

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Masterarbeit

**Untersuchung von
Automatisierungsstufen beim
dynamischen Überlagern physischer
und digitaler Entitäten**

Erik-Felix Tinsel

Studiengang: Softwaretechnik

Prüfer/in: Jun.-Prof. Michael Sedlmair

Betreuer/in: Tobias Korb, M.Sc.

Beginn am: 16. November 2018

Beendet am: 16. Mai 2019

Kurzfassung

Durch den Einsatz moderner Mixed Reality Datenbrillen entsteht eine Vielzahl an Herausforderungen zur Anpassung einer virtuellen Umgebung an deren physischen Ursprung. Ein Schwerpunkt ist dabei, real existierende Entitäten sensorisch aufzunehmen und digital zu verarbeiten, um deren virtuelle Abbildungen so im Sichtfeld des Benutzers zu platzieren, dass sie die physischen Ebenbilder zu jedem Zeitpunkt nahezu ideal überlagern. Forschung und Industrie entwickelten unterschiedliche Ansätze und Konzepte zur Bewältigung dieser Aufgabe. Zur Schaffung einer Übersicht und der Möglichkeit der grundsätzlichen Abwägung bei der Wahl eines Überlagerungskonzepts, ergründet diese Arbeit bestehende Techniken der virtuellen Überlagerung auf physischen Entitäten und vergleicht diese daraufhin mit dem Fokus auf den Grad der Automatisierung und dem Einsatz in der Produktion.

Inhaltsverzeichnis

1	Einleitung	13
1.1	Motivation	13
1.2	Zielsetzung	15
1.3	Aufbau der Arbeit	15
2	Grundlagen	17
2.1	Automatisierungsgrade	17
2.2	Mixed Reality	18
2.3	Objekterkennung	21
3	Stand der Technik	33
3.1	Augmented Reality SDKs	33
3.2	Datenbrillen	37
3.3	Einsatzgebiete	39
3.4	Verwandte Arbeiten	44
4	Untersuchungsaufbau und -durchführung	47
4.1	Beschreibung der Überlagerungskonzepte	47
4.2	Messgrößen und Metriken	51
4.3	Vorgehen	56
5	Evaluation	61
5.1	Automatisierung Informationsaufnahme	61
5.2	Automatisierung Informationsverarbeitung	62
5.3	Automatisierung bei der Entscheidungsfindung	62
5.4	Automatisierung bei der Ausführung der Aktion	63
5.5	Überlagerungsgenauigkeit	63
5.6	Stabilität	64
5.7	Ressourcenverbrauch CPU	69
5.8	Ressourcenverbrauch RAM	71
5.9	Zeitaufwand	72
5.10	Gesamtübersicht	73
6	Schlussbetrachtung	75
6.1	Zusammenfassung	75
6.2	Ausblick	76
	Literaturverzeichnis	77

Abbildungsverzeichnis

1.1	Abbildung einer virtuellen Komponente auf eine physische Entität	14
2.1	Mixed Reality-Spektrum nach Milgram und Kishino	19
2.2	Mixed Reality-Einflussfaktoren nach Bray et al.	20
2.3	Auswahl planarer Markersysteme nach Fiala	24
2.4	NFT-Algorithmus nach Neumann und You	27
2.5	Anzahl der Vergleiche des SIFT-Algorithmus	29
2.6	Bestimmung der Ausrichtung eines Merkmalpunkts in SURF	31
3.1	ChArUco-Brett	35
3.2	Erfassung mittels des Vuforia Object Scanner	37
3.3	Microsoft HoloLens	38
3.4	Visual Inspection and Documentation GUI	42
4.1	Technische Umsetzung der manuellen Positionierung	49
4.2	Referenzmarker des Versuchsaufbaus	50
4.3	Technische Umsetzung der modellbasierten Positionierung	51
4.4	Aufnahme des Versuchsaufbaus	58
5.1	Aufnahme der manuellen Positionierung aus der Position nah	64
5.2	Durchschnittliche Abweichung der Überlagerung in Millimetern	65
5.3	Standardabweichung der Überlagerung	65
5.4	Kastengrafik der Abweichung der Überlagerung in Millimetern	65
5.5	Durchschnittliche Abweichung aus Sicht der übrigen Positionen in Millimetern	67
5.6	Standardabweichung der Abweichung aus Sicht der übrigen Positionen	67
5.7	Kastengrafik der Abweichung aus Sicht der übrigen Positionen in Millimetern	67
5.8	Durchschnittliche Auslastung der CPU in Prozent	70
5.9	Standardabweichung der Auslastung der CPU	70
5.10	Durchschnittlicher RAM-Verbrauch in Prozent	71
5.11	Standardabweichung des RAM-Verbrauchs	72
5.12	Insgesamt benötigte Vorbereitungszeit in Minuten	73

