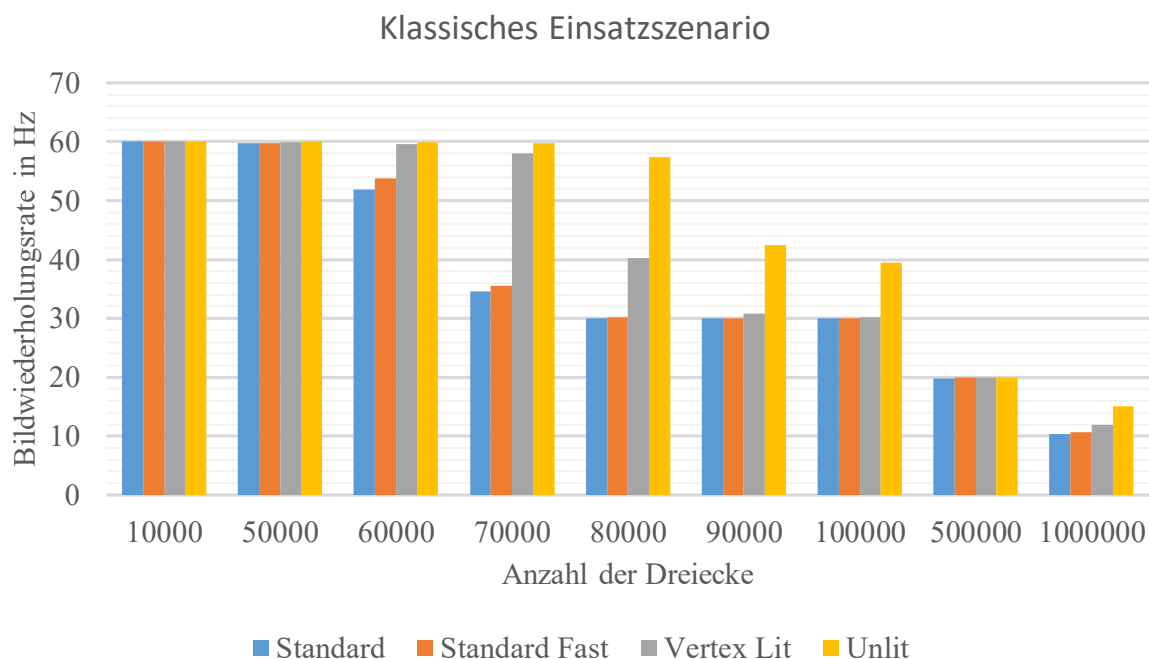
Sichtfeld der HoloLens verteilt, sodass immer alle Kugeln zu sehen sind. Zu Beginn eines Tests werden die für die Dreiecksanzahl benötigten Kugeln initialisiert. Nach einer 60 Sekunden dauernden Testphase wird die durchschnittliche Bildwiederholungsrate ausgegeben.

### 3.1.2 Ergebnisse der Messungen

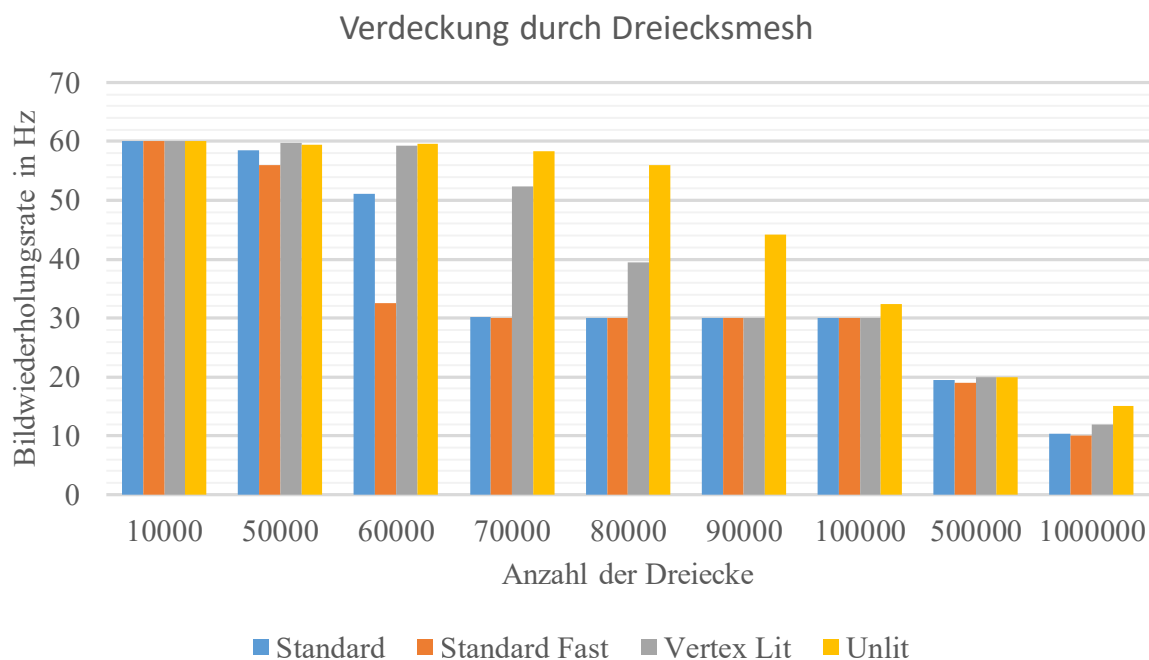
Die Ergebnisse der Messungen sind in den Diagrammen in Abbildung 3.2, Abbildung 3.3, Abbildung 3.4 und Abbildung 3.5 dargestellt.

Um die Ergebnisse auswerten zu können, muss zuerst ein Blick auf die Funktionsweise des Displays gelegt werden. Das Display der HoloLens arbeitet mit 60 Hertz. Da die Anwendung auf die vertikale Synchronisation des Displays wartet kann die Frequenz von 60 Hertz nicht überschritten werden. Außerdem ist zu beobachten, dass die Frequenz sich in dem meisten Fällen auf 60, 30 oder 20 Hertz einpendelt. Der Grund dafür liegt darin, dass ein Berechnungszyklus immer ähnlich lange dauert und deshalb entweder jeden, jeden zweiten oder jeder dritten Zyklus des Displays der Bildspeicher aktualisiert wird. Zwischen diesen Werten existieren instabile Zwischenwerte. Bei einer Bildwiederholungsrate von beispielsweise 45 Hertz benötigt der Render-Zyklus in 50% der Fälle kurzer als einen Display-Zyklus und in 50% der Fälle länger. Die Gefahr besteht darin, dass eine geringe Erhöhung der Komplexität dazu führt, dass der

### 3 Technische Möglichkeiten der Microsoft HoloLens

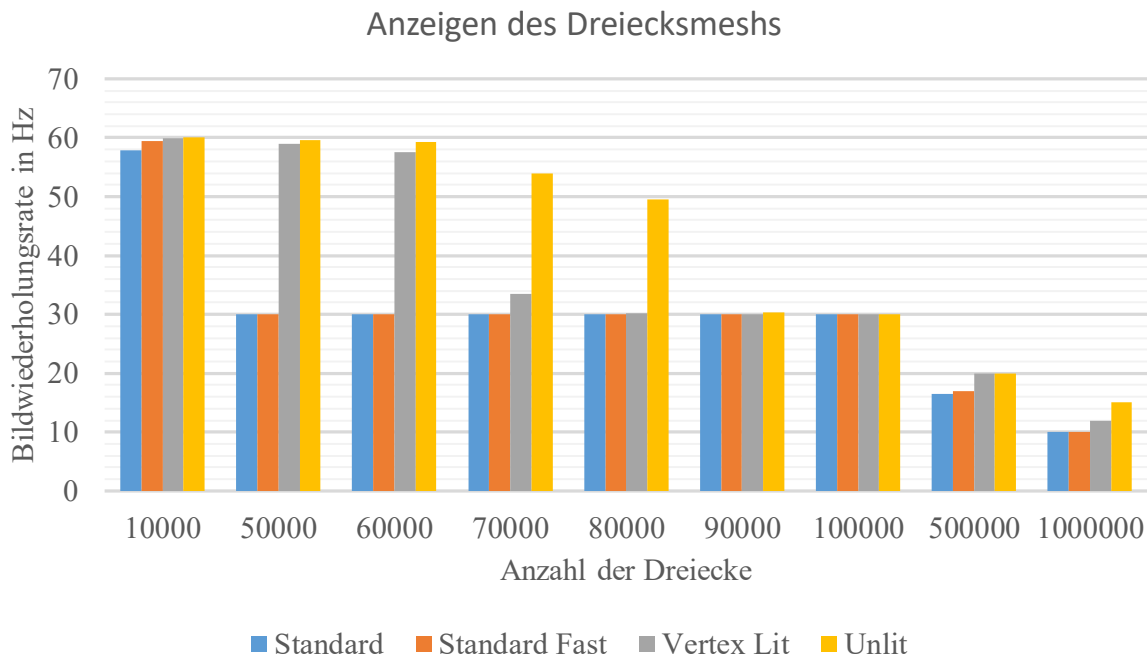


**Abbildung 3.3:** Ergebnisse des Benchmark-Tests für das Szenario *Klassisches (AR/VR-) Einsatzszenario* bei verschiedenen Dreiecksanzahlen und Shadern



**Abbildung 3.4:** Ergebnisse des Benchmark-Tests für das Szenario *Verdeckung durch reale Objekte* bei verschiedenen Dreiecksanzahlen und Shadern





**Abbildung 3.5:** Ergebnisse des Benchmark-Tests für das Szenario *Anzeigen des Dreiecksmeshs* bei verschiedenen Dreiecksanzahlen und Shadern

Renderzyklus immer länger als einen Displayzyklus benötigt und die Bildwiederholungsrate sehr schnell auf 30 Hertz abfällt.

In den Ergebnissen, siehe Abbildung 3.2, zeigt sich, dass der Standard Fast Shader nicht bedeutend schneller als der Unity Standard Shader ist. Im Vergleich zu den Standard Shadern bringt der Einsatz eines einfachen Vertex Lit Shaders einen enormen Geschwindigkeitsvorteil. Der damit verbundene Qualitätsverlust ist vor allem bei großen Dreiecken sichtbar. Durch Beachtung dieser Nachteile beim Design der 3D-Modelle können diese Effekte reduziert werden. Die Aufteilung von großen Dreiecken in mehrere kleine führt zwar zu mehr Dreiecken, kann aber trotzdem den Berechnungsaufwand reduzieren wenn dafür ein einfacherer Shader verwendet werden kann.

Der Einsatz eines Unlit Shaders kann die Geschwindigkeit weiter steigern. Die Beleuchtung spielt allerdings bei der Erfassung eines Modells oder einer kompletten Szene eine besondere Rolle. Durch das Verzichten auf Beleuchtung können Strukturen und relative Positionen der Modelle trotz 3D-Sicht der Brille nicht zuverlässig erkannt werden. Aus diesem Grund sollte der Unlit Shader nur in speziellen Fällen wie zum Beispiel für die Benutzeroberfläche eingesetzt werden.

Die Anzahl der maximal darstellbaren Dreiecke reicht von 60.000 für den Standard Shader bis zu 100.000 für den Unlit Shader. Die geringe Anzahl bedeutet nicht, dass nur so viele Dreiecke in der Szene sein dürfen, sondern, dass nicht mehr Dreiecke auf einmal im Sichtfeld

angezeigt werden können. Durch verschiedene Verfahren wie zum Beispiel Level of Detail (LOD), Frustum Culling, usw. kann die Komplexität einer Szene verringert und so beherrschbar für die Grafikkarte der HoloLens werden. Allerdings führen diese Verfahren zu einer höheren Auslastung der CPU.

Beim Vergleichen der ersten beiden Szenarien fällt auf, dass die Bildwiederholungsrate im zweiten Szenario geringer ist. Die Interaktionsmodule benötigen Rechenzeit. Aus diesem Grund muss die Komplexität der Szene reduziert werden, wenn diese Module aktiv sind, siehe Abbildung 3.3.

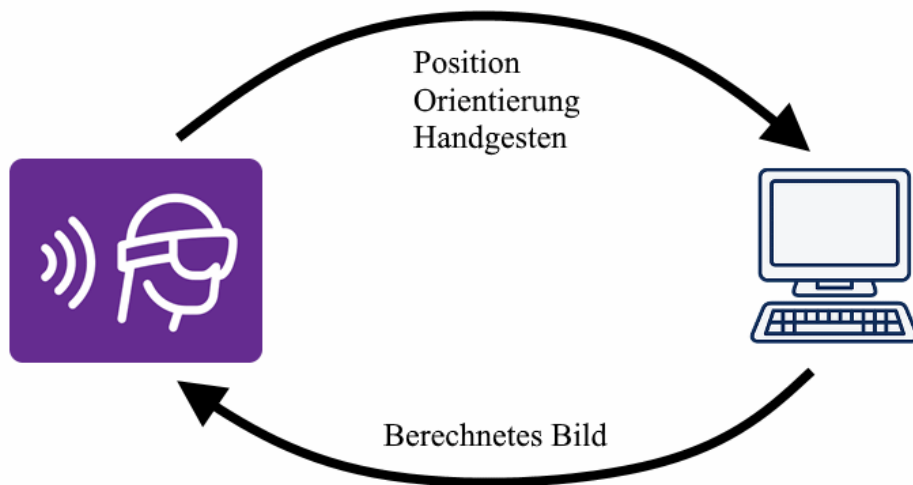
Ein besonderes Augenmerk muss dabei auf das Anzeigen des generierten Dreiecksnetzes der Umgebung gelegt werden, also Szenario drei und vier. Durch die Unvorhersehbarkeit der Größe und Komplexität des Raumes kann der Einfluss auf die Performance stark variieren. Im Vergleich zu Szenario zwei ist der Einfluss jedoch deutlich höher, siehe Abbildung 3.4 und Abbildung 3.5. Selbst in einem durchschnittlichen Büro ist die HoloLens oft nur durch das Anzeigen des Dreiecksnetzes schon ausgelastet, sodass nur wenige weitere Objekte angezeigt werden können.

#### 3.1.3 Fazit

Um die Bildwiederholungsrate auf konstanten 60 Hertz zu halten, muss bei jedem Objekt versucht werden, auf komplexe Geometrie zu verzichten und, wenn möglich, Vertex Shader oder sogar Unlit Shader zu verwenden. Das direkte Anzeigen von komplexen Modellen wie zum Beispiel einer Industrieanlage mit mehreren Millionen Dreiecken ist nicht ohne weitere Anpassungen oder Optimierungen möglich. Die HoloLens ist ein mobiles Gerät und verfügt nur über leistungsschwache Hardware im Vergleich zu einem Desktop-Computer. Der Einsatz von komplexitätsreduzierenden Algorithmen, wie sie in vielen 3D-Modellierungsprogrammen enthalten sind, können die Komplexität von Modellen reduzieren.

## 3.2 Analyse der „Holographic Remoting“ Funktionen

Um das Problem der geringen Rechenleistung zu lösen, bietet Microsoft für die HoloLens ein Holographic Remoting System an. Dieses System besteht auf der einen Seite aus der HoloLens mit aktivem Holographic Remoting Player, auf der anderen Seite aus einer Windows-Desktop-Anwendung mit Holographic-Remoting-Fähigkeiten. Die HoloLens ist dabei der Klient und der Desktop PC der Server. Der Server erhält einen Eingabestrom aus Positionsdaten und Eingabeevents von der HoloLens, berechnet ein Bild und sendet dieses an die HoloLens zurück welche dieses dann anzeigt, siehe Abbildung 3.6. Verbunden sind Server und Klient über eine WLAN-Verbindung. Durch diesen Aufbau können aufwendigere Modelle angezeigt werden, da die Komplexität nicht durch die Hardware der HoloLens beschränkt ist, sondern nur durch den verbundenen Desktop-Computer. Zusätzlich ist es durch dieses System möglich, bestehende



**Abbildung 3.6:** Schematische Darstellung des Holographic-Remoting-Systems. Die Berechnung des Bildes auf dem Desktop-PC entlastet die HoloLens.

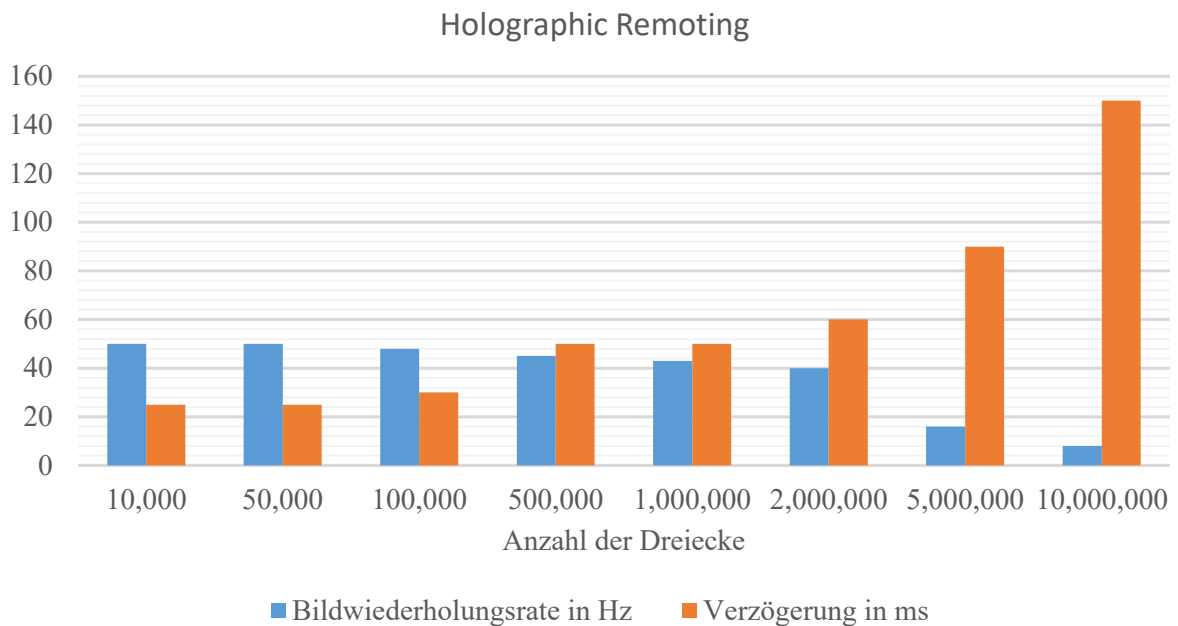
3D-Anwendungen, wie zum Beispiel CAD-Programme, um holographische Funktionen zu erweitern, ohne eine eigenständige neue Anwendung nur für die HoloLens zu entwickeln. Solche Einsatzszenarien sind vor allem für Produktionsstraßen oder sonstige Industrieanlagen interessant, da die dort verwendeten CAD-Modelle zu komplex für die HoloLens sind.

### 3.2.1 Qualität des holographischen Remoting

Wie im Abschnitt *Evaluierung der Grafikleistung* beschrieben ist die Bildwiederholungsrate und die damit verbundene Verzögerung zwischen dem Messen der aktuellen Position und dem korrekten Anzeigen der Grafiken kritisch. Bei einer Bildwiederholungsrate von 60 Hertz ergibt sich eine Verzögerung von ungefähr 16 Millisekunden für eine klassische holographische Anwendung.

$$t_{\text{Verzögerung}} = \frac{1}{60\text{Hz}} = 16,66\text{ms}$$

Durch die Kommunikation über das Netzwerk beim Holographic Remoting hängt die Verzögerung nun hauptsächlich von der Zeit zwischen dem Senden der Eingabeströme und Empfangen des anzuzeigenden Bildes ab. Microsoft gibt die Verzögerung beim Holographic Remoting mit durchschnittlich 50 Millisekunden an. In einem Blogpost von John Carmack [Car14] über Virtual Reality werden 20 Millisekunden Verzögerung als nicht bemerkbar und 50 Millisekunden als gerade noch akzeptable Verzögerung beschrieben.



**Abbildung 3.7:** Ergebnisse des Benchmark-Tests unter der Verwendung von Holographic Remoting. Die Verzögerung und die Bildwiederholungsrate sind entkoppelt und müssen getrennt betrachtet werden.

#### Messung der Verzögerung

Im Folgenden wird die Verzögerung und Bildwiederholungsrate für unterschiedliche Einsatzszenarien gemessen. Dazu steht im Holographic Remoting Player von Microsoft eine Option zur Verfügung, mit der diese Größen gemessen werden können. Es wird das klassische (AR/VR-) Einsatzszenario aus dem Abschnitt *Evaluierung der Grafikleistung* verwendet. Es wird nur eine Testreihe für verschiedene Dreiecksanzahlen unter Verwendung des Unity Standard Shaders aufgenommen. Dabei wird jede Messung 60 Sekunden ausgeführt und im Anschluss die durchschnittliche Bildwiederholungsrate und Verzögerung ausgegeben. Die Ergebnisse sind in Abbildung 3.7 dargestellt. Der verwendete Desktop-Computer besitzt einen I7-6700K Prozessor, eine NVIDIA GeForce GTX 1080 Grafikkarte und 32GB Arbeitsspeicher.

Die Verzögerung und Bildwiederholungsrate ist maßgeblich von der Qualität des drahtlosen Netzwerks abhängig. Aus diesem Grund sind verlässliche Messungen sehr schwierig. Die dargestellten Werte schwanken während der Messung um bis zu 30%. Wie in Abbildung 3.7 zu sehen ist, bleibt die Latenz bis zu einer Millionen Dreiecken unter der imaginären Grenze von 50 Millisekunden. Die Bildwiederholungsrate steigt während des Tests nie über 50Hz. Im Vergleich zu klassischen HoloLens-Anwendungen ist beim Holographic Remoting die Bildwiederholungsrate niedriger und die Latenz höher. Durch die Unzuverlässigkeit der drahtlosen Verbindung friert das Bild ab und zu ein. Während diesen kurzen Zeitspannen werden die 3D-Objekte nicht mehr korrekt in der Welt platziert.