Tabellenverzeichnis

4.1	Zuweisung der Automatisierungsstufen	53
4.2	Zuweisung der Überlagerungsgenauigkeit	54
4.3	Zuweisung der Stabilität	54
4.4	Zuweisung des Ressourcenverbrauchs (CPU)	55
4.5	Zuweisung des Ressourcenverbrauchs (Arbeitsspeicher)	55
4.6	Zuweisung des Zeitaufwands	56
5.1	Einzelergebnisse der Messung zur Überlagerungsgenauigkeit	66
5.2	Einzelergebnisse der Messung zur Stabilität	68
5.3	Einzelergebnisse der Messung zur Auslastung der CPU	70
5.4	Einzelergebnisse der Messung des RAM-Verbrauchs	71
5.5	Analyseergebnisse als Harvey Ball-Diagramm	74

Verzeichnis der Algorithmen

2.1	ArUco Algorithmus zur Auswahl von Markern	25
-----	---	----

1 Einleitung

Eine wachsende Zahl an Unternehmen nutzt die Möglichkeiten des Einsatzes von Mixed Reality-Technologien, um ihre Produkte interessierten Kunden vorzustellen [Fit11][Keh18]. So kann zum Beispiel mit Hilfe einer Datenbrille das neuste Serienfahrzeug direkt auf die reale Welt projiziert werden, damit der Benutzer das Automobil von allen Seiten aus betrachten kann [Keh17]. Weck [Wec06] stellt in seinem Buch *Werkzeugmaschinen 4: Automatisierung von Maschinen und Anlagen* zudem ein Beispiel vor, bei dem eine Aufspannvorrichtung effizienter an ein Werkstück angebracht werden kann, indem sie für den Facharbeiter virtuell über das physisch vorliegende Werkstück überlagert wird.

Eine solche Abbildung kann auch dazu genutzt werden, um eine in der Umgebung bestehende Entität mit Informationen anzureichern, welche ausschließlich oder zu großen Teilen digital vorliegen. Auf Maschinen und Anlagen in der Produktion können so erweiterte Informationen zu den einzelnen Produktionsschritten eingeblendet oder ein digitaler Testlauf simuliert werden. Dieser Einsatz verschafft dem Endanwender eine zusätzliche Informationsschicht, welche er gezielt für die Tätigkeit an der physischen Entität nutzen kann. Ein exemplarischer Überlagerungsvorgang aus der Sicht durch eine Datenbrille ist in Abbildung 1.1 zu sehen. Das rot dargestellte virtuelle Modell des Förderbands muss an das reale Förderband angepasst werden. Liegen Modell sowie reales Urbild schließlich räumlich aufeinander, kann ein digitaler Testlauf durchgeführt werden, ohne die reale Anlage tatsächlich zu betätigen. Ebenso können Anleitungen eingeblendet werden, die dem Betrachter eine korrekte Bedienung erläutern. Eine solche Einblendung trägt in Teilen dazu bei Reparaturzeiten zu verkürzen und senkt somit die Kosten, die einem Unternehmen entstehen [Wec06].

Um einen Überblick zu geben, wie diese Arbeit in das beschriebene Themengebiet eingebettet ist, begründet dieses Kapitel in Abschnitt 1.1 die Idee einer Gegenüberstellung von Überlagerungstechniken auf Grundlage von Automatisierungsstufen. Abschnitt 1.2 definiert dabei sämtliche Ziele, die mit dieser Vorgabe zu lösen sind und Abschnitt 1.3 fasst die unterschiedlichen Kapitel dieser Arbeit schließlich kurz zusammen.