### 3.2.2 Holographic Remoting in der Entwicklungsumgebung *Unity*

Zu einer Einführung in Unity siehe Abschnitt 4.4.1. In Unity ist zum Zeitpunkt dieser Arbeit die Holographic Remoting Funktionalität eingebaut. Diese funktioniert allerdings nur im Vorschau-Modus, das heißt nur während der Entwicklung der Software. Dabei dient es hauptsächlich als Werkzeug, um während des Testens den langwierigen Bereitstellungsprozess vom Computer auf die HoloLens zu verkürzen. Es kann bis jetzt in Unity keine Produktivanwendung gebaut werden, die die Holographic-Remoting-Funktionalität unterstützt. Weitere Einschränkungen sind in der folgenden Liste zusammengefasst.

- Spracherkennung wird zurzeit nicht unterstützt.
- Bilder werden durch eine Kamera des Server-PCs aufgenommen.
- Während des Remotings wird die komplette Audioausgabe des Server-PCs auf die HoloLens umgeleitet.

### 3.2.3 Vorteile und Nachteile des Holographic Remoting

Im Folgenden werden die Vorteile und Nachteile des Holographic Remoting gegenübergestellt.

#### Vorteile

- Durch die bedeutend höhere Rechenleistung des Desktop-PCs im Vergleich zur HoloLens können komplexere Szenen dargestellt werden. Effekte wie Schatten auf den 3D-Objekten, Kantenglättung, usw. sorgen für eine realistischere und glaubhaftere Darstellung der 3D-Objekte.
- Bestehende Anwendungen können um holographische Fähigkeiten erweitert werden. Es muss keine komplett eigenständige Anwendung entwickelt werden.
- Externe Bibliotheken, die in einer HoloLens Anwendung verwendet werden müssen auf das UWP Framework aufbauen. Da beim Holographic Remoting die Anwendung auf dem Desktop-Computer ausgeführt wird gilt diese Einschränkung nicht mehr und es kann eine größere Breite von Bibliotheken verwendet werden.
- Rechenintensive Algorithmen müssen nicht ausgelagert werden, sondern können direkt auf dem Desktop-Computer berechnet werden.

#### Nachteile

- Die Verzögerung und Bildqualität hängt vom WLAN ab. Ist das WLAN überlastet oder wird die Verbindung schlechter, kann die Anwendung nicht mehr verwendet werden.
- Die HoloLens ist nicht mehr unabhängig von zusätzlicher Hardware. Um die Anwendung zu starten ist immer ein zusätzlicher PC nötig.
- Die Verzögerung ist schlechter als bei reinen HoloLens-Anwendungen. Durch die Unzuverlässigkeit der drahtlosen Verbindung kann keine gute Qualität garantiert werden. Außerdem stehen die 3D-Objekte weniger stabil in der realen Welt.
- In Unity ist die Verwendung nur mit den zuvor genannten Einschränkungen möglich.

Da der große Vorteil der HoloLens die Unabhängigkeit von zusätzlicher Hardware ist, wird der Prototyp in dieser Arbeit als klassische HoloLens Anwendung implementiert. Zusätzliche Kriterien sind die Einschränkungen im Funktionsumfang bei Verwendung von Holographic Remoting und Unity.

### 3.3 Analyse des „Holographic Sharing“

Beim Holographic Sharing geht es darum, dass mehrere Personen durch mehrere HoloLens Brillen die gleichen 3D-Objekte sehen. Dadurch ergeben sich für die Augmented-Reality-Technologie viele neue Einsatzfelder in unterschiedlichsten Gebieten. Viele Anwendungen profitieren davon, sie mit mehreren Personen zu teilen. So können zum Beispiel Architekten über ein Haus diskutieren und dabei auf natürliche Weise auf die 3D-Objekte zeigen wie bei einem realen Objekt. Auch in dem in dieser Arbeit entwickelten Planungstool bringt der Einsatz von Holographic Sharing einen Mehrwert, da so mehrere Personen gemeinsam eine Konfiguration einer Produktionsstraße analysieren und optimieren können.

Zur Klassifizierung der unterschiedlichen Einsatzgebiete steht folgende Liste zur Verfügung [Mic16d].

- **Wie sieht die Zusammenarbeit aus?** Sehen alle Teilnehmer das gleiche oder existieren unterschiedliche Rollen.
- **Wie viele Personen sind beteiligt?** Arbeiten nur zwei Personen zusammen? Ist es eine kleine Gruppe (weniger als sieben Personen) oder eine große Gruppe (größer als sieben Personen)?
- **Wo befinden sich die einzelnen Teilnehmer?** Sind alle in einem Raum oder auf viele Räume verteilt? Oder ist es eine Mischform aus beidem? Abhängig davon müssen Avatare für die Personen in anderen Räumen eingeblendet werden.

- **Was für Geräte werden verwendet?** Benutzt jeder der Teilnehmer eine Microsoft HoloLens oder werden unterschiedliche Geräte wie Virtual Reality Brillen, Desktop Computer und Handys eingesetzt?

Abhängig von diesen Fragen wird die Art des Holographic Sharing klassifiziert und ein Szenario entwickelt.

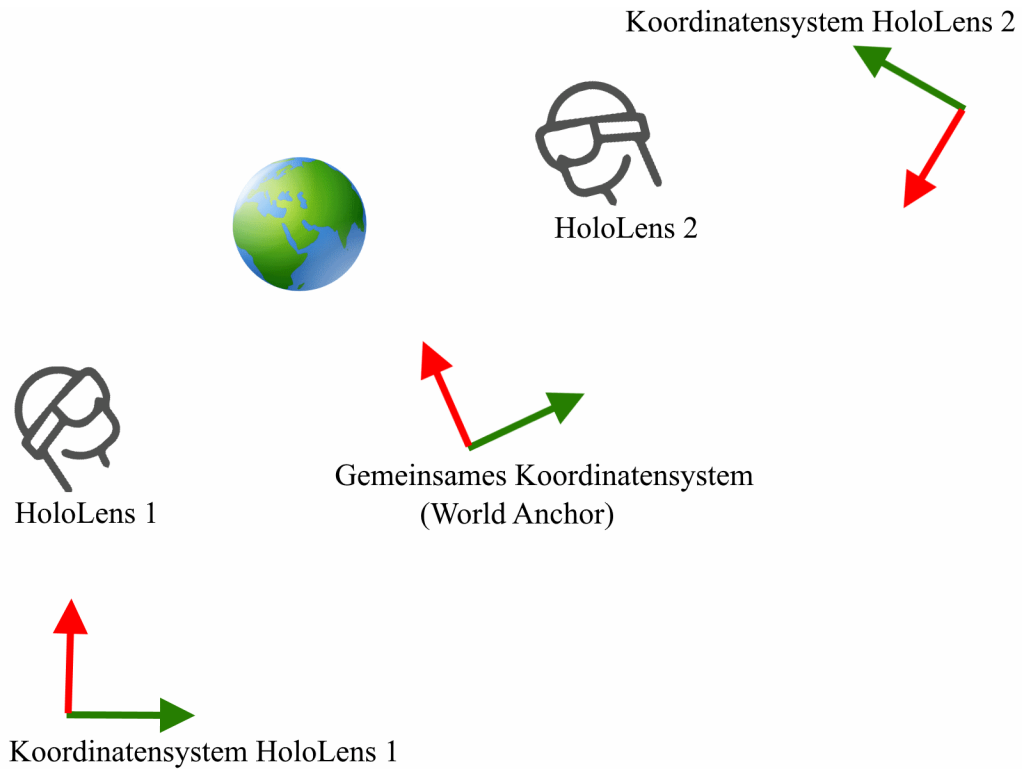
Im Rahmen dieser Arbeit wird nur auf Szenarien eingegangen, bei denen sich die Personen im gleichen Raum befinden und alle eine Augmented-Reality-Brille verwenden.

### 3.3.1 Technologie

Um in mehreren Geräten die gleichen 3D-Objekte an derselben Stelle im Raum anzuzeigen müssen die Koordinatensysteme der Geräte synchronisiert werden. Ohne diesen Abgleich wäre die Position X auf Gerät A an einem anderen Ort als auf Gerät B. Zur Synchronisation werden auf der HoloLens sogenannte World Anchor (deutsch: Weltanker) verwendet. Diese Anker können nicht verschoben werden und bleiben in der realen Welt an einer festen Position. In den Koordinatensystemen der Geräte A und B können sie allerdings ihre Position ändern. Um die World Anchor auf mehreren Geräten abzugleichen kann dieser exportiert werden. Dazu werden Informationen über die relative Position des Ankers im Raum abgespeichert. Diese Daten werden über das Netzwerk an alle anderen Geräte verteilt und dort importiert. Nun haben alle Geräte ein gemeinsames Koordinatensystem und müssen nur noch relative Position zu diesem World Anchor austauschen. Wie in Abbildung 3.8 zu sehen ist, befindet sich der Globus relativ zu den Koordinatensystemen der Geräte an unterschiedlichen Positionen. In Bezug auf das gemeinsame Koordinatensystem ist der Globus an derselben Position.

## 3.4 Markerbasierte Objekterkennung mit der Microsoft HoloLens

In diesem Abschnitt werden die Möglichkeiten der HoloLens im Bezug auf markerbasierte Objekterkennung evaluiert. In einem ersten Schritt wird die Qualität der Kamera getestet. Anschließend werden zwei Objekterkennungsbibliotheken verglichen. Ziel ist es zu evaluieren, in welcher Form markerbasierte Verfahren in einer HoloLens-Anwendung eingesetzt werden können. Die Markererkennung kann verwendet werden, um in Industrieanlagen kontextbezogene Daten zu einer Anlage einzublenden. Durch die Marker weiß die HoloLens, wo die Anlage steht und kann so virtuelle Elemente in der realen Anlage anzeigen.



**Abbildung 3.8:** 2D-Skizze der Funktionsweise des Holographic Sharing mit zwei HoloLens Brillen. Zu sehen sind die unterschiedlichen Koordinatensysteme die zum Einsatz kommen.

#### 3.4.1 Test der HoloLens Kamera

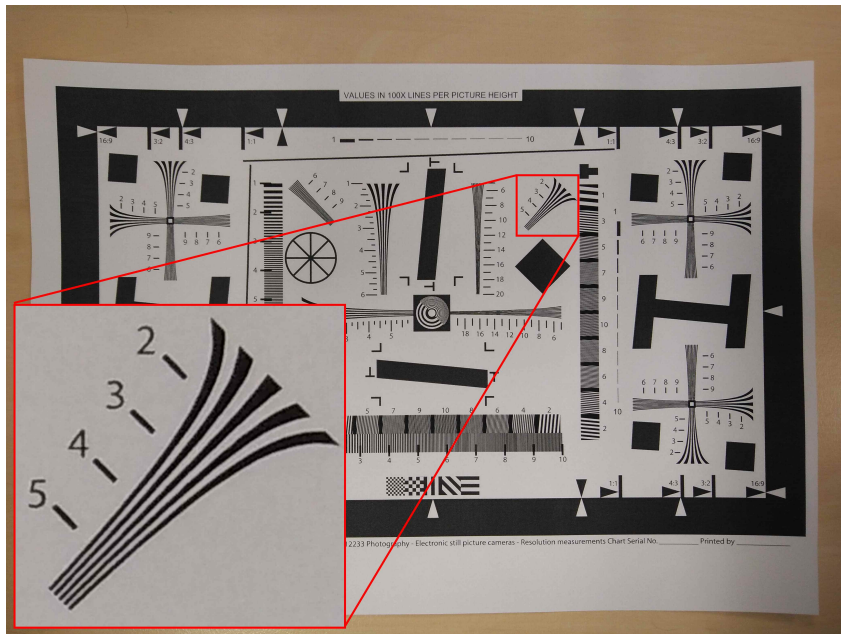
Zum Test der HoloLens Kamera wird ein ISO 12233-Testbild verwendet und das Ergebnis mit einer 13 Megapixel Handy Kamera verglichen, verwendetes Handy: *One Plus X*. Wie in Abbildung 3.9 und Abbildung 3.10 zu sehen ist, schneidet die HoloLens schlechter ab. Durch die geringe Auflösung von nur 1408 mal 792 Pixeln können deutlich weniger Details abgebildet werden. Das Bild wird schnell unscharf, was vor allem bei weit entfernten Markern zu Problemen bei der Erkennung führt.

#### 3.4.2 ARToolkit

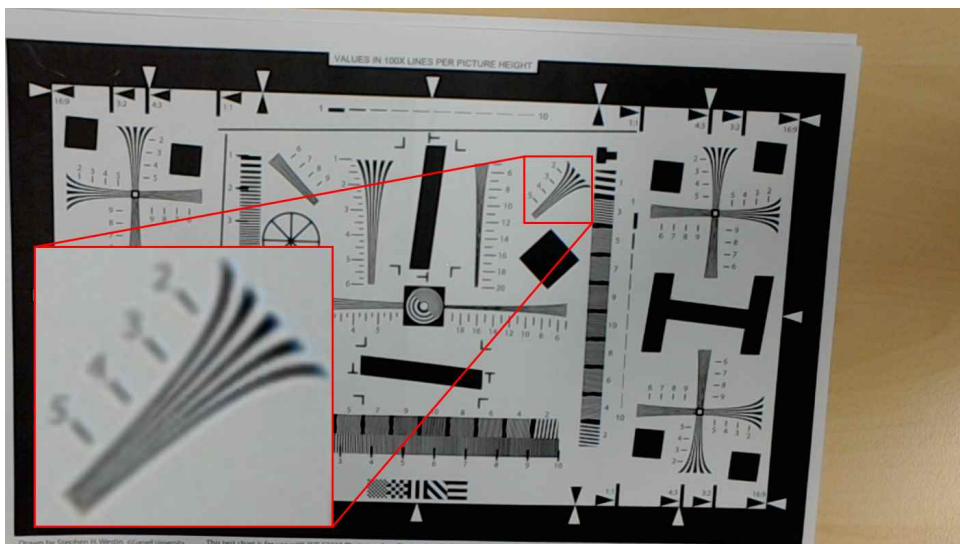
Das ARToolkit [ART16] existiert seit 1999 und ist eine Open-Source Bibliothek. Zu ihren Features zählt unter anderem das Erkennen von einfachen quadratischen Markern. Die Bibliothek erlaubt die Erkennung mit Hilfe von einfachen Kameras und Stereokameras. Mitgeliefert werden Tools zur Kamerakalibrierung und zur Markererzeugung.



### 3.4 Markerbasierte Objekterkennung mit der Microsoft HoloLens



**Abbildung 3.9:** ISO 12233 Testaufnahme mit einem Handy. Zu sehen ist, dass das Bild auch komplexe Details scharf abbildet.



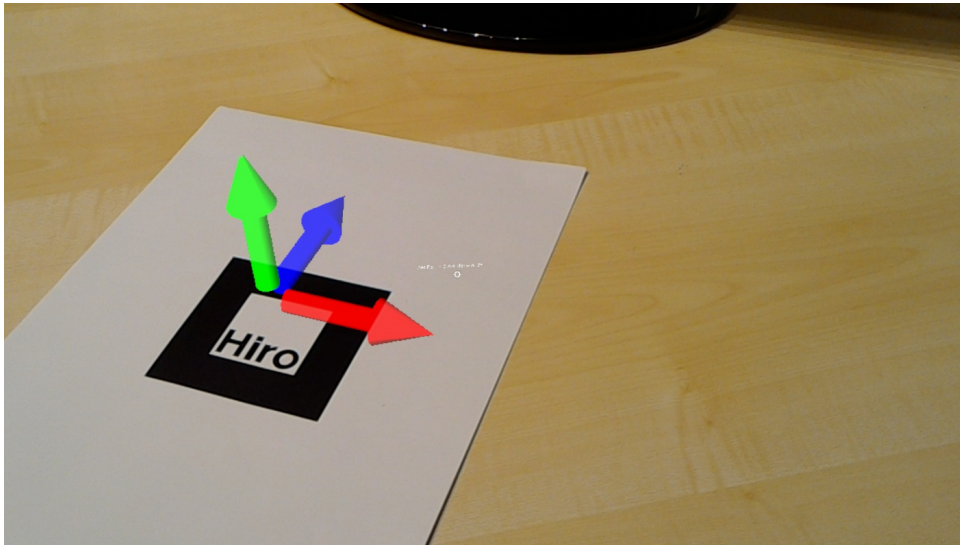
**Abbildung 3.10:** ISO 12233 Testaufnahme mit der Microsoft HoloLens. Die Aufnahme ist durch die deutlich geringere Auflösung der HoloLens Kamera verschwommen.

Zum Zeitpunkt dieser Masterarbeit bietet das ARToolkit keine offizielle Unterstützung der HoloLens an. Es existiert eine inoffizielle Portierung auf das UWP Framework, die auch auf der HoloLens funktioniert. Um das ARToolkit auf der HoloLens zu verwenden muss eine Kamerakalibrierung durchgeführt werden. Auch die Bildaufnahme ist nicht im ARToolkit enthalten und muss selbst entwickelt werden. In der HoloLens wird mit aufgenommenen Bildern eine Transformationsmatrix mitgeliefert. In dieser ist die Position und Rotation der Kamera zum Aufnahmezeitpunkt abgespeichert. Mit diesen Daten kann nach der Erkennung eines Objektes durch das ARToolkit die Position vom Kamerakoordinatensystem in das Weltkoordinatensystem umgerechnet werden und anschließend korrekt in der virtuellen Szene platziert werden.

Für diesen Test wird eine Auflösung der HoloLens Kamera von 1408 mal 792 Pixeln verwendet. Das ARToolkit benötigt auf der HoloLens ca. 40 Millisekunden um ein Bild dieser Größe nach Markern zu untersuchen. Damit wären theoretisch Aktualisierungsraten von  $1/40ms = 25Hz$  möglich. Der limitierende Faktor liegt aber bei der Aufnahme der Fotos. Die Bilder mit Lokalisierung werden nur im Fotomodus und nicht im Videomodus bereitgestellt. Aus diesem Grund liegt die maximale Aufnahmegeschwindigkeit bei circa vier Bildern pro Sekunde. Bei einer geringeren Auflösung von 896 mal 504 Pixeln dauert die ARToolkit-Berechnung nur 20 Millisekunden. Die Aufnahmegeschwindigkeit liegt aber auch hier nur bei fünf Bildern pro Sekunde. Die Qualität der Objekterkennung ist durch die geringere Auflösung aber schlechter.

Der verwendete Marker hat eine Größe von acht mal acht Zentimetern. Wie in Abbildung 3.11 zu sehen ist, wird der Marker erkannt und das Dummy-Objekt korrekt ausgerichtet und platziert. Die Präzision ist dabei nicht perfekt. Das Objekt wird nicht genau auf dem Marker positioniert, sondern leicht nach oben und zur Seite versetzt. Die Verschiebung beträgt pro Meter Abstand zum Marker circa einen Zentimeter. Die einfache Struktur der Marker erlaubt auch eine Erkennung in größeren Entfernungen. Der nur acht Zentimeter große Marker kann aus Entfernungen von bis zu fünf Metern erkannt werden. Diese einfachen Strukturen besitzen aber den Nachteil, dass sie nur über wenige Features verfügen und deshalb die Genauigkeit und Zuverlässigkeit der Erkennung gering ist. Aus diesem Grund muss die errechnete Position gefiltert werden, bevor diese weiterverwendet wird. Eine Möglichkeit wäre das Tiefpassfiltern der Position im Weltkoordinatensystem. Für Marker die sich nicht oder nur selten bewegen können so stabile Erkennungen realisiert werden.

Bei sich bewegenden Markern kommt die Erkennung mit Hilfe des ARToolkit an die Grenzen. Durch die geringe Wiederholungsrate und die nötigen Filter sind bewegte Marker nur schwer zu verfolgen. Wichtig ist außerdem eine ausreichende Beleuchtung der Marker, um die Verschlusszeit der Kamera gering zu halten. Da ein Mensch seinen Kopf nicht ruhig hält, kommt es sonst zu Verwischungen im Bild, was die Erkennung erschwert.



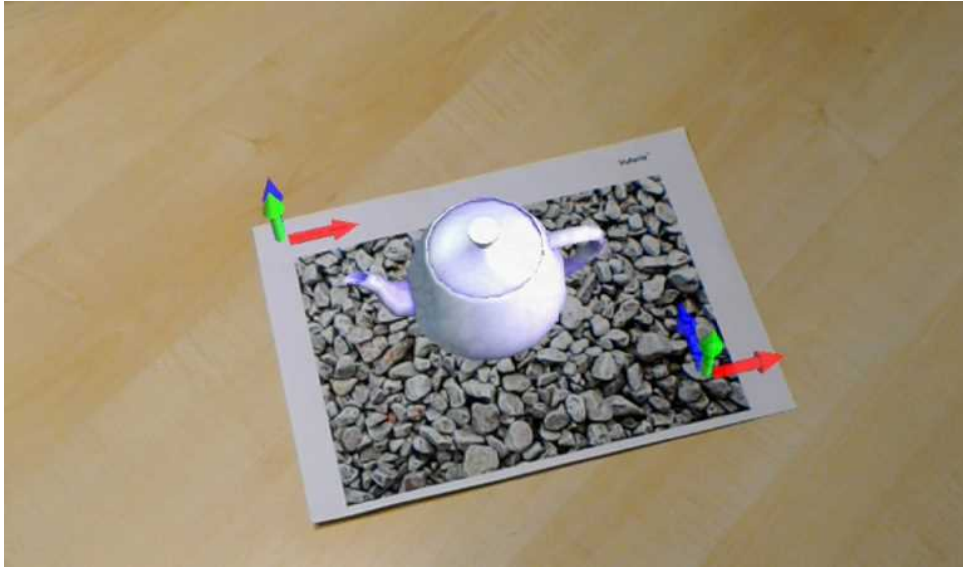
**Abbildung 3.11:** Erkennung eines Hiro Markers mit Hilfe der Microsoft HoloLens und dem ARToolkit. Das Dummy-Objekt wird nicht perfekt auf dem Marker positioniert.