1.1 Motivation

Dangelmaier et al. [DFG+05] beschreiben die Integration von AR-Technologien in Simulationsprozessen, bedingt durch verkürzte Produkterstellungszyklen. Dabei weisen sie auf die Vorzüge hin, die sich aus einer solchen Eingliederung ergeben kann. Die Möglichkeit einer Überblendung von Informationen auf die reale Produktion ermöglicht es, die Planung von Anlagen zu übernehmen und vor Ort zu visualisieren. Ebenso können bestehende Anlagen überprüft werden oder um einzelne Objekte, zunächst rein virtuell, erweitert werden. Die Flexibilität des Einsatzes der Augmented Reality in unmittelbarer Anwendungsumgebung wird damit zu einem großem Vorteil der Technologie. Ein statischer Einsatz von AR-Geräten würde diesen Vorteil vermindern, da ein erhöhter

Aufwand der Installation und der Verlagerung in andere Abschnitte der Anlage die Effektivität übersteigen könnte. Ein dynamischer Einsatz der beschriebenen Technologie erfordert dabei, dass die Funktionalität dieser dem Anwendungsfall entsprechend verlässlich auszuführen sind.

Der Einsatz des einleitend beschriebenen Überlagerungsvorgangs in einem Industrieprozess erfordert somit eine Abwägung bei der Auswahl der Konzepte und Technologien, da diese in unterschiedlichen Ausführungen existieren und somit teils abweichende Eigenschaften besitzen. Um diesen Vorgang benutzerfreundlich und effizient zu gestalten und integrieren, sollte der gesamte Prozess der Überlagerung zudem so automatisch vollzogen werden, wie es die eingesetzten Konzepte maximal zulassen. Ob und in welchem Maße ein Überlagerungskonzept jedoch in welchen Phasen des Prozesses automatisch auszuführen ist, erfordert zum einen die initiale Elaboration potenzieller Konzepte und zum anderen einen Vergleich auf der Basis definierter Automatisierungsstufen.

Unterschiede bezüglich der Automatisierung sind nicht die alleinigen Einflussgrößen, wird die Wahl eines geeigneten Überlagerungskonzepts in Erwägung gezogen: Welche korrelierende Konzept- und Technologieeigenschaften durch eine Auswahl zusätzlich beeinflusst werden, muss im Vorfeld für die produktionsrelevanten Teilaspekte erarbeitet werden. Beispielsweise eignet sich ein Überlagerungsvorgang, der eine lange Vorbereitungszeit benötigt, dafür aber vollautomatisiert ausführt, nicht für einen zeitkritischen Produktionsprozess. Die beschriebene Vorgehensweise erklärt die Notwendigkeit des Erarbeitens geeigneter Überlagerungskonzepte und einen Vergleich der technischen Vertreter, was das Kernziel dieser Arbeit darstellt und im folgenden Abschnitt detailliert erläutert wird.