### 3.4.3 Vuforia

Vuforia ist eine kommerzielle Bibliothek für das Erstellen von Augmented Reality Anwendungen. Während der Entwicklung ist die Anwendung kostenlos. Wird aber eine Anwendung ausgeliefert muss eine Lizenz erworben werden. Neben der Erkennung von natürlichen Markern können auch spezielle VuMark Marker erstellt und erkannt werden. Mit Einschränkungen ist auch die Erkennung von 3D-Objekten möglich. Zusätzlich existieren Möglichkeiten zur Spracherkennung und zum Rekonstruieren von 3D-Umgebungen.

Vuforia bietet offiziell Support für die Microsoft HoloLens an. Das Vuforia-Framework übernimmt die Aufnahme von Bildern und die Umrechnung der Markerposition von Kamerakoordinatensystem in das Weltkoordinatensystem und ist deshalb ohne weiteren Aufwand einsatzbereit. In Vuforia ist die Kamerakalibrierung bereits vorgenommen und muss nicht mehr durchgeführt werden. Vuforia erreicht eine Aktualisierungsrate von zehn Hertz. Da die berechneten Positionen nicht gefiltert werden, wirken die Bewegungen trotzdem abgehakt und stockend. Durch einen nachgeschalteten Filter, zum Beispiel Bayes-Filter der neben der Weltkoordinaten auch die Geschwindigkeit einbezieht könnten flüssige Bewegungen erzeugt werden.

In Abbildung 3.12 ist die Erkennung eines Natural-Feature-Markers mit Hilfe von Vuforia dargestellt. Der Natural-Feature-Marker ist 25 Zentimeter breit. Dieser Marker besitzt sehr viele Features und kann deswegen auch bei teilweiser Verdeckung erkannt werden. Die Erkennung ist dadurch zuverlässig und stabil. Der Offset in Abbildung 3.12 wirkt durch die Aufnahme größer, als er in Wirklichkeit ist. Beim Betrachten der Szene durch die HoloLens entsteht ein



**Abbildung 3.12:** Erkennung eines Natural Feature Markers mit Vuforia. Durch die HoloLens gesehen ist der Fehler geringer als in dieser Aufnahme.

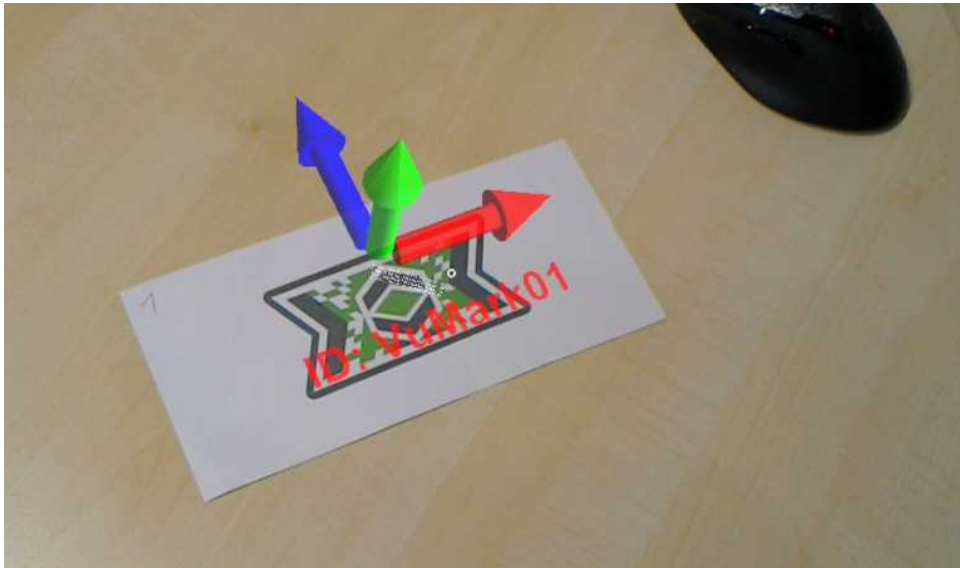
Fehler von circa einem halben Zentimeter. Die Erkennung ist bis zu einer Entfernung von nur 2,5 Metern möglich. Bei langsamen Bewegungen können auch bewegte Marker zuverlässig verfolgt werden. Allerdings sind die Bewegungen stockend.

VuMark-Marker sind frei designbare Marker mit integrierter ID, die aber Designregeln folgen müssen, um trotzdem einfach zu Erkennen zu sein. In Abbildung 3.13 ist die Erkennung eines VuMark-Markers mit Hilfe von Vuforia dargestellt. Durch die feinen Strukturen im Vergleich zu Matrix Markern im ARToolkit können diese nicht von großer Entfernung erkannt werden. Der VuMark Marker in Abbildung 3.13 ist zehn Zentimeter groß und kann erst ab einer Entfernung von einem Meter erkannt werden. Dafür ist die Positionierung zuverlässig.

#### 3.4.4 Vergleich von Vuforia und dem ARToolkit

Im Folgenden werden die Unterschiede der beiden Bibliotheken aufgelistet.

<b>ARToolkit</b>	<b>Vuforia</b>
Open-Source Bibliothek	Kommerzielle Anwendung
Matrix Marker	Natural Feature Tracking und VuMark Marker
Erkennung aus größerer Entfernung	Nur kurze Entfernungen
Höhere Zuverlässigkeit und Stabilität	Höhere Aktualisierungsrate
	Einfacher zu verwendende Bibliothek



**Abbildung 3.13:** Erkennung eines VuMark Markers mit Vuforia. Durch die HoloLens gesehen ist der Fehler geringer als in dieser Aufnahme.

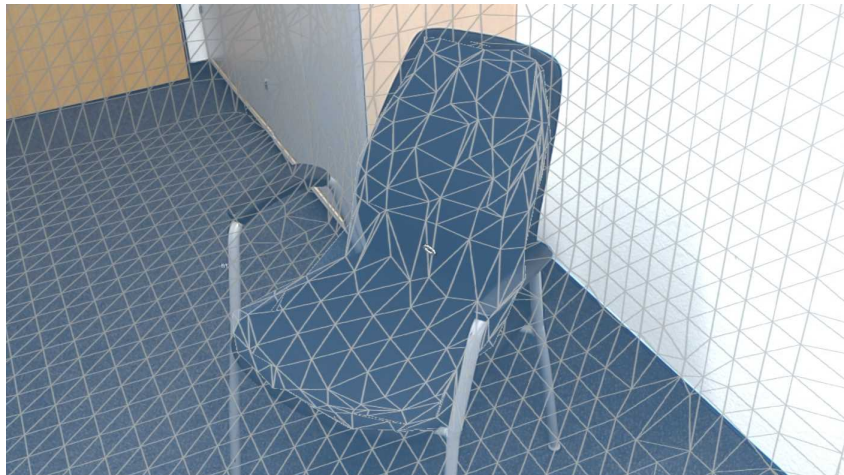
### 3.5 Das dreidimensionale Umgebungsmodell

Zur Raumerkennung verwendet die HoloLens mehrere Sensoren. Die insgesamt vier umgebungsverstehenden Kameras in Verbindung mit der inertialen Messeinheit bestimmen die Position der Brille im Raum. Aufbauend darauf wird mit Hilfe der Infrarot-Tiefenkamera ein 3D-Drahtgitter-Umgebungsmodell erstellt. Ein Beispiel für ein solches Umgebungsmodell ist in Abbildung 3.14 dargestellt. Die Tiefenkamera wird auch zur Handgestenerkennung verwendet. Durch die verwendeten Technologien (Farbkamera und Infrarotkamera) ergeben sich mehrere Bedingungen und Probleme.

- Die Benutzung der Brille in einem komplett dunklen Raum ist unmöglich da die Kameras auf Umgebungslicht angewiesen sind.
- Da sichtbares Licht und Infrarot-Licht verwendet wird, sind alle Gläser, wie zum Beispiel Fenster, nicht erkennbar. Auch alle spiegelnden oder reflektierenden Objekte wie Bilderrahmen, metallische Objekte, LCD Displays, Spiegel, usw. werden überhaupt nicht oder nur schlecht erkannt.
- Die Tiefenkamera hat eine maximale Reichweite von 3,20 Metern. Veränderungen im Umgebungsmodell werden nur in diesem Bereich erkannt.



**Abbildung 3.14:** Darstellung des von der HoloLens aufgenommenen Umgebungsmodells eines Stuhls als Drahtgitter.



**Abbildung 3.15:** Darstellung des Umgebungsmodells eines Stuhls aufgearbeitet mit dem HoloToolkit Modul „Spatial Understanding“

#### 3.5.1 Qualität des erstellten Umgebungsmodells

Zur besseren Darstellbarkeit wird das Drahtgitter aus Abbildung 3.14 mit Hilfe des HoloToolkit-Moduls „Spatial Understanding“ verarbeitet und dargestellt. Das Resultat dieser Verarbeitung ist in Abbildung 3.15 zu sehen. Das Umgebungsmodell wurde strukturiert und in gleichmäßige Dreiecke unterteilt. Beim Betrachten des Umgebungsmodells fallen einige Dinge auf.

- Die Armlehnen und Stuhlbeine sind nicht im Umgebungsmodell enthalten. Objekte unter fünf Zentimetern Größe werden nicht zuverlässig erkannt. Erst ab zehn Zentimetern wird eine zuverlässige Erkennung möglich.

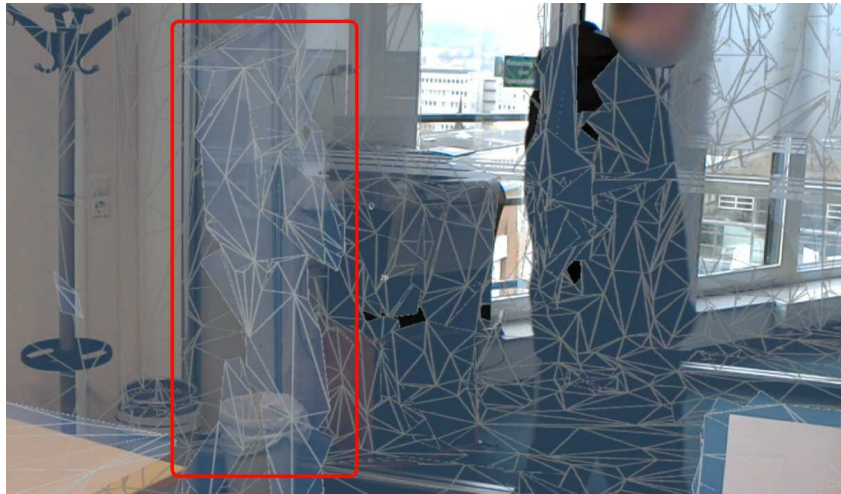


**Abbildung 3.16:** Darstellung des Umgebungsmodells eines Tisches mit mehreren Gegenständen. Zu sehen ist, dass nur größere Elemente einen Einfluss auf das Umgebungsmodell haben.

- Das Umgebungsmodell schließt nicht bündig mit dem realen Objekt ab. Es ergeben sich Messfehler von circa fünf bis zehn Zentimetern.

Diese Probleme treten auch in Abbildung 3.16 auf. Die zwei Locher und die Wasserflasche werden nicht erkannt. Nur der Stifthalter ist groß genug für eine Erkennung. Allerdings sind diese Gegenstände genau an der Grenze dessen, was im Umgebungsmodell angezeigt wird. Bei mehreren Wiederholungen der Messung werden zufällig die Objekte erkannt oder nicht erkannt. Weitere Probleme treten bei dünnen Objekten wie Türen, Stuhllehnen oder Bildschirmen auf. Werden diese nur von einer Seite betrachtet, funktioniert die Erkennung ohne Probleme. Bei einer Betrachtung der Rückseite vermutet die Erkennung Inkonsistenzen und entfernt die erkannten Flächen. Keine Probleme hat das Umgebungsmodell beim Erkennen von großen Flächen wie zum Beispiel Wänden, Böden, Decken, Tischen, usw.

Eine Schwäche der Raumerkennung sind bewegte Objekte. In Abbildung 3.17 wurde versucht, eine sich bewegende Person zu erkennen. Dabei zeigt sich, dass das Umgebungsmodell immer einige Sekunden hinter der Realität hängt. Zusätzlich ist die Qualität der Erkennung schlechter, da das Umgebungsmodell bei statischen Objekten inkrementell verfeinert wird. Bei sich bewegenden Objekten hat es dazu keine Möglichkeit, da die Erkennung immer wieder von neuem beginnt.



**Abbildung 3.17:** Erkennung einer laufenden Person mit Hilfe des HoloLens-Umgebungsmodells. Rot eingerahmt ist das verzögerte Drahtgittermodell der laufenden Person. Durch die Verzögerung der Erkennung befindet sich das Umgebungsmodell hinter der Person.

#### 3.5.2 Nutzung des Umgebungsmodells in Augmented Reality Anwendungen

Wie im vorherigen Abschnitt gezeigt, hat die Raumerkennung der Microsoft HoloLens Stärken und Schwächen. Dadurch ergeben sich verschiedene Einsatzmöglichkeiten, die geeignet oder ungeeignet sind.

Gut eignet sich die Raumerkennung für Szenarien, in denen die Trägheit und relative Ungenauigkeit des Umgebungsmodells keine Rolle spielt. Beispiele dafür sind:

- Die Verdeckung von 3D-Objekten durch reale Gegenstände. So können zum Beispiel virtuelle Objekte unter Tischen versteckt werden.
- Die Platzierung von 3D-Objekten an Wänden oder auf Tischen.
- Die Generierung eines Spielfelds in einem Jump 'n' Run Spiel.

Ungeeignet ist die Raumerkennung zur Verfolgung von Objekten, da das Umgebungsmodell zu langsam aktualisiert wird und das Objekt schnell außerhalb des Sichtfelds der Tiefenkamera gelangt. Für eine Verfolgung von Objekten eignet sich die Farbkamera besser, da mit ihr Aufnahmen in schnellerer Reihenfolge möglich sind.

Der Einsatz der Raumerkennung zur Objekterkennung oder zur Vermessung von 3D Objekten ist nur bedingt möglich. Die Genauigkeit des Drahtgitters ist zu gering, um Strukturen in Objekten abzubilden. Zur Erkennung von großen einfachen Objekten wie Quadern oder Kugeln ist die Erkennung aber ausreichend.



### 3.5 Das dreidimensionale Umgebungsmodell

---

Im Kontext dieser Arbeit eignet sich die Erkennung des Umgebungsmodells, um große Flächen wie Tische oder Fußböden zu erkennen, um auf diesen Modelle von Produktionsstraßen darzustellen. Dadurch hängen die Modelle nicht frei in der Luft, sondern werden dargestellt, als würden sie auf dem Tisch oder Fußboden stehen.



# 4 Layoutplanung von Produktionsstraßen mit der Microsoft HoloLens

Die im vorherigen Kapitel zusammengetragenen Möglichkeiten der HoloLens werden in diesem Kapitel verwendet um ein Konzept zu entwickeln. Ziel ist es den Fabriklayoutsimulator von Wörner et al. [WE11; WE13] auf die HoloLens zu portieren. Neben der direkten Portierung sollen auch Anpassungen durchgeführt werden, um die vollen Möglichkeiten der Augmented-Reality-Brille zu zeigen. Die immersive dreidimensionale Darstellung und Visualisierung soll helfen, Probleme von Fabriklayouts besser zu verstehen.

## 4.1 Fabriklayout-Simulator

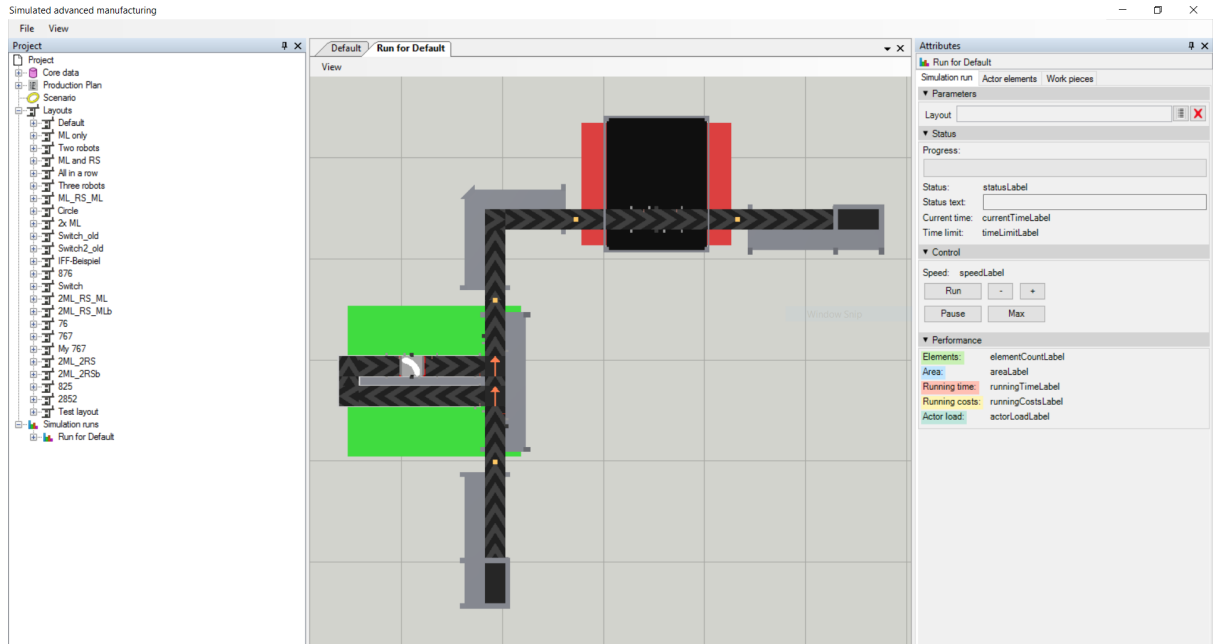
Im Folgenden wird der Fabriklayout-Simulator von Wörner et al. [WE11; WE13] vorgestellt. Er dient als Grundlage für diese Arbeit. Die verwendeten Baugruppen basieren dabei auf dem iTRAME (Intelligent Transformable Assembly and Manufacturing Equipment) System [RKK+07]. In Abbildung 4.1 sind Aufbauten dieses Systems zu sehen.

Dieses System besteht aus folgenden modularen Baugruppen:



**Abbildung 4.1:** Realer Aufbau des iTRAME-Systems am Institut für Industrielle Fertigung und Fabrikbetrieb (IFF) der Universität Stuttgart [Ins17]. Die gleichen Baugruppen können für unterschiedliche Anordnungen verwendet werden.

## 4 Layoutplanung von Produktionsstraßen mit der Microsoft HoloLens



**Abbildung 4.2:** Fabriklayout-Simulator von Wörner et al. während der Simulation eines Layouts. Auf der linken Seite befinden sich die Daten der Anwendung wie zum Beispiel die zur Verfügung stehenden Layouts. In der Mitte ist das Layout visualisiert. Im rechten Teil werden Simulationsergebnisse angezeigt.

- Transportbänder
- Handarbeitsstationen
- Bearbeitungsroboter
- Einlagerungssysteme
- Bauteilaufzüge
- Kamerastationen

Bei diesem System verfügen alle Module über standardisierte Verbindungen, sodass jedes Modul mit jedem verbunden werden kann. So können die einzelnen Baugruppen nahezu beliebig angeordnet werden.

Mit Hilfe des Fabriklayout-Simulators ist es möglich, ein iTRAME-Layout digital zu planen und anschließend zu simulieren, siehe Abbildung 4.2. Die Aufgabengebiete des Simulators können dabei in zwei wichtige Teilgebiete unterteilt werden: Analysieren eines Layouts und Optimieren eines Layouts.

### 4.1.1 Analysieren eines Layouts

Zur Analyse der Layouts gehört in erster Linie eine Darstellung der Layouts, siehe Abbildung 4.2 mitte. Dies ist im Simulator in einer zweidimensionalen Draufsicht und in einer dreidimensionalen Ansicht möglich. Neben der statischen Darstellung kann das Layout auch simuliert werden. Dabei werden die Bauteile über die Bänder befördert und an den jeweiligen Arbeitsstationen bearbeitet. Während der Simulation können Information zu den Bauteilen und Arbeitsstationen abgefragt werden, siehe Abbildung 4.2 rechts. Dazu gehören die Auslastung der Arbeitsstationen und der Verlauf der Auslastung während der Simulation, der aktuelle Zustand der Werkstücke und die Historie der Zustände, der Fortschritt der Simulation und die aktuellen Metriken des Layouts. Zur Unterstützung der Analyse werden Daten während der Simulation visualisiert, siehe Abbildung 4.2 mitte. Die Auslastung der Arbeitsstationen wird durch eine sich farblich veränderbare Bodenplatte visualisiert. Je grüner die Platte, desto ausgelasteter ist die Station. So werden überlastete oder unausgelastete Anlagen erkannt. Das Förderband wird an den Stellen, an denen die Bauteile warten müssen eingefärbt. Je öfter Bauteile in einem Abschnitt stillstehen desto rötlicher wird das Band eingefärbt. So werden Flaschenhälse in einem Layout sichtbar. Durch die Darstellung des Layouts können auch Platzverhältnisse der Anlage eingeschätzt werden.

### 4.1.2 Optimieren eines Layouts

Zur Optimierung des Layouts stehen dem Simulator zwei Möglichkeiten zur Verfügung. Zum einen kann das Layout von Hand abgeändert werden. Dazu dienen die Erkenntnisse aus dem Analyse-Schritt, um Flaschenhälse zu entdecken und Schwächen des aktuellen Layouts zu erkennen. Es können neue Elemente hinzugefügt oder gelöscht werden. Bestehende Elemente können verschoben oder rotiert werden. Außerdem können die Arbeitsschritte der einzelnen Module angepasst werden. Je nach Ziel der Optimierung (maximaler Durchsatz, minimale Kosten, minimaler Platzverbrauch) kann das Layout zielgerichtet angepasst werden.

Eine zweite Möglichkeit besteht in der automatischen Verbesserung eines Layouts. Dabei wird mit Hilfe eines evolutionären Algorithmus versucht, ein besseres Layout zu finden. Durch die fünf Metriken Kosten, Laufzeit, Auslastung der Arbeitsstationen, benötigte Fläche und Anzahl der Stationen wird eine Bewertung des Layouts durchgeführt. Anhand dieser Bewertung verbessert der Algorithmus über mehrere Generationen das Layout.