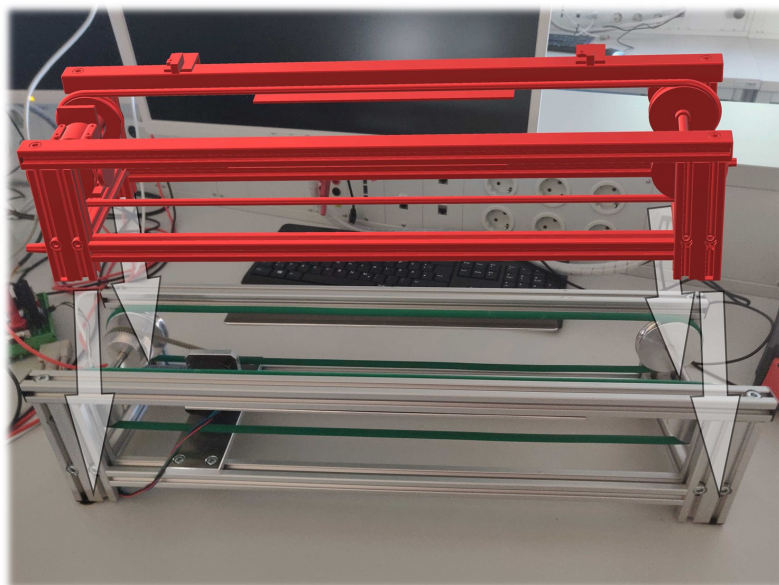


Abbildung 1.1: Abbildung einer virtuellen Komponente auf eine physische Entität

1.2 Zielsetzung

Ziel dieser Arbeit ist zum einen die initiale Ergründung geeigneter Überlagerungskonzepte im Bereich der erweiterten Realität, um digital vorliegende 3D-Modelle gemeinsam mit deren physischen Ursprung zu überlagern. Dabei ist zu beachten, dass die Konzepte ausreichend unterschiedliche Vorgehensweisen als Grundlage der Überlagerung beschreiben – ein Vergleich von Umsetzungen semantisch gleicher Konzepte allein bildet ein zu enges Spektrum ab und wurde bereits in verwandten Arbeiten erarbeitet. Zum anderen werden technische Realisierungen der Überlagerungskonzepte recherchiert, um, auf eine Implementierung dieser aufbauend, einen präziseren Vergleich der unterschiedlichen Ansätze zu ermöglichen und weitere Eigenschaften der Konzepte messbar zu gestalten.

Auf den recherchierten Befunden aufbauend wird eine Korrelation der Daten hergestellt, indem diese auf ihre Automatisierbarkeit geprüft und eingestuft werden. Als Metrik gelten dabei wissenschaftlich definierte Automatisierungsgrade in den unterschiedlichen Phasen des Überlagerungsvorgangs. Zusätzlich werden weitere messbare Eigenschaften der entsprechenden Überlagerungstechniken erfasst, um die Evaluation durch mögliche Korrelationen der technischen Eigenschaften und der Konzepte anzureichern. Zur Durchführung des Vergleichs der Konzepte werden alle notwendigen technischen Umsetzungen der Abbildungsmöglichkeiten selbst erstellt.

1.3 Aufbau der Arbeit

Die Arbeit wird in folgende Kapitel eingeteilt:

Kapitel 2 beinhaltet sämtliche Grundlagen, die für das Verständnis dieser Arbeit notwendig sind. Dazu gehören insbesondere die Erörterung existenter Automatisierungsgrade oder Erkenntnisse im Bereich der Objekterkennung.

In Kapitel 3 ist der Stand der Technik zusammengefasst, welcher unter anderem eine Auswahl von marktführenden Augmented Reality SDKs und Datenbrillen vorstellt.

Kapitel 4 analysiert die zu untersuchenden Überlagerungskonzepte und beschreibt das für den Versuch notwendige Vorgehen.

Der Analyse anschließend folgt in Kapitel 5 die Evaluation der Ergebnisse, die sich in die definierten Messgrößen unterteilt.

Abschließend beinhaltet Kapitel 6 eine Zusammenfassung der Arbeit und gibt einen Ausblick auf mögliche zukünftige Arbeiten.

2 Grundlagen

Dieses Kapitel beschreibt sämtliche konzeptionelle und technische Grundlagen, welche als Fundament für den Inhalt dieser Arbeit benötigt werden. Ziel dieses Kapitels ist zudem, benötigte Fachbegriffe für den weiteren Verlauf der Arbeit einzuführen. Zu Beginn werden die für den Vergleich notwendigen Automatisierungsgrade sowie deren Einteilung vorgestellt (vgl. Abschnitt 2.1). Daraufhin werden Definitionen, Konzepte und Technologien im Bereich der Mixed Reality erläutert (vgl. Abschnitt 2.2). Schließlich werden die grundlegenden Erkenntnisse aus der Forschung bei der Erkennung von Objekten (vgl. Abschnitt 2.3) zusammengefasst.

2.1 Automatisierungsgrade

Um eine Einteilung aller erarbeiteten Überlagerungskonzepte vorzunehmen, müssen diese voneinander abgrenzbar sein. Ein möglicher Ansatz ist die Klassifizierung der Konzepte nach dem Grad der Automatisierung. Einem Konzept, das größtenteils von Benutzereingaben abhängt, wird so eine niedrigere Wertigkeit zugeordnet, als einem anderen Konzept, welches in großen Teilen automatisiert durchgeführt werden kann.