## 4.2 Anforderungen an die Produktionsstraßenplanung mit der Microsoft HoloLens

Im Folgenden werden die funktionalen und nicht funktionalen Anforderungen der Anwendung zusammengetragen. Diese Anforderungen ergeben sich aus dem Fabriklayout-Simulator von

Wörner et al. [WE11; WE13] und aus den besonderen Anforderungen einer Augmented-Reality Anwendung.

### 4.2.1 Anforderungen des Layout-Simulator

Die im folgenden beschriebenen Anforderungen beschreiben die Funktionalität, die die Anwendung bereitstellen soll.

#### Nichtfunktionale Anforderungen

- **Zusammenhänge verstehen:** Durch die Visualisierung des Layouts und der Simulation versteht der Benutzer die Zusammenhänge der Produktionsstraße.
- **Performance:** Über Visualisierungen erhält der Benutzer Rückmeldung zur Performance und Auslastung von einzelnen Modulen und zum kompletten Layout.
- **Informationstransport:** Durch die Visualisierungen kann der Benutzer Probleme und Optimierungsmöglichkeiten erkennen und auf diese durch eine Abänderung des Layouts reagieren. So werden Flaschenhälse und unausgelastete Module verhindert.

#### Funktionale Anforderungen

- **Laden von Layouts:** Die Anwendung zeigt alle verfügbaren Layouts an und lässt den Benutzer eines auswählen. Dieses wird anschließend geladen.
- **Darstellen eines Layouts:** Das aktuell geladene Layout wird dargestellt.
- **Verschieben von Elementen:** Es ist möglich, Elemente des Layouts zu verschieben. Dabei kann entweder ein einzelnes oder alle Elemente verschoben werden.
- **Rotieren von Elementen:** Es ist möglich, Elemente des Layouts zu rotieren. Dabei kann entweder ein einzelnes oder alle Elemente rotiert werden.
- **Löschen von Elementen:** Bestehende Elemente können aus dem Layout gelöscht werden.
- **Invertieren der Richtung:** Bei Elementen mit Transportband kann die Richtung dieses Bandes invertiert werden.
- **Erstellen von neuen Elementen:** Es können neue Elemente erstellt werden.
- **Ändern der Einstellungen von Elementen:** Von bestehenden Elementen können die Produktionsschritte, die dieses Element durchführt, abgeändert werden.

- **Speichern von Änderungen:** Änderungen an Layouts können persistent abgespeichert werden.
- **Simulieren von Layouts:** Es ist möglich, die Simulation eines Layouts zu starten und zu stoppen.
- **Visualisieren der Simulation:** Die einzelnen Bauteile bewegen sich während der Simulation über die Förderbänder und werden an Arbeitsstationen bearbeitet.
- **Ändern der Geschwindigkeit:** Während eine Simulation läuft, ist es möglich, die Geschwindigkeit der Simulation zu verändern.
- **Simulationsstatus:** Während einer Simulation können allgemeine Infos wie Laufzeit und Fortschritt abgefragt werden.
- **Simulationsstatus Arbeitsstation:** Während einer Simulation können Information zu den Arbeitsstationen abgefragt werden, wie Auslastung oder aktueller Zustand.
- **Simulationsstatus Bauteile:** Während einer Simulation können Information zu den Bauteilen abgefragt werden, wie Zustand, zurückgelegte Strecke oder Laufzeit.
- **Simulationsstatus Förderband:** Während einer Simulation werden Förderbandabschnitte farblich hervorgehoben, auf denen Bauteile warten müssen.

### 4.2.2 Anforderungen der Augmented-Reality Anwendung

Im Folgenden werden die Anforderung aufgelistet, die sich aus der Augmented-Reality-Umgebung ergeben.

- **Manuelles Platzieren der Grundplatte:** Da es in Augmented-Reality-Anwendungen keinen Nullpunkt gibt, muss dieser definiert werden. Dies soll durch eine Grundplatte realisiert werden, die verschoben, rotiert und skaliert werden kann.
- **Automatisches Platzieren der Grundplatte:** Die HoloLens soll die Umgebung erkennen und die Grundplatte automatisch auf eine große Fläche platzieren.
- **Skalieren des Layouts:** Es ist möglich das Layout zu skalieren, sodass es entweder als Modell oder in realer Größe angezeigt wird.
- **Verzicht auf klassische Benutzeroberflächen:** Da in Augmented-Reality Anwendungen ständig sichtbare Benutzeroberflächen, wie zum Beispiel Menüleisten, stören, soll auf diese verzichtet werden.
- **Konstant hohe Bildwiederholungsrate:** Um die Verwendung der Anwendung angenehm zu gestalten, soll eine konstant hohe Bildwiederholungsrate von 60Hz erreicht werden.

- **Erweiterung von Anlagen:** Bestehende Anlagen sollen von der HoloLens erkannt und virtuell erweitert werden.
- **Kollaboratives Arbeiten:** Mehrere Benutzer der Anwendung sollen gleichzeitig an einem Layout arbeiten können und über dieses diskutieren können.

### 4.3 Konzept

Aufbauend auf den Anforderungen im letzten Abschnitt wird ein Konzept für die HoloLens-Portierung des Layout-Simulators entwickelt. Im ersten Teil dieses Abschnitts wird die übergeordnete Struktur der Anwendung vorgestellt. In den weiteren Abschnitten werden Konzepte diskutiert, welche die Anforderungen erfüllen.

#### 4.3.1 Zustandsautomat

Wären alle Funktionalitäten in einem Zustand abrufbar wäre dieser überladen und der Benutzer würde sich schwer zurechtfinden. Aus diesem Grund wird die Funktionalität auf mehrere Zustände aufgeteilt, zwischen denen der Benutzer wechseln kann. Die grundlegende Struktur ist in Abbildung 4.3 dargestellt.

Zu Beginn befindet sich der Benutzer im Zustand „*Select plane placement method*“. Im diesem Zustand entscheidet er sich für eine Methode die Grundplatte zu platzieren. Die Grundplatte ist nötig, da eine Augmented Reality Umgebung keinen natürlichen Nullpunkt besitzt. Der Mittelpunkt dieser Grundplatte fungiert als virtueller Nullpunkt. Der Nutzer hat die Wahl zwischen automatischer oder manueller Platzierung der Grundplatte. Entsprechend wechselt er in den Zustand „*Manual plane placement*“ oder „*Automatic plane placement*“.

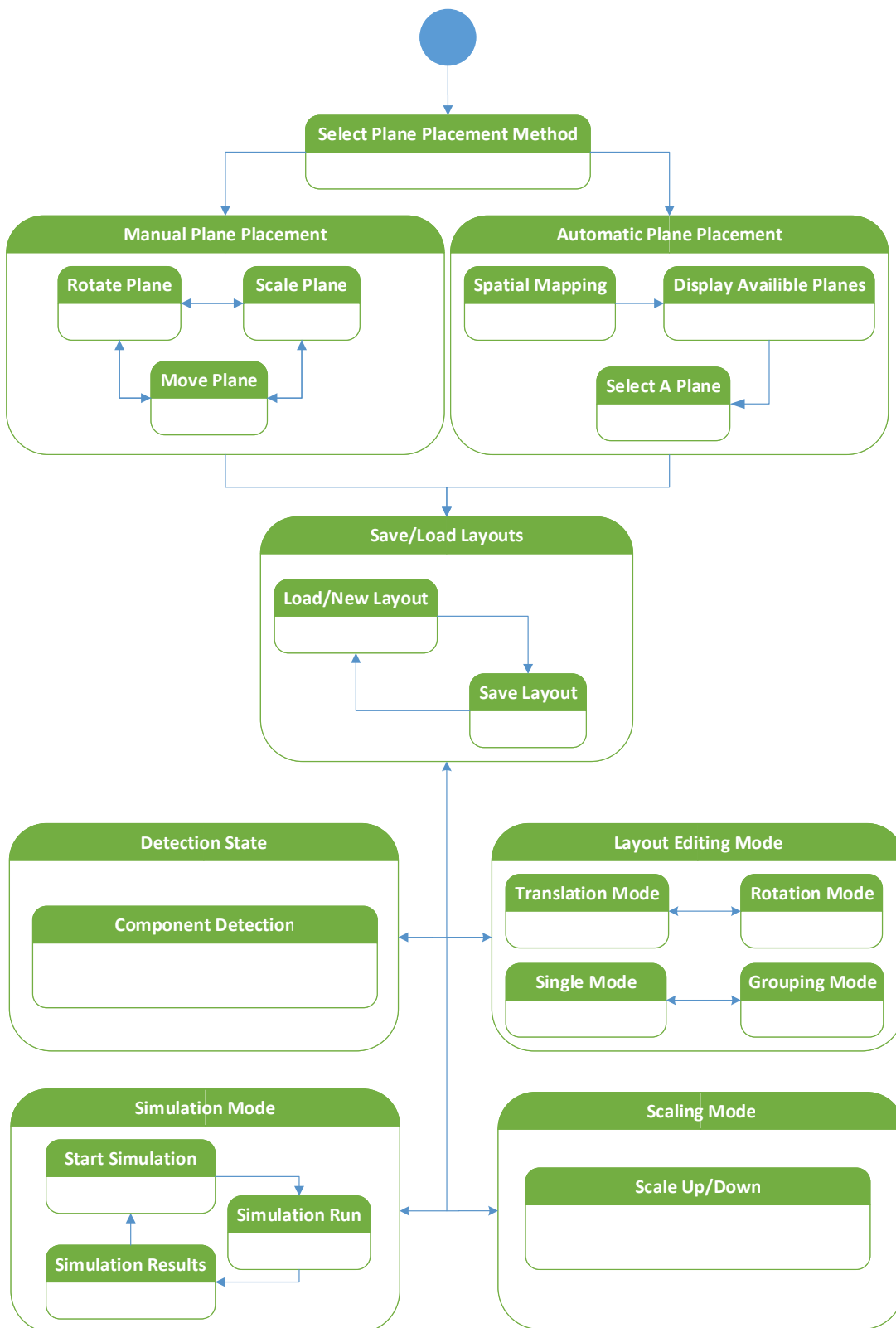
Im Zustand „*Manual plane placement*“ kann die Grundplatte verschoben, rotiert und skaliert werden. Nach einer Bestätigung der aktuellen Positionierung wird die Grundplatte gespeichert und der Zustand wechselt zu „*Layout Selection Mode*“.

Im Zustand „*Automatic plane placement*“ wird zuerst der Raum, in dem sich der Nutzer befindet, vermessen. Nach dieser Messung wird ein Algorithmus gestartet, der Flächen, wie zum Beispiel Tische, sucht. Alle gefundenen Flächen werden angezeigt und der Benutzer kann sich für eine dieser Flächen entscheiden. Diese ausgewählte Fläche wird als neue Grundplatte gespeichert und der Zustand wechselt zu „*Layout Selection Mode*“.

Die Zustände „*Layout Selection Mode*“, „*Layout Editing Mode*“, „*Scaling Mode*“, „*Simulation Mode*“, „*Detection Mode*“ sind direkt miteinander verbunden. Aus jedem dieser Zustände kann jederzeit in jeden anderen Zustand gewechselt werden.

Der Zustand „*Layout Selection Mode*“ dient entweder zum Laden eines vorhandenen Layouts oder zum Erstellen eines neuen Layouts. Dazu werden alle vorhandenen Layouts aufgelistet,





**Abbildung 4.3:** Zustandsautomat, der alle übergeordneten Zustände beinhaltet. „*Select plane placement method*“ ist der Startzustand. Zwischen den unteren fünf Zuständen kann beliebig gewechselt werden.

bei der Auswahl eines dieser Layouts wird dieses geladen und der Zustand wechselt zu „*Layout Editing Mode*“. Der gleiche Ablauf wird auch beim Erstellen eines neuen Layouts ausgeführt. Eine weitere Option ist das persistente Abspeichern der Änderungen eines Layouts.

Im Zustand „*Layout Editing Mode*“ wird das Layout bearbeitet. Dieser Zustand besitzt zwei untergeordnete Zustandsautomaten. Der erste Zustandsautomat entscheidet ob das ausgewählte Element verschoben oder rotiert wird. Der zweite Zustandsautomat beeinflusst, wie Objekte ausgewählt werden. Im „*Single Mode*“ werden immer nur einzelne Objekte ausgewählt. Im „*Grouping Mode*“ werden alle mit dem ausgewählten Element verbundenen Elemente ausgewählt. Zusätzlich besteht die Möglichkeit, Elemente zu löschen, neue Elemente hinzuzufügen und von bestehen Elementen die Richtung des Transportbandes zu invertieren.

Der Zustand „*Simulation Mode*“ dient der Simulation des Layouts. Simulation können gestartet und gestoppt werden. Die Geschwindigkeit der Simulation kann angepasst werden. Während der laufenden Simulation und nach Abschluss der Simulation werden Simulationsergebnisse visualisiert.

Im Zustand „*Scaling Mode*“ kann das dargestellte Layout skaliert werden. Es existieren Kommandos zum Vergrößern und Verkleinern. Zusätzlich kann das Layout auf Originalgröße skaliert werden.

Im Zustand „*Detection Mode*“ werden reale Produktionsstraßen erkannt und in das aktuell geladene Layout integriert. So können bestehende Anlagen virtuell erweitert werden.

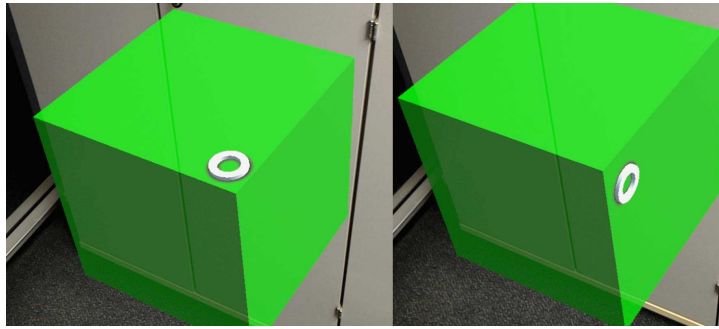
### 4.3.2 Feedback beim Interagieren mit der HoloLens

#### Blickrichtung

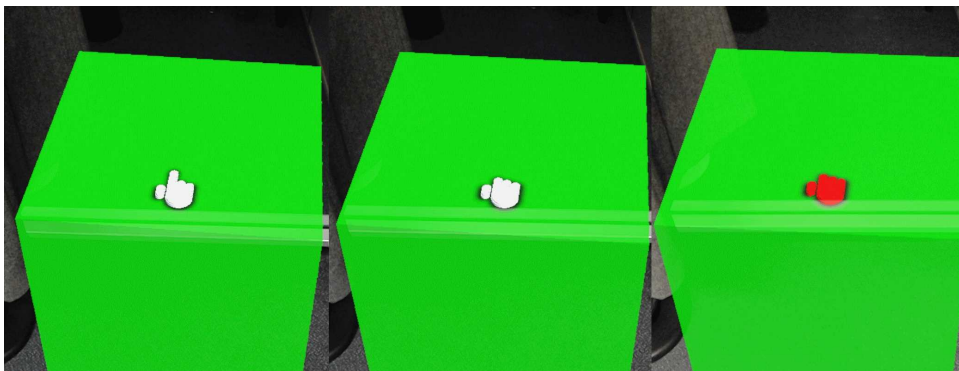
Um eine Rückmeldung über die Blickrichtung zu erhalten, wird ein *Cursor* eingeblendet. Der Cursor wird auf dem ersten Objekt, das mit dem Blickrichtungsstrahl kollidiert, platziert. Wenn der Strahl mit keinem Objekt kollidiert, wird der Cursor in einer Standardentfernung angezeigt. In Abbildung 4.4 ist zu sehen, dass sich der Cursor auf dem ersten Objekt befindet und an der Oberfläche des Objektes ausgerichtet wird.

#### Handgesten

Ein Problem von Handgesten im Vergleich zu Maustasten oder Tastaturtasten ist die fehlende Rückmeldung beim Ausführen dieser Gesten. Vor allem unerfahrene Benutzer können so leicht frustriert werden, da sie nicht wissen, warum die Geste zu keiner Reaktion der Anwendung führt. Da haptisches Feedback wie beim Betätigen einer Taste nicht möglich ist, wird audiovisuelles Feedback verwendet. Die Air-Tap-Geste beginnt mit der Erkennung des gestreckten Zeigefingers. Um den Benutzer in dieser Situation mitzuteilen, dass seine Hand erkannt wird, verändert sich der Cursor von seiner ursprünglichen Form in eine Hand mit gestrecktem



**Abbildung 4.4:** Der Cursor wird auf dem fokussierten Objekt dargestellt. Dabei wird der Cursor an der Oberfläche des Objekts ausgerichtet.



**Abbildung 4.5:** Die Rückmeldung des Cursors während Handgesten durchgeführt werden. Links: Hand erkannt. Mitte: Air-Tap Geste erkannt. Rechts: Je Rötlicher der Cursor ist, desto näher befindet sich die Hand am Rande der Erkennung.

Zeigefinger, siehe Abbildung 4.5 links. Wird der Finger gesenkt, verändert sich der Cursor zu einer Hand mit gesenktem Zeigefinger, siehe Abbildung 4.5 mitte. Zusätzlich wird ein Ton wie beim Betätigen einer Taste abgespielt. Wird der Finger wieder gehoben, ändert sich der Cursor entsprechend und es wird ein Ton wie beim Loslassen einer Taste abgespielt. So kann der Benutzer während der Durchführung der Geste genau erkennen, ob die Geste erkannt wurde, oder welche Teile nicht erkannt wurden.

Während der Manipulation eines Objektes kann es passieren, dass sich die Hand aus dem Gestenerkennungsbereich herausbewegt. Um dies zu verhindern, wird der Cursor immer rötlicher, je näher die Hand an den Rand der Erkennung herankommt, siehe Abbildung 4.5 rechts. So hat der Nutzer die Chance, dies zu erkennen und seine Bewegung anzupassen.

### 4.3.3 Platzieren einer Grundplatte

Wird eine 3D-Anwendung auf der HoloLens gestartet, wird der Nullpunkt des Weltkoordinatensystems an die Position der HoloLens gesetzt. In den meisten Fällen sollen die 3D-Objekte

aber abhängig von den realen Objekten in der Umgebung und nicht abhängig von der HoloLens platziert werden. Aus diesem Grund wird im Layout-Simulator zu Beginn immer die Grundplatte platziert. Diese Grundplatte dient dem Großteil des Layout-Simulators als Basiskoordinatensystem. Der Mittelpunkt der Platte ist der Nullpunkt der Anwendung. Um diese Platte zu platzieren stehen drei Möglichkeiten zur Verfügung.

### **Manuelles Platzieren der Grundplatte**

Beim manuellen Platzieren wird eine 1 x 1 Meter große Platte zwei Meter vor der HoloLens platziert. Diese Platte kann mit Handgesten verschoben, skaliert und rotiert werden. Um zwischen dem Verschieben, Skalieren und Rotieren zu wechseln, existieren Sprachbefehle. Mit diesen Manipulationen kann die Grundplatte auf einen Tisch oder den Fußboden oder auch frei in der Luft platziert werden. Mit einem weiteren Sprachbefehl wird die Positionierung der Platte abgeschlossen.

### **Automatisches Platzieren der Grundplatte**

Beim automatischen Platzieren kommt das Umgebungsmodell der HoloLens zum Einsatz. Dabei wird in einem ersten Schritt das Umgebungsmodell als Dreiecksnetz angezeigt. Der Benutzer scannt den Raum, indem er ihn durchläuft. Wenn der Raum ausreichend gescannt ist führt der Benutzer eine Air-Tap-Geste durch, das Umgebungsmodell wird ausgeblendet und die Platzierung geht in den zweiten Schritt. In diesem wird das Spatial-Understanding-Modul des Holo-Toolkits genutzt, um große Flächen aus dem Umgebungsmodell zu extrahieren. Diese Flächen werden anschließend angezeigt. Der Benutzer kann eine dieser Flächen als neue Grundfläche übernehmen, indem er diese mit dem Cursor markiert und einen Air-Tap ausführt.

### **Verbinden über *Holographic Sharing***

Verbindet sich der Layout-Simulator zu einer zweiten HoloLens Instanz, ist in dieser die Grundplatte schon platziert. Das heißt, die Platzierung wird von der anderen Instanz des Layout-Simulators übernommen. Durch den im Abschnitt über Holographic Sharing besprochenen Mechanismus wird die Position der Grundplatte von der ersten Instanz zur zweiten übertragen. Neben der Position spielt auch die Größe und Skalierung der Grundplatte eine Rolle. Diese wird ebenfalls von der ersten zur zweiten Instanz übertragen um beide Grundplatten abzustimmen.

### 4.3.4 Skalieren eines Layouts

Bei der Verwendung des Layout-Simulators können zwei Einsatzszenarien unterschieden werden. Der erste Anwendungsfall dient zum Analysieren und Optimieren von Layouts. Dies findet in dem meisten Fällen in Modellgröße statt. Die komplette Anlage ist so skaliert, dass sie auf einem Tisch oder ähnlichem Platz findet. So kann die Anlage leicht überblickt und kontrolliert werden. Die Modellgröße entspricht einem Skalierungsfaktor von 0,1.

Ein zweites Szenario ist der Einsatz in einer Werkshalle. Hier wird die Anlage in Originalgröße dargestellt, sodass die Größenverhältnisse eingeschätzt werden können und sich die Benutzer ein Bild von der geplanten Anlage machen können. Die Originalgröße entspricht einem Skalierungsfaktor von eins. Um zwischen diesen beiden Einsatzszenarien wechseln zu können, existiert eine Möglichkeit, die Grundplatte zu skalieren. Wird diese skaliert, passt sich die Größe des Layouts automatisch an. Die Skalierung wird über Sprachbefehle durchgeführt. Mit den Sprachbefehlen „Scale Up“ und „Scale Down“ wird die Skalierung verändert. Mit den Befehlen „Make Full Size“ und „Make Model Size“ kann direkt zwischen den Extremen der Skalierung gewechselt werden. Um vor allem bei großen Skalierungsänderungen die Immersion nicht zu beeinträchtigen, wird zwischen den Zuständen interpoliert.

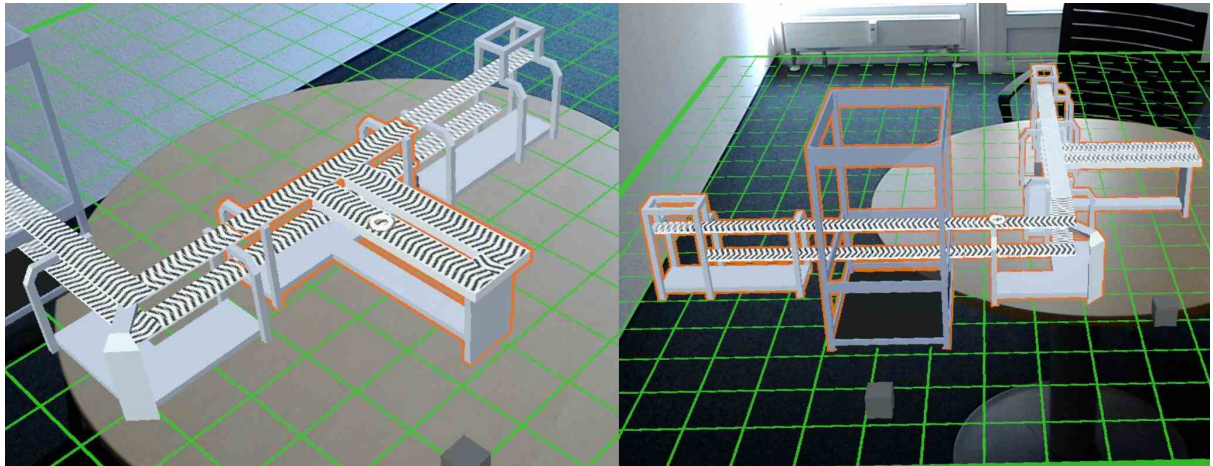
### 4.3.5 Editieren eines Layouts

Eine der beiden Hauptaufgaben des Layout-Simulators ist das Optimieren von Layouts. Dazu muss das Layout verändert werden können. Neben dem Erstellen und Löschen von Elementen müssen auch Elemente abgeändert werden können. Zum Ändern gehört das Verschieben von Elementen, das Rotieren von Elementen und die Änderung der Richtung des Transportbandes. Im Folgenden wird beschrieben, wie diese Funktionen mit Hilfe der HoloLens-Interaktionsmöglichkeiten realisiert werden können.

#### Markieren von Elementen

Wird ein Befehl zum Ändern eines Elements abgegeben, muss die HoloLens wissen, welches Element geändert werden soll. Dazu wird ein Cursor verwendet. Liegt der Cursor auf einem Element, ist dieses markiert und damit das Ziel aller Befehle. Visualisiert wird dieser Zustand durch einen orangenen Rahmen um das Element. Ein Beispiel für ein markiertes Element ist in Abbildung 4.6 links zu sehen.

Zum Markieren von mehreren Elementen wird ein zweiter Modus eingeführt, der alle verbundenen Elemente auf einmal markiert statt nur ein einzelnes Element. Zum Wechseln zwischen den beiden Modi zum Markieren von einzelnen Elementen oder Gruppen von Elementen existieren zwei Sprachbefehle, *Single Mode* und *Grouping Mode*. Ein Beispiel für eine Gruppenmarkierung ist in Abbildung 4.6 rechts dargestellt.



**Abbildung 4.6:** Die orangene Umrandung des Elements zeigt das aktuell markierte Element. Die Markierung erfolgt mit Hilfe des Cursors. Links: Markierung von einzelnen Elementen. Rechts: Markierung von allen verbundenen Elementen.

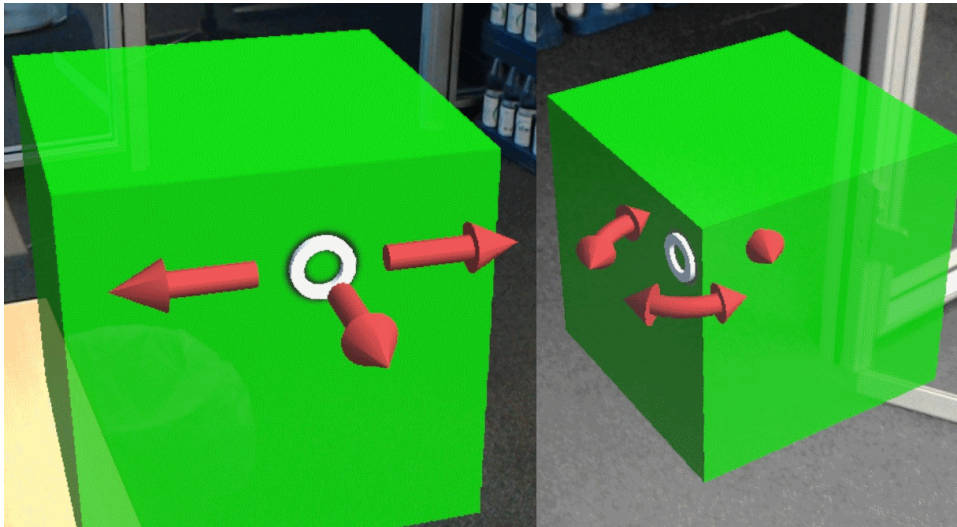
### Erstellen und Löschen von Elementen

Zur Erstellung eines neuen Elements kommen alle Interaktionsarten der HoloLens zum Einsatz. Durch den Sprachbefehl *Create Element* wird ein neues Dummy-Objekt erstellt. Die Position dieses Dummy-Objekts wird durch die Blickrichtung während des Sprachbefehls festgelegt. Das Dummy-Objekt wird an der Position des Cursors erstellt. Befindet sich der Cursor aktuell nicht auf der Grundfläche, wird ein Fehlerton ausgegeben und die Erstellung abgebrochen. Ist das Dummy-Objekt erstellt, wird über diesem eine Liste dargestellt. Diese Liste beinhaltet alle Elemente, die erstellt werden können. Wird eines dieser Elemente in der Liste mit der Blickrichtung, also dem Cursor, markiert und über einen Air-Tap bestätigt, wird das vorherige Dummy-Objekt zu dem ausgewählten Element und die Erstellung ist abgeschlossen.

Um ein Element wieder zu löschen, wird dieses markiert und mit Hilfe des Sprachbefehls *Delete Object* gelöscht. Ist die Gruppenmarkierung aktiv, können auch mehrere Elemente auf einmal gelöscht werden.

### Verschieben und Rotieren von Elementen

Das Verschieben und das Rotieren sind die beiden am meisten genutzten Funktionen beim Verändern eines Layouts. Aus diesem Grund sollen diese unkompliziert durchzuführen sein. Wie auch beim Markieren werden für diese Funktionen zwei Zustände verwendet, der Verschieben-Zustand und der Rotieren-Zustand. Gewechselt wird zwischen diesen Zuständen mit den Sprachbefehlen *Move Object* und *Rotate Object*. Beim Markieren eines Elements ändert sich der Cursor abhängig vom aktuellen Zustand. Können aktuell Elemente verschoben werden, erhält der Cursor Rote Pfeile in alle vier horizontalen Richtungen. Bei möglicher Rotation

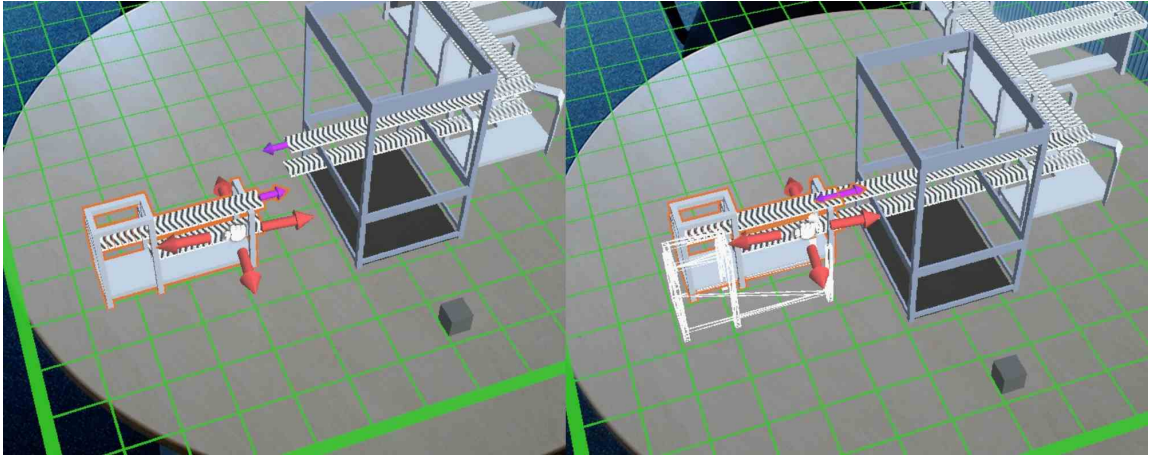


**Abbildung 4.7:** Rote Pfeile um den Cursor zeigen die mögliche Manipulation des Elements an. Links: Verschieben von Elementen, rechts: Rotieren von Elementen.

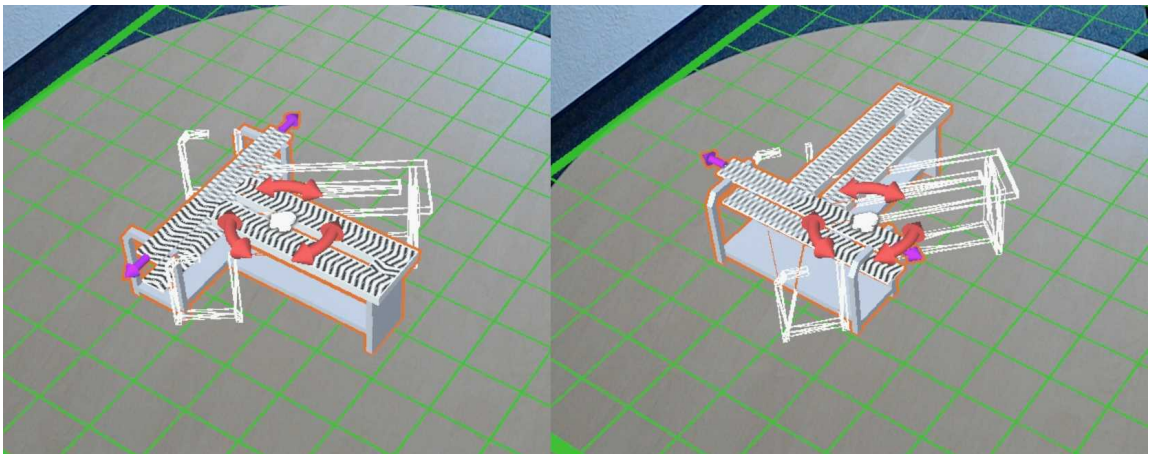
werden Pfeile eingeblendet, die eine Rotation um die vertikale Achse symbolisieren, siehe Abbildung 4.7.

Zum Verschieben eines Elements muss dieses markiert werden und die Manipulationsgeste durchgeführt werden. Da im Layout-Simulator nur Verschiebungen in der horizontalen Ebene sinnvoll sind, wird die vertikale Komponente des Offsets auf null gesetzt. Ist der Abstand zwischen zwei offenen Verbindungen kleiner als ein definierter Schwellwert, werden die beiden Verbindungen zusammengeschlossen. Dabei dockt das sich bewegende Element an das andere an. An der ursprünglichen Position des Elements wird ein Drahtgitter dargestellt, siehe Abbildung 4.8. Dieser Vorgang wird *Snapping* genannt. Wird die Geste während des *Snappings* beendet, bleibt das Element an der angedockten Position und die Verbindungen werden zusammengeschlossen. So wird sichergestellt, dass Verbindungen, die nahe beieinander liegen auch wirklich verbunden sind. Der Schwellwert des *Snapping* kann individuell angepasst werden. Basierend auf Tests mit verschiedenen Größen sollte der Wert auf ungefähr ein Viertel des Elementdurchmessers gesetzt werden.

Das Rotieren eines Elements funktioniert ähnlich wie das Verschieben. Das Element muss markiert sein und die Manipulationsgeste durchgeführt werden. Es werden nur Rotationen um die vertikale Achse zugelassen, aus diesem Grund wird nur der Manipulationsoffset nach links oder rechts beachtet. Im Layout-Simulator sind nur Winkel von vollen 90 Grad erlaubt, also 0 Grad, 90 Grad, 180 Grad oder 270 Grad. Befindet sich der Rotationswinkel zwischen diesen Werten, wird ein Drahtgittermodell eingeblendet, um den Zustand der Rotationsgeste zu visualisieren. Dieses Drahtgittermodell wird um den Rotationswinkel rotiert. Die Rotation des eigentlichen Elements wird auf das nächste Vielfache von 90 Grad gerundet und angezeigt. So ist der Rotationszustand des Elements zu jeder Zeit valide.



**Abbildung 4.8:** Beim Verschieben springt das Element zur nächsten validen Position, wenn der Abstand zwischen den beiden Verbindungen klein genug ist. An der ursprünglichen Position des Elements wird ein Drahtgittermodell angezeigt.

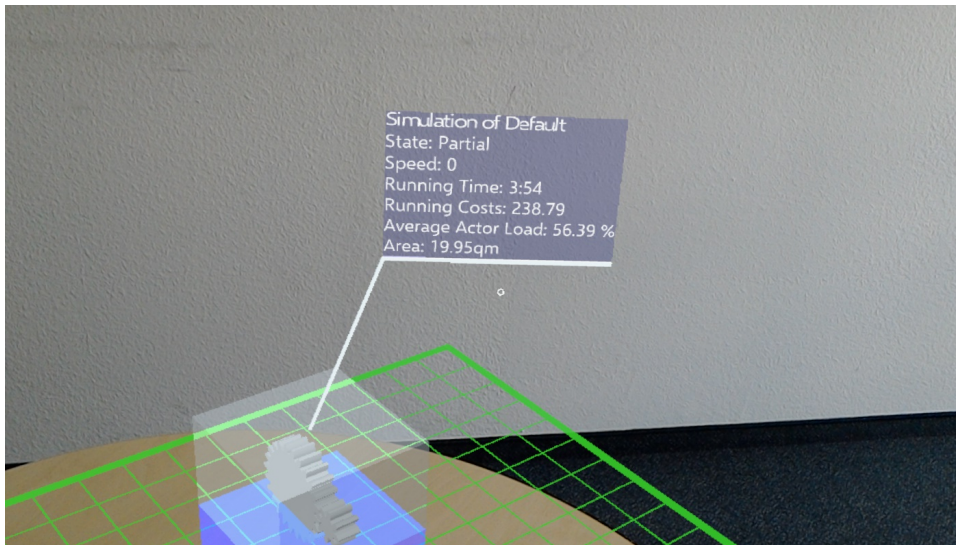


**Abbildung 4.9:** Beim Rotieren nimmt das Element nur Rotationen mit vollen 90 Grad Winkeln an. Das Drahtgittermodell wird kontinuierlich rotiert um den aktuellen Rotationsstatus anzuzeigen.

### 4.3.6 Visualisieren von Simulationsergebnissen

Um zielgerichtete Optimierungen von Layouts zu ermöglichen, werden Simulationsergebnisse während und nach einer Simulation angezeigt. Das Ziel der Visualisierungen ist es, ein Gefühl für die Performance eines Layouts zu erhalten. Die Visualisierungen können in zwei Gruppen aufgeteilt werden. Die erste Gruppe besteht aus Visualisierungen, die einen schnellen Überblick über die Performance und Probleme des Layouts geben sollen. Für diese Gruppe wird versucht, auf Text zu verzichten, da viel Text unübersichtlich ist. Die zweite Gruppe dient der detaillierten Analyse des Layouts mit Hilfe von Detailinformationen. Der Simulationszustand und die



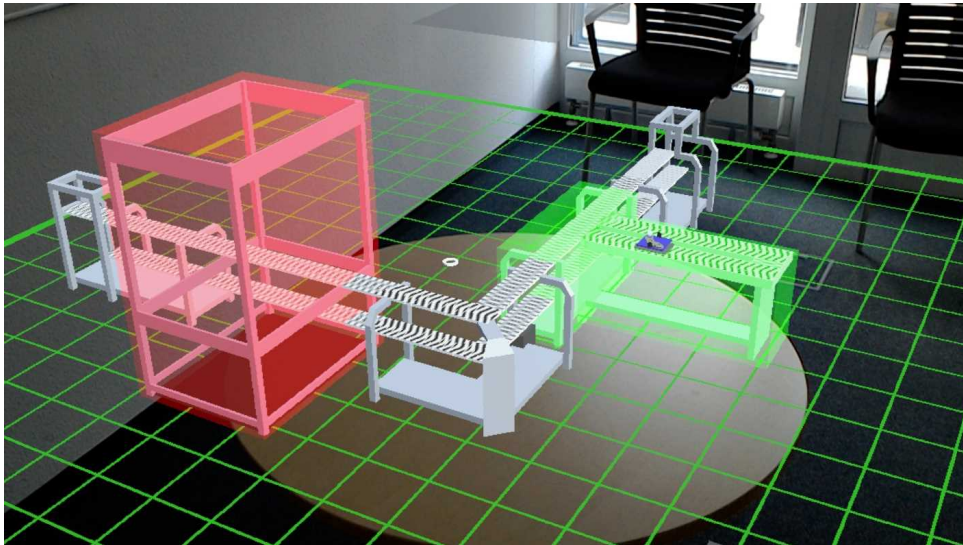


**Abbildung 4.10:** Die Simulationenstatusbox zeigt den aktuellen Zustand der Simulation. Die wichtigsten Informationen wie Status, Fortschritt und Geschwindigkeit werden visualisiert. Im Pop-up werden in Textform Detailinformationen eingeblendet

Simulationsergebnisse werden dabei aus der Simulationsbibliothek des Layout-Simulators von Wörner et al. [WE11; WE13] ausgelesen.

### Visualisierung des Simulationszustandes

Um einen Überblick über die aktuelle Simulation, also Werte wie Laufzeit der Simulation, prozentualer Fortschritt, Geschwindigkeit der Simulation, Status der Simulation (Simulation Läuft, Gestoppt, Fehler, Pausiert, Abgeschlossen) und die Metriken zur Performanceberechnung zu bekommen, wird eine Simulationsstatusbox verwendet. Diese Box besteht aus Zahnrädern in einem transparenten Würfel, siehe Abbildung 4.10. Stehen die Zahnräder still, ist die Simulation entweder gestoppt oder pausiert. Bewegen sich die Zahnräder, läuft die Simulation. Je schneller die Zahnräder sich bewegen, desto höher ist die Geschwindigkeit der Simulation. Der Würfel besteht aus zwei Teilen. Der untere Teil ist blau und der obere Teil ist grau. Die Größe des blauen Anteils ist verbunden mit dem prozentualen Fortschritt der Simulation. In Abbildung 4.10 kann so ein Simulationsfortschritt von circa 40% abgelesen werden. Für detaillierte Informationen werden bei längerem Betrachten der Box Detailinformationen in Textform angezeigt, siehe Abbildung 4.10. Die angezeigten Informationen sind: Name des Layouts, Status, Geschwindigkeit, Laufzeit, Kosten, durchschnittliche Auslastung der Arbeitsstationen, die benötigte Fläche und ein Score (deutsch: eine Bewertung) des Layouts.



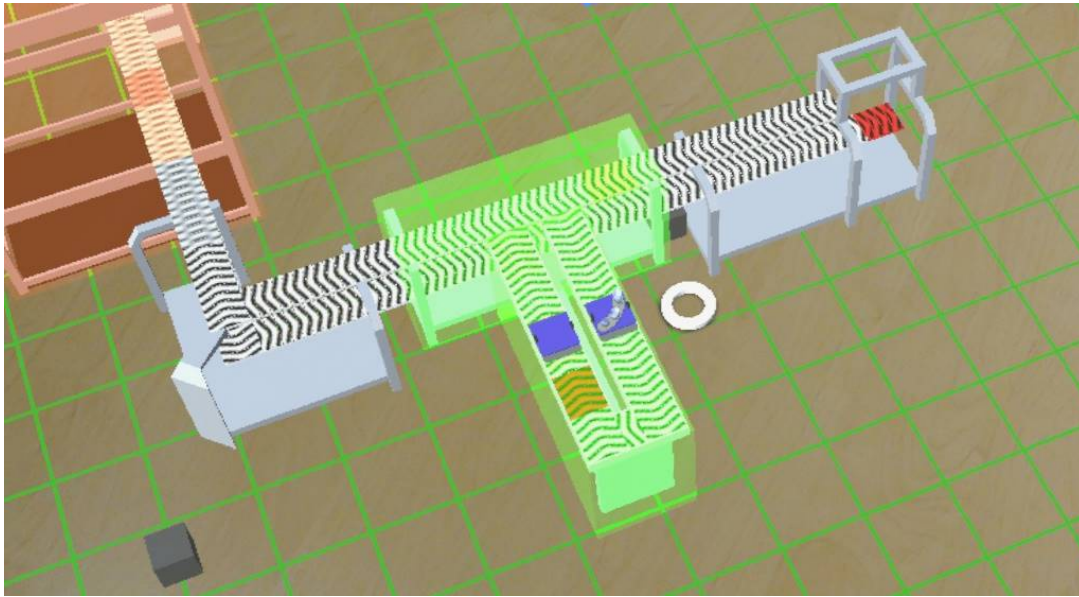
**Abbildung 4.11:** Visualisierung der Auslastung einer Arbeitsstation. Rot entspricht geringer Auslastung, Grün entspricht hoher Auslastung.

### Visualisierung der Auslastung

Die Auslastung von Arbeitsstationen ist eine wichtige Kenngröße bei der Analyse von Layouts. Je größer die Auslastung einer Arbeitsstation ist, desto effizienter werden die zur Verfügung stehenden Ressourcen genutzt. Aus diesem Grund wird die Auslastung der einzelnen Arbeitsstationen dauerhaft sichtbar visualisiert. Dazu wird um die Arbeitsstation ein transparenter Quader gelegt. Dieser erlaubt weiterhin die Sicht auf die Station und Werkstücke, die diese Station bearbeitet. Durch die Farbe des transparenten Quaders können gleichzeitig Informationen zur Auslastung dargestellt werden. Die Farbe wird dabei anhand einer Farbpalette von Rot über Gelb nach Grün berechnet. Rot steht dabei für eine Auslastung von 0%, Gelb für eine Auslastung von 50% und Grün für eine Auslastung von 100%. Diese Zuordnung wurde gewählt, da eine geringe Auslastung durch die rote Farbe stärker ins Auge fallen soll. Ein Beispiel für die Auslastungsvisualisierung ist in Abbildung 4.11 dargestellt. Die Roboterstation auf der linken Seite hat eine geringe Auslastung und ist deswegen rot, während die Handarbeitsstation auf der rechten Seite voll ausgelastet ist und grün dargestellt wird. Durch Optimierungen sollte nun versucht werden, die Roboterstation ebenfalls voll auszulasten.

### Visualisierung der Wartezeiten

Eine zweite Metrik zur Erkennung von Engpässen oder Problemen in einem Layout ist die Visualisierung der Wartezeit der Werkstücke. Diese Metrik soll direkt auf dem Förderband dargestellt werden. Dazu werden die Positionen, auf denen die Werkstücke im Zustand *Warten* sind, aufgezeichnet. Jeder Förderbandabschnitt hat eine kumulierte Werkstückwartezeit die sich mit der Zeit erhöht. Abhängig von dieser Wartezeit wird der Abschnitt eingefärbt. An



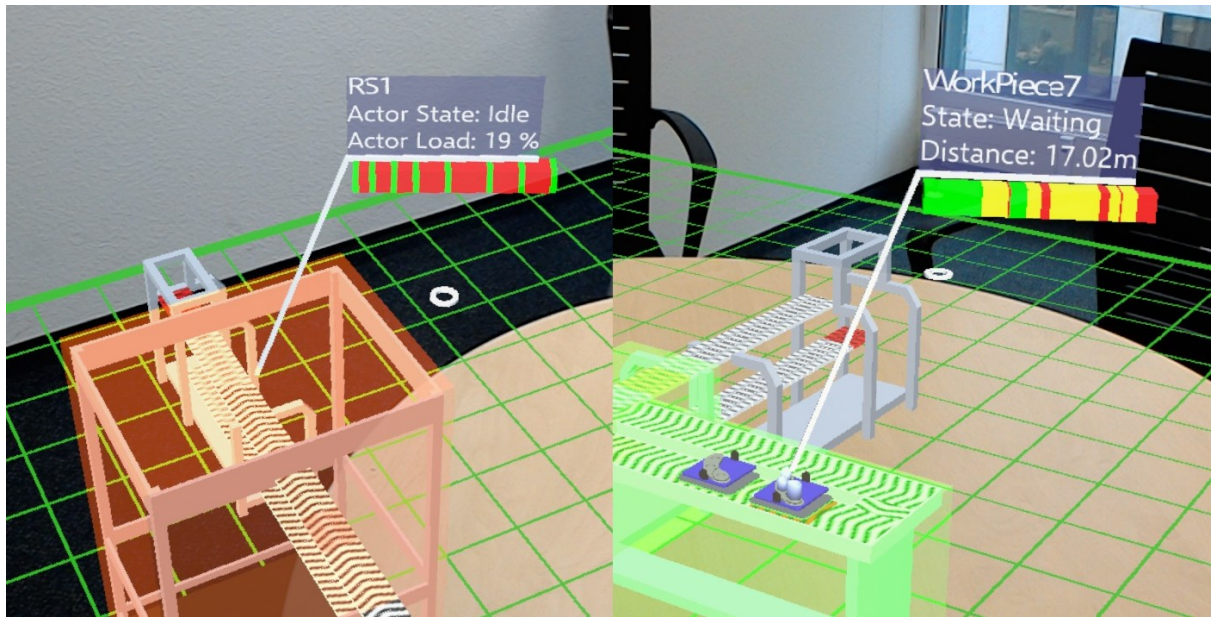
**Abbildung 4.12:** Visualisierung der kumulierten Wartezeit der Werkstücke. Je rötlicher der Förderbandabschnitt ist, desto länger mussten Werkstücke dort warten. Im Beispiel müssen die Werkstücke im Lift oder vor der Handarbeitsstation warten.

Stellen, an denen die Wartezeit null ist, wird das Förderband in seiner ursprünglichen Farbe angezeigt. Je höher die Wartezeit ist, desto rötlicher wird der Förderbandabschnitt dargestellt. Da die Zeiten ständig steigen, ergibt eine statische Farbzuordnung keinen Sinn. Stattdessen wird die Farbzuordnung dynamisch an die maximale Wartezeit angepasst. In Abbildung 4.12 ist zu sehen, dass vor und in der Handarbeitsstation Förderbandabschnitte rot gefärbt sind. An diesen Abschnitten müssen Werkstücke im dargestellten Layout vermehrt warten.

## Popups

Um auf Abruf detaillierte Informationen zu Werkstücken oder Arbeitsstationen zu erhalten, werden Popups verwendet. Wird ein Objekt länger als eine Sekunde mit dem Cursor markiert, erscheint ein Popup, siehe Abbildung 4.13. Dieser Popup bleibt geöffnet, bis ein anderes Objekt markiert wird und dort ein Popup geöffnet wird. Die angezeigten Informationen beinhalten grundsätzlich den Namen des Objekts. Je nach Typ werden weitere Informationen angezeigt.

Die Popups werden beim Öffnen so ausgerichtet, dass die Normale der Textebene zum Benutzer zeigt. So können die Informationen optimal abgelesen werden. Damit die Popups immer perfekt zu lesen sind, sollten diese sich dauerhaft anhand der relativen Position zwischen Popup und Benutzer ausrichten. In dieser Arbeit sind die Popups jedoch bewusst statisch. Bei Tests hat sich herausgestellt, dass die dauerhafte Neuausrichtung der Popups unnatürlich wirkt.



**Abbildung 4.13:** Zur Anzeige von Detailinformationen werden Popups verwendet. Die Informationen sind je nach Objekt unterschiedlich. Rechts: Informationen zu einem Werkstück inklusive Zustandshistorie. Links: Informationen zu einer Arbeitsstation inklusive Auslastungshistorie.

Ist das Objekt ein Werkstück, werden neben dem Namen auch der aktuelle Zustand (*Warten, Bewegung, Bearbeitung*) und die zurückgelegte Strecke angezeigt. Unterhalb des Textes wird die Zustandshistorie des Werkstücks visualisiert. Dazu wird jeder Zustand einer Farbe zugeordnet, Rot entspricht *Warten*, Gelb entspricht *Bewegung*, Grün entspricht *Bearbeitung*. Wie in Abbildung 4.13 links zu sehen ist, wird ein Balken eingeblendet. Die längste Seite entspricht der Zeitachse und die Farbe dem Zustand. So kann verfolgt werden, in welchem Zustand sich das Werkstück im Verlauf der Zeit befunden hat.

Ist das Objekt eine Arbeitsstation, werden neben dem Namen auch die durchschnittliche Auslastung und der Zustand (*Warten, Arbeiten*) angezeigt. Ähnlich wie bei Werkstücken wird unterhalb des Textes erneut die Zustandshistorie in Form eines länglichen Balkens visualisiert. Grün entspricht *Arbeiten* und Rot entspricht *Warten*, siehe Abbildung 4.13 rechts.

### 4.3.7 Holographic Sharing in Verbindung mit dem Layout-Simulator

Bei der Planung und Analyse von Fabriklayouts oder beim Einpassen eines Fabriklayouts in eine Produktionsstraße ist selten nur eine Person beteiligt. Aus diesem Grund soll in den Layout-Simulator die Holographic-Sharing-Funktionalität eingebaut werden. So können mehrere Personen den Layout-Simulator gleichzeitig nutzen.

Zur Kommunikation zwischen den Teilnehmern wird die Desktop-Anwendung *Sharing-Server* des Holo-Toolkits verwendet. Diese Anwendung läuft auf einem Desktop-Computer im gleichen Netzwerk. Die einzelnen Instanzen verbinden sich mit Hilfe der IP-Adresse des Desktop-PCs mit diesen Server. Alle Teilnehmer die Daten austauschen möchten treten der gleicher Sitzung bei. Die Layout-Simulator-Instanz die die erste Instanz in einer Sitzung ist arbeitet als Master, jede weitere Instanz arbeitet als Slave. Die Grundplatte wird im Master einmalig platziert, die Slaves übernehmen die Positionierung, siehe Abschnitt 4.3.3. So haben alle Instanzen ein gemeinsames Koordinatensystem und können Positionsdaten austauschen. Ein erster Datensatz, der ausgetauscht wird, ist die Größe und Skalierung der Grundplatte. Mit Hilfe von diesen Informationen sehen alle Instanzen dieselbe Grundplatte.

Ein weiterer Datensatz, der ausgetauscht wird ist die aktuelle Kopfposition und die aktuelle Cursorposition jedes Teilnehmers. Durch das Einblenden der Cursor der anderen Instanzen wird die Kommunikation erleichtert. Durch das Wissen über die Kopfposition kann eine Kopfbedeckung oder ähnliches dargestellt werden. Diese Kopfbedeckung ist farblich identisch zum Cursor des Benutzers. So können bei vielen Teilnehmern die einzelnen Farben den Personen zugeordnet werden.

Neben diesen allgemeinen Informationen müssen auch anwendungsspezifische Daten ausgetauscht werden. Dazu gehört vor allem der aktuelle Aufbau des Layouts und die Simulationsdaten. Für eine Liste aller relevanten Simulationsdaten wird auf Abschnitt 4.3.6 verwiesen. Um diese Daten synchron zu halten, existieren zwei Ansätze:

- Synchronisieren der Datenstruktur der Layout-Simulator-Bibliothek
- Synchronisieren der Unity-Objektstruktur

In den folgenden beiden Abschnitten werden diese Ansätze evaluiert und verglichen.

### **Synchronisieren der Layout-Simulator Bibliothek Datenstruktur**

Da die in der Layout-Simulator Bibliothek verwendete Datenstruktur für Layouts tief verschachtelt ist, lässt sich diese sich nur schlecht serialisieren und über das Netzwerk versenden. Jede Änderung in der Datenstruktur muss allen Teilnehmern mitgeteilt werden. Da jedes Element Parent- und Child-Referenzen beinhaltet, kann ohne zusätzliche Arbeit nur die komplette Datenstruktur serialisiert werden. Sollen nur Teile der Datenstruktur serialisiert werden, müssen Referenzen manuell aufgelöst werden. Dies ist vor allem bei kontinuierlichen Änderungen wichtig, da sonst zu große Datenmengen über das Netzwerk gesendet werden.

Vorteil dieses Ansatzes ist es, dass es für die übergeordnete Unity-Anwendung keinen Unterschied macht, ob der Layout-Simulator als Master oder Slave arbeitet. Die Anpassungen des Unity-Anwendungscode sind minimal.

### **Synchronisieren der Unity-Objektstruktur**

Zur Synchronisierung von Unity-Objekten bietet das Holo-Toolkit einfach zu verwendende Hilfsklassen an. Diese sorgen für die Synchronisation der Datenstrukturen. Der Nachteil dieses Ansatzes liegt in dem doppelten Programmieraufwand. Der Master muss die Daten aus dem Simulator auslesen und an den Sharing-Server weiterleiten während die Slaves die Daten aus den *Sharing*-Bibliotheken auslesen müssen. Dadurch muss ein großer Teil der Programmierung angepasst werden. Die Unterscheidung zwischen Master und Slave ist größer als beim ersten Ansatz. Möchte ein Slave auf Simulationsergebnisse zugreifen muss diese Anfrage über den Master gesendet werden.

Der Vorteil dieses Ansatzes liegt in der bereits vorhandenen Kommunikation zwischen den Teilnehmern. Die Synchronität der Daten muss nicht selbst sichergestellt werden, sondern ist bereits implementiert.

### **Einschränkungen der Slaves**

Um Komplikationen und Inkonsistenzen zu vermeiden, können die meisten Operationen nur vom Master ausgeführt werden. Dazu gehört das Wechseln des Zustands, das Laden eines neuen Layouts, das Skalieren der Grundplatte und das Starten und Stoppen von Simulationen. Die Interaktionsmöglichkeiten der Slaves beschränken sich auf das Modifizieren von Layouts, das heißt, alle Benutzer können gleichzeitig das geladene Layout manipulieren. Um Inkonsistenzen zu vermeiden, muss verhindert werden, dass zwei Benutzer gleichzeitig das selbe Element verschieben können. Dies wird durch eine Blockierung, ähnlich eines Semaphors realisiert.

## **4.4 Implementierung**

Die Anforderungen und Konzepte der letzten Abschnitte wurden in Form eines Prototyps implementiert. In diesem Kapitel wird dieser Prototyp beschrieben. Dabei wird auf Werkzeuge, Bibliotheken und die Architektur eingegangen. Im letzten Abschnitt wird auf die besonderen Probleme bei der Entwicklung für die HoloLens eingegangen. Zur Softwareentwicklung des Prototyps wurde hauptsächlich ein 15 Zoll Laptop verwendet. Der Laptop besitzt einen Intel Core I7-3740QM Prozessor mit 2,6 Gigahertz und 4 Kernen, eine NVIDIA Quadro K2000M Grafikkarte und 16 Gigabyte Arbeitsspeicher.

### 4.4.1 Verwendete Werkzeuge und Bibliotheken

Zur Entwicklung der Software werden folgende Werkzeuge verwendet:

- **Unity 5.5** ist eine Entwicklungsumgebung für 2D und 3D-Spiele. Unity zeichnet sich vor allem durch seine Vielzahl an unterstützten Geräten aus, dazu gehört auch die Microsoft HoloLens.
- **Microsoft Visual Studio 2015 Enterprise** ist eine Entwicklungsumgebung für verschiedene Programmiersprachen. In diesem Projekt dient sie zur Entwicklung von C#-Programmcode für Unity und zum Kompilieren und Bereitstellen der Anwendungen für die HoloLens.

Neben diesen Werkzeugen wurden folgende Bibliotheken verwendet:

- **HoloToolkit-Unity**, diese Bibliothek für Unity und HoloLens unterstützt Entwickler beim Einstieg in die Anwendungs-Entwicklung für Augmented Reality. Die Bibliothek bietet Komponenten für Gestenerkennung, Raumerkennung, dreidimensionalen Sound, Buildvorgänge und kollaborative Anwendungen für mehrere HoloLens-Brillen. Der Quelltext steht zur freien Verfügung und kann beliebig erweitert werden [Mic17b].
- **Vuforia** ist eine Augmented-Reality-Plattform. In dieser Arbeit wird das Modul zur Markererkennung verwendet [PTC17a].
- **ARToolkit** ist eine Bibliothek zum Erkennen und Tracken von Markern [ART16].

### 4.4.2 Verwendung von Unity-Szenen

Unity verwendet zum Aufbau einer Anwendung Szenen. Eine solche Szene besteht aus verschiedenen Objekten wie zum Beispiel Lichtern, Kameras, und 3D-Objekten. In klassischen Spielen bildet jede dieser Szenen ein Level ab. Im Layout-Simulator wird stattdessen jeder Zustand des Zustandsautomaten als Szene realisiert. Da die HoloLens-Anwendung Objekte und Funktionen wie zum Beispiel die Kamera, der Cursor, Spracherkennung, Handgestenerkennung und das Umgebungsmesh in jeder Szene benötigt, wird eine Basisszene verwendet. Diese Basisszene wird von der Anwendung dauerhaft geladen und abhängig vom aktuellen Zustand wird die darüberliegende Szene ständig ausgetauscht. So können die einzelnen Funktionen möglichst unabhängig voneinander entwickelt werden und dabei immer auf die gleiche Basis zugreifen.

### 4.4.3 Probleme bei der Entwicklung für die HoloLens

Bei der Entwicklung der Anwendung für die HoloLens traten Probleme auf, die in klassischen Desktop-Anwendungen nicht auftreten. Diese Probleme und die gefundenen Lösungen sind im folgenden Kapitel zusammengetragen.

```
#if WINDOWS_UWP
    thread = new System.Threading.Tasks.Task(Run);
#else
    thread = new Thread(Run);
    thread.IsBackground = true;
    thread.Priority = ThreadPriority.BelowNormal;
    thread.Name = "Simulation thread";
#endif
thread.Start();
```

**Listing 4.1:** C# Code für unterschiedliche .Net Versionen. Die richtige Version wird automatisch vom Compiler aktiviert

### Portierung der Fabriklayout-Simulator-Bibliothek auf die HoloLens

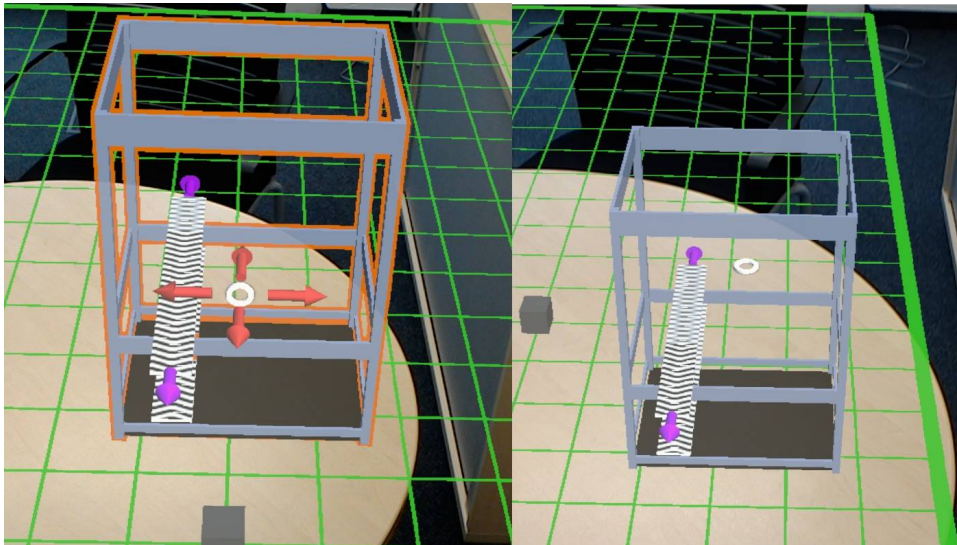
Der Fabriklayout-Simulator basiert auf einer Fabriklayout-Simulator-Bibliothek. Diese Bibliothek beinhaltet die Logik zum Laden verschiedener Layouts aus einer Datei und zum Simulieren dieser Layouts. Sie wird auch im Fabriklayout-Simulator für die HoloLens verwendet. Die Bibliothek ist in C# geschrieben und verwendet das .Net 4.0 Framework. Unity arbeitet mit Version 3.5 des .Net Frameworks. Um die Bibliothek in Unity zu verwenden, muss diese zu .Net 3.5 portiert werden.

Die HoloLens arbeitet nur mit C#-UWP-Bibliotheken. Darum muss die Bibliothek zusätzlich in das .Net UWP Framework portiert werden. Um die Bibliothek in Unity und auf der HoloLens verwenden zu können, werden zwei verschiedene Versionen der Bibliothek benötigt. Die erste Version in .Net 3.5 für Unity, die zweite Version auf UWP-Basis für die HoloLens.

Durch die Änderungen in den .Net Bibliotheken muss an einigen Stellen der Fabriklayout-Simulator-Bibliothek unterschiedlicher Programmcode verwendet werden. Der Code sollte allerdings nicht auf zwei Versionen aufgeteilt werden, da sonst jede Änderung in beiden Versionen parallel geändert werden muss. Um diesen doppelten Aufwand zu eliminieren werden die entsprechenden inkompatiblen Codeabschnitte mit Compilerdirektiven versehen, siehe Listing 4.1.

So kann die gleiche Codebasis für beide Versionen der Bibliothek verwendet werden, da der Compiler je nach verwendeter .Net Version die korrekten Codeabschnitte aktiviert. Alle Änderungen müssen nur einmal und nicht doppelt eingefügt werden. In der portierten Bibliothek mussten 90 Codeabschnitte angepasst werden. Von diesen 90 Inkompatibilitäten waren nur 8 verschieden. Das heißt der gleiche Lösungsansatz konnte für viele Codeabschnitten verwendet werden.





**Abbildung 4.14:** Unterschiedliche Kollisionsstrukturen der Roboterstation. Rechts: Das Dreiecksmesh wird verwendet. Dadurch kann der Cursor durch die Anlage hindurch dahinterliegende Objekte fokussieren. Rechts: Ein Quader, der das Objekt umgibt, wird als Kollisionsstruktur verwendet. Die gröbere Struktur vereinfacht die Fokussierung und damit die Bedienbarkeit der Anwendung.

### Fokussierung von Fabriklayoutelementen

Die Blickrichtung ist in der HoloLens ein wichtiges Mittel der Interaktion. Allerdings ist die exakte Fokussierung auf kleine oder weit entfernte Objekte schwierig, da der Kopf nie ganz ruhig gehalten werden kann. Diese Probleme treten aber nicht nur bei kleinen Objekten auf, sondern auch bei Objekten mit dünnen Strukturen und Löchern. Ein Beispiel für eine solche Struktur ist die Roboterstation des Fabriklayout-Simulators. Beim Editieren von Layouts kommt es vor, dass beim Markieren der Roboterstation der Cursor nicht auf einer der Stützen liegt, sondern Objekte hinter der Station fokussiert, siehe Abbildung 4.14 rechts. Die Folge ist, dass Manipulationen der Roboterstation scheinbar nicht funktionieren, weil der Cursor immer wieder den Hintergrund fokussiert.

Aus diesem Grund wird für die Kollisionsstruktur der einzelnen Elemente des Layoutsimulators nicht das Dreiecksmesh, sondern vereinfachte Ersatzgeometrien verwendet. Im Fall der Roboterstation wird das Mesh durch einen Quader, der das Objekt umschließt ersetzt, siehe Abbildung 4.14 links. So ist eine einfachere und zuverlässigere Fokussierung der Objekte möglich. Ein Nachteil dieses Ansatzes ist es, dass Werkstücke, die sich in der Roboterstation befinden nicht mehr fokussiert werden können. Um dieses Problem zu umgehen könnte ein Mechanismus verwendet werden, der zuerst prüft ob ein Objekt in der Roboterstation fokussiert wird. Nur wenn dies nicht der Fall ist, wird die Roboterstation fokussiert.

### Highlighting von Elementen

Um Objekte zu Markieren, soll im Fabriklayout-Simulator ein Highlighting-Effekt eingesetzt werden. Allerdings unterstützt die HoloLens nur Forward Rendering und Screen-Space-Effekte sollten aus Performance Gründen nicht verwendet werden [Mic17c]. Aus diesem Grund wird ein eigener Highlight-Shader verwendet. Dieser Highlight-Shader nutzt intern einen Geometry Shader um das Objekt entlang der Normalen aufzublähen. Der Renderablauf eines Objekts mit Highlighting besteht aus zwei Schritten:

1. Das Objekt wird mit dem Highlight-Shader gerendert. Dabei wird das Objekt aufgebläht und komplett einfarbig dargestellt. Während dieses Schrittes wird der Z-Buffer nicht geschrieben.
2. Das Objekt wird mit dem ursprünglichen Shader in seiner normalen Größe gerendert. Da im vorherigen Schritt der Z-Buffer nicht geschrieben wurde überdeckt dieser zweite Rendervorgang den Highlight-Shader überall außer an den Rändern.

Das Ergebnis dieses Ablaufs ist in Abbildung 4.6 im Abschnitt „Konzept“ zu sehen.

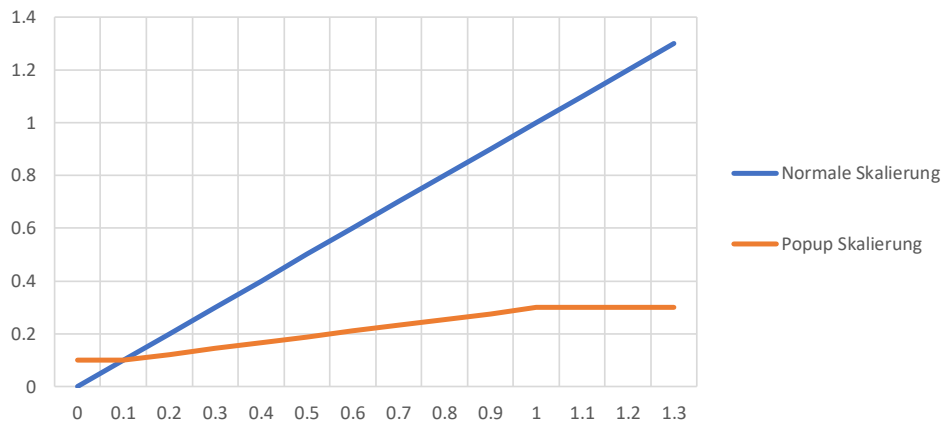
### Skalierung der Popups

Ein Problem bei der Verwendung von Popups ist die Skalierung. Ausgegangen wird vom Layout in Modellgröße, so dass es auf einen Tisch passt. In dieser Skalierung ist das Popup circa 40 Zentimeter groß und aus ein bis zwei Metern Abstand optimal abzulesen.

Wird das Layout nun auf Originalgröße hochskaliert und die Popups skalieren mit, sind die Popups so groß, dass nicht gleichzeitig das Element und das Popup betrachtet werden können. Besitzen die Popups stattdessen eine feste Größe, also werden nicht von der Skalierung vergrößert oder verkleinert, sind diese viel zu klein, wenn das Layout auf Originalgröße skaliert wird.

Aus diesem Grund wird eine Zwischenlösung verwendet. Die Skalierung des Layouts wird nur in Teilen zur Skalierung des Popups verwendet. Das bedeutet, dass das Popup größer wird wenn das Layout größer wird, jedoch nur in abgeschwächter Form. Das Popup hat eine Mindestgröße und eine maximale Größe und zwischen diesen einen linearen Übergang, siehe Abbildung 4.15. Dadurch wird das Popup zwar größer, allerdings nicht so groß, dass es nicht mehr auf einen Blick zu erfassen ist.

Um dieses Problem komplett zu verhindern könnten auch ein 2D-Overlay in der Fokusebene verwendet werden. Der Text wäre dadurch immer gut zu erkennen. Egal wohin der Benutzer schaut und wie weit er vom Objekt entfernt steht. Allerdings sollten in Augmented-Reality-Anwendungen alle Objekte einen örtlichen Bezug haben. Objekte die fest an die Kamera geheftet sind stören in vielen Fällen, da man nicht um diese herum schauen kann oder von diesen wegläufen kann. Stattdessen befinden Sie sich ständig im Blickfeld, egal ob man diese gerade betrachten will oder nicht.



**Abbildung 4.15:** Skalierung eines Popups im Vergleich zur normalen Skalierung. Die Mindestskalierung liegt bei 0,1 und die maximale Skalierung bei 0,3. Dazwischen liegt ein linearer Übergang.

## 4.5 Evaluation

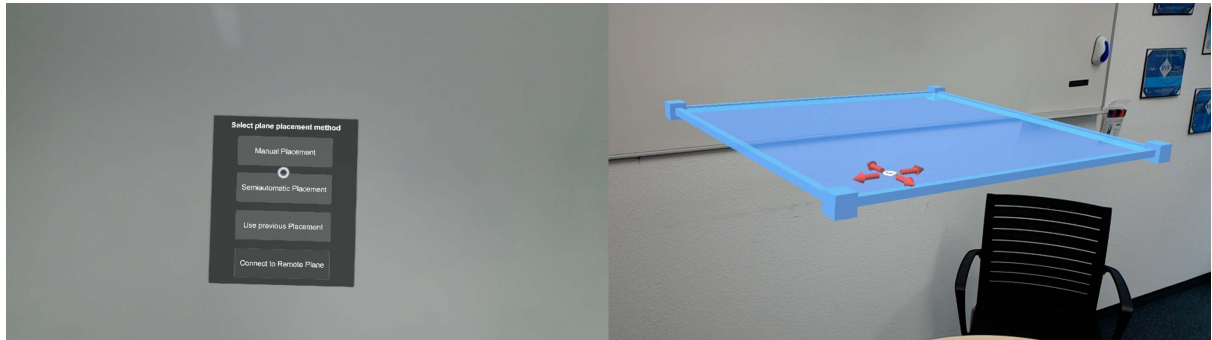
In diesem Abschnitt wird im ersten Teil die Erfüllung der aufgestellten Anforderungen durch den Prototyp überprüft. Im zweiten Teil wird evaluiert, in wie weit die Anwendung in realen Szenarien eingesetzt werden kann. Im letzten Teil wird die Bedienbarkeit und der Informationstransport für verschiedene Benutzer überprüft.

### 4.5.1 Vergleich des Prototyps mit den gestellten Anforderungen

Die meisten Anforderungen, die sich aus der Augmented-Reality-Umgebung ergeben, wurden erfüllt. Nur die Forderung nach einer **konstant hohen Bildwiederholungsrate** und die Forderung **kollaboratives Arbeiten** zu ermöglichen, wurden aus Zeitgründen nicht erfüllt. Eine genaue Messung der Bildwiederholungsrate ergibt keinen Sinn, da dieser Wert stark abhängig vom gezeigten Layout und vom betrachteten Bildabschnitt abhängt. Im Standard Layout welches aus 5 Elementen besteht fällt die Bildwiederholungsrate in manchen Fällen auf 30 Bilder pro Sekunde ab. Je größer das Layout desto öfter werden keine 60 Bilder pro Sekunde mehr erreicht. In Kapitel 6 werden Möglichkeiten der Optimierung aufgezeigt um eine Konstant hohe Bildwiederholungsrate von 60 Bildern pro Sekunde zu erreichen.

Für die Synchronisierung von Layouts zwischen mehreren HoloLens-Brillen existieren zwei Konzepte die jedoch nicht mehr getestet und evaluiert werden konnten.

Die funktionalen Anforderungen, die sich aus der Desktop-PC-Version des Layout-Simulators ableiten, wurden zum großen Teil erfüllt. Die Anforderungen zum **Erstellen von neuen Elementen** und die Anforderung zum **Ändern der Einstellungen von Elementen** konnten



**Abbildung 4.16:** Links: Auswahl wie die Grundplatte platziert werden soll. Rechts: Manuelle Platzierung der Grundplatte.

aus Zeitgründen nicht erfüllt werden. Zu diesen beiden Szenarien wurden Konzepte entwickelt. Die Implementierung dieser Konzepte würde sich in den verwendeten Techniken nur geringfügig von anderen implementierten Funktionen unterscheiden und ist kein essenzieller Bestandteil des Konzepts und wurde aus diesem Grund nicht priorisiert.

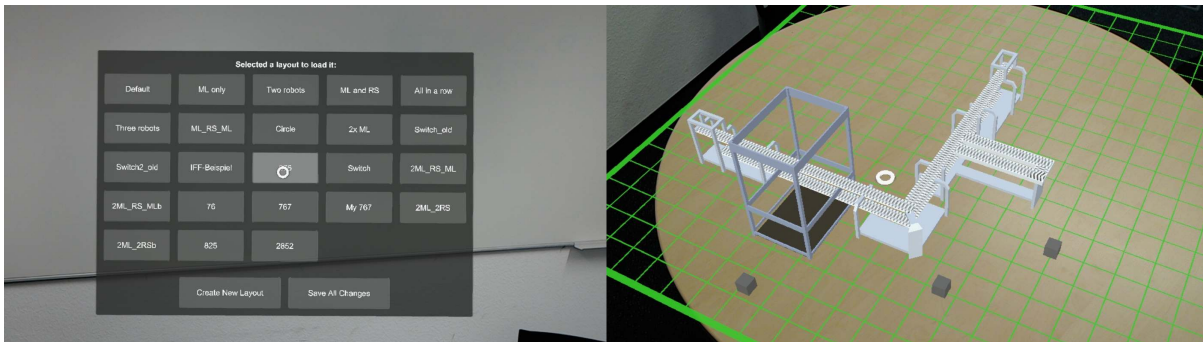
Die nicht funktionalen Anforderungen werden in den folgenden Abschnitten genauer untersucht.

### 4.5.2 Demonstration eines Analyseablaufs durch den Simulator

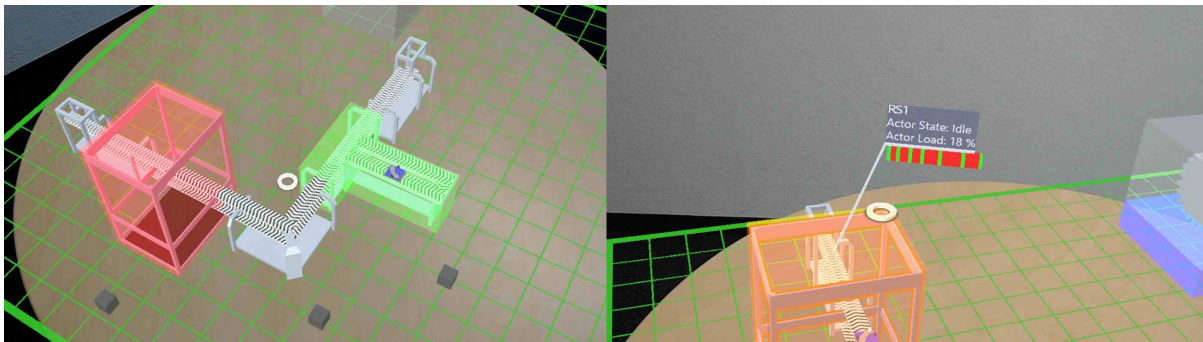
Im diesem Abschnitt wird ein klassischer Arbeitsablauf des Fabriklayout-Simulators in der HoloLens demonstriert. Dabei wird ein Layout zuerst analysiert und anschließend optimiert. Nach dem Starten der Anwendung erscheint ein Interface, in dem sich der Benutzer für eine Methode zur Platzierung der Grundplatte entscheiden muss, siehe Abbildung 4.16 links. In diesem Beispiel wird die Platte manuell platziert. Während der Platzierung der Platte kann diese mit den Sprachbefehlen *Rotate Plane*, *Move Plane*, *Scale Plane* und der Manipulationsgeste korrekt platziert werden, siehe Abbildung 4.16 rechts.

Durch den Sprachbefehl *Finish Placement* wird die aktuelle Platzierung übernommen und dem Benutzer wird eine Liste von verfügbaren Layouts angezeigt, siehe Abbildung 4.17 links. Der Benutzer wählt das Layout *Default* aus, welches geladen und angezeigt wird, siehe Abbildung 4.17 rechts.

Mit dem Sprachbefehl *Simulation Mode* und einem Klick auf die Simulationsbox wird die Simulation gestartet, siehe Abbildung 4.18 links. Die Sprachbefehle *Faster*, *Slower* und *Pause Simulation* beschleunigen, verlangsamen oder pausieren die Simulation. Während der Simulation wird das Layout mit Hilfe der Visualisierungen analysiert. Es fällt auf, dass die Handarbeitsstation voll ausgelastet ist (grüne Farbe), während die Roboterstation fast nicht ausgelastet ist (rote Farbe). Durch das Markieren von Objekten mit dem Cursor können detaillierte Informationen in Form von Popups eingeblendet werden, siehe Abbildung 4.18 rechts. In diesem Fall wird die



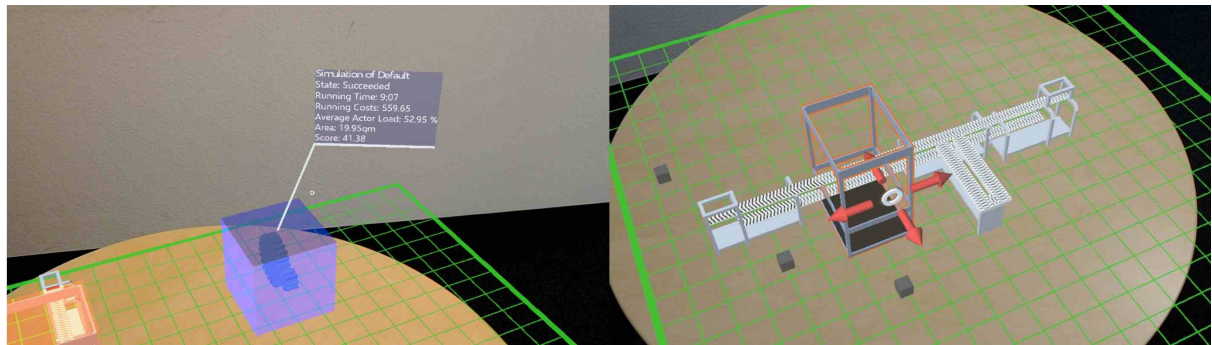
**Abbildung 4.17:** Links: Alle verfügbaren Layouts werden dargestellt. Der Benutzer wählt eines aus. Rechts: Das ausgewählte Layout wird auf der im vorherigen Schritt platzierten Grundplatte dargestellt.



**Abbildung 4.18:** Links: Eine gestartete Simulation. Zu sehen sind Werkstücke beim durchlaufen des Produktionsprozesses und Visualisierungen zur Auslastung. Rechts: Detailinformationen zur Roboterstation. Angezeigt wird die Auslastung, der Zustand und die Zustandshistorie.

Auslastung der Roboterstation von 21% ausgelesen. Um die Roboterstation stärker auszulasten, könnte eine zweite Handarbeitsstation parallel zur ersten eingebaut werden.

Nach Abschluss der Simulation kann die Bewertung des Layouts über die Detailinformationen der Simulationsbox abgefragt werden, siehe Abbildung 4.19 links. Um die Bewertung des Layouts zu verbessern wird in diesem Beispiel die Kurve aus dem Layout entfernt um Platz und Kosten einzusparen. Dazu wird zuerst die Kurve markiert und mit dem Sprachbefehl *Delete Selection* gelöscht. Anschließend werden die verbleibenden Elemente rotiert und verschoben um wieder mit der restlichen Anlage verbunden zu sein. Dazu kommen die Sprachbefehle *Rotate Object* und *Move Object* sowie die Manipulationsgeste zum Einsatz. Das resultierende Layout ist in Abbildung 4.19 rechts zu sehen. Um die Änderungen im Layout zu speichern muss zuerst der Sprachbefehl *Layout Selection Mode* gesprochen werden und anschließend im sich öffnenden Interface der Button *Save All Changes* ausgewählt werden.



**Abbildung 4.19:** Links: Detailinformation zum Simulationsergebnis, dazu gehören alle Metriken und die daraus folgende Bewertung des Layouts. Rechts: Das umkonfigurierte Layout. Die Ecke wurde entfernt und die Roboterstation und der Bauteilaufzug wurden gedreht und wieder mit dem restlichen Layout verbunden.

### 4.5.3 Anwendung des Prototyps in einer realen Umgebung

Der Einsatz des Layout-Simulators soll nicht nur auf Büroräume beschränkt sein, sondern auch in Fabrikhallen funktionieren. Dabei treten vor allem Schwächen der HoloLens-Hardware auf, die unabhängig von der verwendeten Anwendung sind.

In einer leeren Fabrikhalle, in die beispielsweise ein Layout gestellt werden soll, um die Platzverhältnisse besser zu verstehen, befinden sich wenige Objekte und kahle Wände. Die HoloLens benötigt aber Objekte oder Strukturen, um sich zu orientieren. Befinden sich keine oder wenige Merkmale in der Umgebung, wechselt die HoloLens in einen sogenannten „Limited Mode“ und kann nur noch 3D-Objekte relativ zur Kamera und nicht statisch in der Welt platzieren. Mögliche Lösungen wären das Hinzufügen von künstlichen Merkmalen wie zum Beispiel Werbebanner, Kleiderständer oder ähnliches.

Ein weiteres Problem ist die Verwendung der HoloLens in großen Menschenmassen. Da die HoloLens ein innovatives Produkt ist, wird diese oft auf Messen zu Marketingzwecken ausgestellt. Für ein zuverlässiges Tracking der HoloLens muss die Umgebung aber statisch sein. Bei einer bewegenden Umgebung ist das Tracking der HoloLens nicht stabil. Die Lösung wäre, die HoloLens nur in kleinen Gruppen zu verwenden oder genügend sichtbare und statische Merkmale in der Umgebung zu haben.

In manchen Situation, zum Beispiel beim Überprüfen der Platzverhältnisse eines Layouts, wird dieses in seiner originalen Größe dargestellt. Die Anzeigen und Visualisierungen sind aber auf Modellgröße optimiert. Dies führt zu unerwarteten Effekten bei der Anzeige von Simulationsergebnissen. Ein Beispiel dafür ist die Visualisierung der Auslastung einer Roboterstation, siehe Abbildung 4.11. Dabei wird die komplette Roboterstation von einem transparenten Quader umgeben. Steht man vor der Roboterstation in realer Größe, ist das komplette Sichtfeld mit diesem transparenten Quader ausgefüllt. Das wirkt verwirrend und trägt nicht zum Verständnis

des Layouts bei. Erst beim Betrachten aus größerer Entfernung entfaltet die Visualisierung wieder ihre Wirkung.

#### 4.5.4 Benutzbarkeit

Da die HoloLens nur zwei Handgesten kennt, von denen nur eine zur Interaktion mit Anwendungen verwendet werden kann, ist die Bedienung der HoloLens einfach zu erlernen. Nach einer kurzen Eingewöhnungszeit mit der neuen Technologie können Anwendungen flüssig bedient werden. Die Benutzung des Layout-Simulators ist dementsprechend einfach zu erlernen. Die erste Benutzung sollte allerdings unter Anweisung ausgeführt werden um neue Benutzer nicht zu frustrieren, da das Grundprinzip sich im Vergleich zu einer Desktopanwendung bedeutend verändert hat.

Die Anwendung wird durch eine Vielzahl an Sprachbefehlen gesteuert. Um nicht alle Sprachbefehle im vornherein auswendig lernen zu müssen, können alle verfügbaren Sprachbefehle jederzeit in der Anwendung angezeigt werden. Ein weiteres Problem der Sprachbefehle ist die Aussprache. Die Spracherkennung der HoloLens arbeitet bisher nur in englischer Sprache. Dies führt bei nicht-englischen Muttersprachlern oder bei Menschen mit Akzent zu Problemen. In der Anwendung ist auf einfach auszusprechende Sprachbefehle geachtet. Trotzdem kommt es vor, dass Sprachbefehle nicht zuverlässig erkannt werden und aus diesem Grund mehrfach gesprochen werden müssen. Werden Sprachbefehle auch nach mehreren Versuchen nicht erkannt, führt dies zu Frust beim Benutzer. Aus diesem Grund sollten unterschiedliche Eingabemöglichkeiten verwendet werden um dem Benutzer eine Alternative zum Sprachbefehl zu geben. Möglichkeiten dafür wären zum Beispiel die Integration von Schaltflächen in Popups, oder das Anzeigen eines Kontextmenüs, wenn die *Air-Tap*-Geste zweimal schnell hintereinander ausgeführt wird.

#### 4.5.5 Vergleich der HoloLens- und Desktop-Version des Layout-Simulators

Die Portierung und Anpassung des Layout-Simulators bietet einige Vorteile, aber auch Nachteile. Diese sollen in diesem Abschnitt betrachtet werden. Dabei sollte der Layout-Simulator für die HoloLens nicht als Weiterentwicklung der Desktop-Version gesehen werden, sondern als parallele Anwendung auf einer anderen Technologie.

Wird nur der aktuelle Funktionsumfang betrachtet, ist die Desktop-Version überlegen, da noch nicht alle Features in der HoloLens-Version eingebaut sind. Wie in Abschnitt 4.5.1 beschrieben hat dies keine technischen Gründe, sondern ist der zeitlichen Begrenzung der Arbeit geschuldet. Auf der HoloLens könnte der volle Funktionsumfang der Anwendung implementiert werden.

Vergleicht man die Zeit die ein Benutzer benötigt, um verschiedene Aufgaben innerhalb der Anwendung auszuführen, ist die Desktop-Version vorzuziehen. Ein durchschnittlicher Benutzer

ist die Verwendung mit Maus und Tastatur gewöhnt und kann deshalb schneller arbeiten als mit Blickrichtung und Handgesten. Auch für längere Arbeiten ist die Desktop-Version überlegen, da der Akku der HoloLens nur für circa zwei Stunden hält und die HoloLens nach längerer Benutzung unangenehm auf den Kopf sitzt.

In der HoloLens-Anwendung kommen bereits die Grenzen der Rechenleistung der HoloLens zum Tragen. Wird versucht in der HoloLens größere Layouts mit Hunderten oder Tausenden Elementen darzustellen, kommt die aktuelle Technik an ihre Grenzen. Während Desktop-PCs deutlich leistungsfähigere Hardware besitzen und auch mit geringeren Bildwiederholungsraten noch einsetzbar sind, muss bei der HoloLens viel Arbeit in Optimierungen gesteckt werden um größere Layouts trotzdem anzeigen zu können.

Die Vorteile der HoloLens-Version liegen vor allem beim Informationstransport und bei kommunikativen Aufgaben. Durch die dreidimensionale Darstellung können Zusammenhänge des Layouts besser verstanden werden. Wird das Layout auf Originalgröße skaliert, werden die räumlichen Dimensionen des Layouts erfassbar. Die Handgesten erlauben eine intuitive, aber unpräzise Manipulation der Layoutelemente. Durch das Snapping spielt die geringe Präzision aber keine große Rolle. Vor allem in Szenarien, in denen mehrere Personen über ein Layout diskutieren, ist die HoloLens der Desktop-Version überlegen. Da alle beteiligten das gleiche sehen, kann auf natürliche Weise diskutiert werden, als würde die Anlage wirklich am Einsatzort stehen.



# 5 Zusammenfassung & Fazit

Im Folgenden wird zuerst der Inhalt der Masterarbeit zusammengefasst und anschließend ein Fazit über die Masterarbeit gezogen.

## 5.1 Zusammenfassung

Ziel dieser Masterarbeit war es, herauszufinden, ob die Microsoft HoloLens zur visuellen Analyse in Produktion und Fertigung geeignet ist. Dazu wurden im ersten Teil der Arbeit die Möglichkeiten der Microsoft HoloLens analysiert. Im zweiten Teil der Arbeit wurde ein Konzept für die Portierung des Layout-Simulators von Wörner et al. [WE11; WE13] auf die HoloLens entwickelt und anschließend in einer prototypischen Anwendung evaluiert.

Um die Möglichkeiten der HoloLens zu analysieren wurden verschiedene Funktionalitäten getestet. Dazu gehören die Evaluierung der Grafikleistung, das Analysieren der Technologien „Holographic Sharing“ und „Holographic Remoting“, das Testen der Qualität der markerbasierten Objekterkennung und das Messen der Auflösung und Genauigkeit des dreidimensionalen Umgebungsmodells.

Auf Basis der Möglichkeiten der Microsoft HoloLens wurde ein Konzept für den Layout-Simulator entwickelt. Ziel war es dabei die Desktop-Anwendung nicht nur zu portieren, sondern auch für die Augmented-Reality-Technologie anzupassen. Besonderes Augenmerk wurde dabei auf die Interaktion zwischen Benutzer und Anwendung gelegt, da diese sich grundsätzlich zur Interaktion mit einer Desktop-Anwendung unterscheidet. Ein weiterer wichtiger Punkt war die Visualisierung der Simulationsergebnisse. Dieses Konzept wurde in Form einer prototypischen Anwendung für die Microsoft HoloLens implementiert. Abschließend wurde der Prototyp evaluiert und mit der vorhandenen Desktop-Version des Layout-Simulators verglichen.

## 5.2 Fazit

Die Microsoft HoloLens ist die erste Augmented-Reality-Brille die autark arbeitet. Sie besitzt durchsichtige Displays. Damit können die Benutzer während der Benutzung weiterhin die reale Welt sehen und sind nicht, wie bei Virtual Reality, von der Außenwelt abgeschnitten. Dies ist vor allem bei der Zusammenarbeit mit mehreren Personen ein großer Vorteil. Durch

diese Kombination ergeben sich eine große Anzahl an Einsatzmöglichkeiten die zuvor nicht möglich waren.

Allerdings hat auch die HoloLens nicht nur Vorteile. Die Interaktion ist vor allem bei der erstmaligen Verwendung gewöhnungsbedürftig. Die Interaktion mit der Blickrichtung und zwei vordefinierten Handgesten ist weniger Benutzerfreundlich als beispielsweise Handcontroller, wie sie bei aktuellen Virtual-Reality-Brillen zum Einsatz kommen. Das geringe Sichtfeld sorgt dafür das ein großes Objekt nicht auf einen Blick erfasst werden kann. Dies mindert die Immersion. Nach einiger Zeit gewöhnt man sich aber an das geringe Sichtfeld.

In dieser Arbeit wurde die Tauglichkeit der Brille zur visuellen Analyse in Produktion und Fertigung überprüft. Das Ergebnis dieser Prüfungen zeigt, dass die Brille in vielen aber nicht allen Szenarios eingesetzt werden kann. Vor allem die geringe Rechenleistung und Grafikleistung der Brille limitiert die Darstellung von realen Produktionsstraßen und komplexen Visualisierungen. Aus diesem Grund müssen die Berechnungen auf der Brille einfach gehalten werden oder die aufwendigen Berechnungen ausgelagert werden. Komplexe Visualisierungen sind nur mit Einschränkungen möglich. Auch bei der Objekterkennung müssen Einschränkungen in Kauf genommen werden. Die Kamera besitzt nur eine geringe Auflösung und die Genauigkeit des 3D Umgebungsnetzes ist zu grob um feine Strukturen zu erkennen. Ein weiteres Problem ist, dass die Brille eine Umgebung benötigt in der sie sich orientieren kann.

Die *Holographic Sharing* Technologie ist zwar eine gute Idee um das Problem der geringen Rechenleistung zu lösen. Die vorhandenen Nachteile, vor allem die reduzierte Hologrammstabilität macht die Benutzung dieser Technologie aber unattraktiv. Stattdessen sollte Zeit mit dem Optimieren verbracht werden, um zwar weniger komplexe aber dafür stabile Hologramme anzuzeigen.

Komplexe Algorithmen, wie beispielsweise der evolutionäre Algorithmus des Layout-Simulators, können zwar auf der HoloLens ausgeführt werden. Allerdings ist der eingebaute Prozessor kein Hochleistungsrechner. Aus diesem Grund kann es bei aufwendigeren Berechnungen sinnvoller sein die Berechnung von einem Server berechnen zu lassen und nur die Ergebnisse darzustellen.

Wie schon erwähnt ersetzt die Portierung des Layout-Simulators auf die HoloLens die Desktop-Anwendung nicht. Stattdessen sollte die beiden Versionen parallel eingesetzt werden. Zur produktiven Planung sollte die Desktop-Version verwendet werden. Zur Diskussion und zu Besprechungen ist die HoloLens-Version besser geeignet. Die HoloLens ist ein neues und innovatives Gerät mit dem viele Personen gerne arbeiten. Dies ist ein Vorteil. Besprechungen in der Werkshalle in denen jeder eine virtuelle Anlage durch die HoloLens sieht sind interessanter, als eine Besprechung in einem Büro, bei der alle Personen die gleiche zweidimensionale Darstellung der Anlage betrachten.

## 6 Ausblick

In diesem Kapitel wird ein Ausblick gegeben, wie der Fabriklayout-Simulator für die HoloLens weiterentwickelt werden kann und welche Perspektiven die Technologie in Produktion und Fertigung hat.

Wie schon im Grundlagenkapitel über Virtual und Augmented Reality beschrieben, ist die Bildwiederholungsrate bei Augmented-Reality-Anwendungen besonders kritisch. Der Layout-Simulator erfüllt diese Anforderungen nicht. Jedoch können durch verschiedene Optimierungen diese Ziele erfüllt werden. Ein erstes Optimierungspotenzial liegt in der Darstellung des Transportbandes. Dieses besteht aus vielen kleinen, gleich großen Quadern. Durch *Batching* dieser Elemente wird das Interface zwischen Grafikkarte und CPU entlastet. Eine zweite Möglichkeit wäre das Transportband nicht durch viele kleine Quader, sondern durch ein 3D-Objekt darzustellen. Dies würde zwar das Einfärben des Transportbandes komplizierter gestalten, würde aber die Grafikkarte entlasten.

Weitere Erkenntnisse über die HoloLens würden durch das Einbinden des Holographic Sharing entstehen. Durch die große Anzahl der Daten im Layout-Simulator ist dies nicht einfach umzusetzen. Eine Möglichkeit, dieses Problem zu lösen, wäre das Auslagern der Layout-Simulation auf einen externen PC oder Server. Die HoloLens-Brille wären damit nur Anzeigegerät und könnte die Simulationsdaten vom einem Server abfragen. Eine weitere Möglichkeit wäre die Simulation isoliert auf jedem Gerät zu starten. Zur Synchronisation müssten dann nur Start, Stopp und Geschwindigkeit der Simulation ausgetauscht werden. Allerdings muss dazu sichergestellt sein, dass auf allen Geräten ein identisches Layout geladen ist.

Die in dieser Arbeit besprochenen Verfahren zur Objekterkennung basieren auf Markern. Das Problem dieser Technik ist, dass die Marker an die Anlagen angebracht werden müssen und dem System mitgeteilt werden muss, auf welchen Positionen diese Marker befestigt sind. Eine von der Verwendung unkompliziertere Lösung wäre die markerlose Erkennung der Anlagen. Dazu könnten Verfahren des maschinellen Lernens eingesetzt werden. Diese Verfahren sind aber zu aufwendig für die HoloLens und müssten auf einen Server ausgelagert werden.

An einigen Stellen in der Anwendung werden klassische zweidimensionale Menüs mit Schaltflächen verwendet welche die dritte Dimension komplett vernachlässigen. Um das volle Potenzial der Brille auszuschöpfen müssen neue dreidimensionale Menükonzepte entwickelt werden. Eine Möglichkeit wäre statt einer textuellen Liste von verfügbaren Layouts einen dreidimensionalen Schrank zu verwenden in welchem in jedem Fach eine Vorschau der Layouts liegt. Durch herausnehmen eines Layouts wird dieses geladen.

Um das volle Potenzial der Anwendung auszunutzen, sollte diese nicht isoliert betrachtet werden, sondern Teil eines Fabrikplanungs- und Bauprozesses oder einer Fabrikplanungssoftware sein. Dabei soll diese Anwendung nicht nur ein Teilproblem lösen sondern den kompletten Lebenszyklus einer Fabrikhalle durchleben. Von der Konstruktion über den Bau und die laufende Produktion bis zur Entsorgung. Mit Hilfe der HoloLens kann zu jeder Zeit der Stand der Planung, der aktuelle Baufortschritt oder der Status der Produktion abgefragt werden. So haben alle an Bau, Planung und Produktion Beteiligten einen gemeinsamen Stand, auf dem diskutiert werden kann. Befindet man sich in der Halle, kann beispielsweise während dem Bau, der Soll-Zustand über den aktuellen realen Baufortschritt gelegt werden. So werden Probleme, wie zum Beispiel zu kleine Abstände, mit höherer Wahrscheinlichkeit erkannt. Eine weitere Möglichkeit, wäre die Kopplung der Simulation mit der laufenden Produktion. Diese synchronen Daten können vor allem bei Ausfällen zur Fehleranalyse eingesetzt werden.

Um diese integrierte Lösung zu realisieren, muss die HoloLens-Anwendung 3D-Modelle nachladen und anzeigen können. Probleme dabei sind vor allem unterschiedliche Datenformate und die hohe Komplexität klassischer 3D-Modelle. Die Umwandlung von CAD-Datenformaten in Polygon-Datenformate ist automatisch mit bestehenden Bibliotheken möglich. Ein größeres Problem ist die Komplexität der Modelle. Da die HoloLens ein mobiles Gerät mit geringer Grafikleistung ist, müssen die Modelle stark vereinfacht werden. Da die manuelle Vereinfachung von Hand aufwendig und zeitaufwendig ist, sollte eine automatisierte Vereinfachung verwendet werden.

Entwickelt sich der Trend von Augmented Reality weiter, werden neue und bessere Produkte auf den Markt kommen. Aktuelle Probleme der HoloLens wie zum Beispiel die geringe Grafikleistung oder die eingeschränkte Handgestenerkennung werden gelöst werden. Dadurch ergeben sich neue und weniger eingeschränkte Einsatzgebiete der Augmented-Reality Brillen in Produktion und Fertigung.

# Literaturverzeichnis

- [AIT16] AIT GmbH Co. KG. *AIT Leistungsspektrum*. 2016. URL: <http://www.aitgmbh.de/leistungsspektrum.html> (besucht am 24. 11. 2016) (zitiert auf S. 10).
- [ART16] ARToolKit. *ARToolKit Documentation*. 2016. URL: <https://artoolkit.org/documentation/> (besucht am 02. 12. 2016) (zitiert auf S. 40, 71).
- [Car14] J. Carmack. *John Carmack's Latency Mitigation Strategies*. 2014. URL: <https://www.twentymillisecons.com/post/latency-mitigation-strategies/> (besucht am 24. 11. 2016) (zitiert auf S. 35).
- [CLZG16] I. Calvo, F. López, E. Zulueta, P. González-Nalda. „Towards a methodology to build virtual reality manufacturing systems based on free open software technologies“. In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* (2016), S. 1–12 (zitiert auf S. 27, 28).
- [CM92] T. P. Caudell, D. W. 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*. Bd. 2. IEEE. 1992, S. 659–669 (zitiert auf S. 23, 24).
- [DB91] T. A. Defanti, M. D. Brown. „Visualization in scientific computing“. In: *Advances in Computers* 33 (1991), S. 247–307 (zitiert auf S. 20).
- [DSAP03] F. Doil, W. Schreiber, T. Alt, C. Patron. „Augmented reality for manufacturing planning“. In: *Proceedings of the workshop on Virtual environments 2003*. ACM. 2003, S. 71–76 (zitiert auf S. 25, 26).
- [Fia05] M. Fiala. „ARTag, a fiducial marker system using digital techniques“. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Bd. 2. IEEE. 2005, S. 590–596 (zitiert auf S. 19).
- [Gar17] Gartner Inc. *Gartner's 2016 Hype Cycle for Emerging Technologies*. 2017. URL: <http://www.gartner.com/newsroom/id/3412017> (besucht am 16. 02. 2017) (zitiert auf S. 9).
- [HF11] S. Henderson, S. Feiner. „Exploring the benefits of augmented reality documentation for maintenance and repair“. In: *IEEE transactions on visualization and computer graphics* 17.10 (2011), S. 1355–1368 (zitiert auf S. 24, 25).
- [HM90] R. B. Haber, D. A. McNabb. „Visualization idioms: A conceptual model for scientific visualization systems“. In: *Visualization in scientific computing* 74 (1990), S. 93 (zitiert auf S. 22).

- [How15] HowStuffWorks.com. *How many senses does a human being have?* 2015. URL: <http://science.howstuffworks.com/life/question242.htm> (besucht am 26. 10. 2016) (zitiert auf S. 11).
- [Ins17] Institut für Industrielle Fertigung und Fabrikbetrieb (IFF). *Physische Modellfabrik iTRAME*. 2017. URL: <http://lf.iff.uni-stuttgart.de/schulungszentrum/itrframe/> (besucht am 08. 03. 2017) (zitiert auf S. 51).
- [KKEM10] D. Keim, J. Kohlhammer, G. Ellis, F. Mansmann. *Mastering the information age solving problems with visual analytics*. Eurographics Association, 2010 (zitiert auf S. 23).
- [KL00] B. Korves, M. Loftus. „Designing an immersive virtual reality interface for layout planning“. In: *Journal of Materials Processing Technology* 107.1 (2000), S. 425–430 (zitiert auf S. 25).
- [MG96] T. Mazuryk, M. Gervautz. *Virtual Reality - History, Applications, Technology and Future*. Techn. Ber. Institute of Computer Graphics und Algorithms, Vienna University of Technology, 1996 (zitiert auf S. 11).
- [Mic16a] Microsoft Corporation. *Color Design*. 2016. URL: [https://developer.microsoft.com/de-de/windows/holographic/color\\_design](https://developer.microsoft.com/de-de/windows/holographic/color_design) (besucht am 10. 10. 2016) (zitiert auf S. 15).
- [Mic16b] Microsoft Corporation. *Hardware Details*. 2016. URL: [https://developer.microsoft.com/en-us/windows/holographic/hardware\\_details](https://developer.microsoft.com/en-us/windows/holographic/hardware_details) (besucht am 10. 10. 2016) (zitiert auf S. 13, 14).
- [Mic16c] Microsoft Corporation. *Hologram*. 2016. URL: <https://developer.microsoft.com/en-us/windows/holographic/hologram> (besucht am 07. 02. 2017) (zitiert auf S. 13, 15).
- [Mic16d] Microsoft Corporation. *Shared holographic experiences*. 2016. URL: [https://developer.microsoft.com/de-DE/windows/holographic/shared\\_holographic\\_experiences](https://developer.microsoft.com/de-DE/windows/holographic/shared_holographic_experiences) (besucht am 02. 12. 2016) (zitiert auf S. 38).
- [Mic17a] Microsoft Corporation. *Designing for mixed reality*. 2017. URL: [https://developer.microsoft.com/de-de/windows/holographic/designing\\_for\\_mixed\\_reality](https://developer.microsoft.com/de-de/windows/holographic/designing_for_mixed_reality) (besucht am 20. 02. 2017) (zitiert auf S. 16).
- [Mic17b] Microsoft Corporation. *HoloToolkit-Unity*. 2017. URL: <https://github.com/Microsoft/HoloToolkit-Unity> (besucht am 07. 02. 2017) (zitiert auf S. 30, 71).
- [Mic17c] Microsoft Corporation. *Performance recommendations*. 2017. URL: [https://developer.microsoft.com/de-de/windows/holographic/performance\\_recommendations](https://developer.microsoft.com/de-de/windows/holographic/performance_recommendations) (besucht am 08. 03. 2017) (zitiert auf S. 74).
- [Mic17d] Microsoft Corporation. *Use gestures*. 2017. URL: <https://support.microsoft.com/en-us/help/12644/hololens-use-gestures> (besucht am 21. 02. 2017) (zitiert auf S. 17, 18).

- [MSP16] MSPoweruser. *Microsoft reveals HoloLens Holographic Processing Unit has 24 cores capable of 1 trillion FLOPS*. 2016. URL: <https://mspoweruser.com/microsoft-reveals-hololens-holographic-processing-unit-24-cores-capable-1-trillion-flops> (besucht am 26. 10. 2016) (zitiert auf S. 13).
- [MTUK95] P. Milgram, H. Takemura, A. Utsumi, F. Kishino. „Augmented reality: A class of displays on the reality-virtuality continuum“. In: *Photonics for industrial applications*. International Society for Optics und Photonics. 1995, S. 282–292 (zitiert auf S. 12).
- [PTC17a] PTC Inc. *Vuforia Developer Portal*. 2017. URL: <https://developer.vuforia.com/> (besucht am 09. 02. 2017) (zitiert auf S. 21, 71).
- [PTC17b] PTC Inc. *VuMark*. 2017. URL: <https://library.vuforia.com/articles/Training/VuMark> (besucht am 16. 02. 2017) (zitiert auf S. 19).
- [RB75] J. T. Reason, J. J. Brand. *Motion sickness*. Academic press, 1975 (zitiert auf S. 13, 29).
- [Rei17] J. Reinhardt. *HoloLens: Evaluierung der 3D Grafikleistung*. 2017. URL: <http://blog.aitgmbh.de/2017/02/07/hololens-evaluierung-der-3d-grafikleistung/> (besucht am 07. 02. 2017) (zitiert auf S. 29).
- [Rek98] J. Rekimoto. „Matrix: A realtime object identification and registration method for augmented reality“. In: *Computer Human Interaction, 1998. Proceedings. 3rd Asia Pacific*. IEEE. 1998, S. 63–68 (zitiert auf S. 19).
- [RKK+07] P. Riffelmacher, S. Kluge, R. Kreuzhage, V. Hummel, E. Westkämper. „Learning factory for the manufacturing industry: Digital learning shell and a physical model factory-iTRAME for production engineering and improvement“. In: *Proceedings of the 20th International Conference on Computer-Aided Production Engineering*. 2007, S. 120–131 (zitiert auf S. 51).
- [Thy17] Thyssenkrupp AG. *Microsoft HoloLens verringert Wartungszeit*. 2017. URL: <https://www.thyssenkrupp.com/de/newsroom/pressemeldungen/press-release-114208.html> (besucht am 07. 02. 2017) (zitiert auf S. 9).
- [WE11] M. Wörner, T. Ertl. „Visual Analysis of Advanced Manufacturing Simulations“. In: *International Workshop on Visual Analytics EuroVA*. Bd. 2011. 2011, S. 29–32. DOI: 10.2312/PE/EuroVAST/EuroVA11/029-032. URL: <http://dx.doi.org/10.2312/PE/EuroVAST/EuroVA11/029-032> (zitiert auf S. 9, 51, 54, 65, 81).
- [WE13] M. Wörner, T. Ertl. „Simulation-based Visual Layout Planning in Advanced Manufacturing“. In: *2013 46th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2013, S. 1532–1541. DOI: 10.1109/HICSS.2013.482. URL: <http://dx.doi.org/10.1109/HICSS.2013.482> (zitiert auf S. 9, 51, 54, 65, 81).
- [Wor17] World Wide Web Consortium. *Speech Recognition Grammar Specification Version 1.0*. 2017. URL: <https://www.w3.org/TR/speech-grammar/> (besucht am 26. 03. 2017) (zitiert auf S. 18).

- [YON16] A. Yew, S. Ong, A. Nee. „Towards a griddable distributed manufacturing system with augmented reality interfaces“. In: *Robotics and Computer-Integrated Manufacturing* 39 (2016), S. 43–55 (zitiert auf S. 25, 27).



## **Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Ort, Datum, Unterschrift