Nach Sheridan und Verplank [SV78] lässt sich eine Automatisierung bei der Interaktion von Mensch und Maschine in zehn Stufen einordnen:

1. **Stufe 1:** Die Maschine bietet keine Unterstützung an. Sämtliche Aktionen werden von dem Benutzer durchgeführt.
2. **Stufe 2:** Die Maschine bietet dem Benutzer eine Gesamtmenge an Vorschlägen an, die der Benutzer als Grundlage für seine Interaktion wählen kann.
3. **Stufe 3:** Die angebotene Gesamtmenge wird von der Maschine zusätzlich auf wenige relevante Vorschläge gekürzt.
4. **Stufe 4:** Genau eine Möglichkeit der Interaktion wird von der Maschine vorgeschlagen.
5. **Stufe 5:** Die von der Maschine erkannte Interaktionsmöglichkeit wird ausgeführt, sobald der Benutzer dies bestätigt.
6. **Stufe 6:** Die von der Maschine erkannte Interaktionsmöglichkeit wird nach einer gewissen Ablaufzeit automatisch ausgeführt, falls der Benutzer nicht eingreift.
7. **Stufe 7:** Die von der Maschine erkannte Interaktionsmöglichkeit wird automatisch ausgeführt und der Benutzer über den Ablauf informiert.
8. **Stufe 8:** Die von der Maschine erkannte Interaktionsmöglichkeit wird automatisch ausgeführt und der Benutzer über den Ablauf informiert, falls dieser dazu eine Anfrage an die Maschine sendet.

9. **Stufe 9:** Die von der Maschine erkannte Interaktionsmöglichkeit wird automatisch ausgeführt. Die Maschine entscheidet selbstständig, ob der Benutzer über den Ablauf informiert wird.
10. **Stufe 10:** Die Maschine führt die erkannte Interaktionsmöglichkeit vollständig autonom aus und ignoriert den Benutzer.

Stufe eins bietet demnach den niedrigsten, Stufe zehn den höchsten Automatisierungsgrad. Zusätzlich bauen die Stufen vollständig aufeinander auf. Mit Hilfe der damit definierten Metrik lässt sich jedes Konzept hinsichtlich seiner Automatisierung einordnen. Um die Automatisierung genauer festzulegen, teilt Parasuraman et al. [PSW00] den Prozess der Mensch-Computer-Interaktion in vier Abschnitte ein, die sich jeweils bezüglich ihres Automatisierungsgrades untersuchen lassen.

- **Informationsaufnahme:** Die Automatisierung im Bereich der Aufnahme von Informationen wird auf die Erfassung und das Empfangen von Eingabedaten angewandt. Diese Aufnahme ist, auf der Hardwareebene, durch Sensoren automatisierbar. Auf Seiten des Benutzers steht dem die Perzeption gegenüber.
- **Informationsverarbeitung:** Die Informationsverarbeitung des Menschen ist ein kognitiver Prozess unter Einbeziehung des Arbeitsgedächtnisses zur Verarbeitung sensorisch aufgenommener Informationen. Dieser Abschnitt lässt sich unter dem Einsatz von Algorithmen automatisieren, sodass, auf Grundlage von Eingabedaten, Schätzungen und Einordnungen maschinell durchführbar sind.
- **Entscheidungsfindung:** Die Auswahl einer Interaktion und der damit einhergehende Findungsprozess ist in unterschiedlichem Maße automatisierbar. So können Entscheidungen ausschließlich auf Grundlage des Benutzers entschieden werden, oder vollkommen automatisch, unter Beihilfe von Entscheidungsbäumen, maschinellem Lernen und weiteren Computertechniken.
- **Ausführung der Aktion:** Die eigentliche Ausführung der Aktion bildet den letzten Abschnitt der Automatisierungsphasen und beschreibt, ob die Interaktion von dem Benutzer oder der Maschine ausgeführt wird.

Mit den hier vorgestellten Automatisierungsgraden und -phasen ist es möglich, einen Prozess der Überlagerung präziser zu unterteilen und für jede Phase einen Automatisierungsgrad festzulegen. Dazu müssen die vorgestellten Definitionen jedoch so angepasst werden, dass sie im Kontext der erweiterten Realität, speziell des Überlagerungsvorgangs von virtuellen Objekten, angewendet werden können.

2.2 Mixed Reality

Der folgende Abschnitt beschreibt den Begriff und die Prinzipien der Mixed Reality nach Milgram und Kishino [MK94].

Als Mixed Reality wird ein Teilgebiet der virtuellen Realität bezeichnet, welches die Darstellung einer real existierenden Umgebung gemeinsam mit einer virtuellen Umgebung beschreibt. Um eine Unterscheidung des Verhältnisses von virtuell und real dargestellten Elementen innerhalb einer Ansicht zu ermöglichen, erfolgt eine Einteilung in das Mixed Reality-Spektrum (nach Milgram und Kishino [MK94] „*reality–virtuality continuum*“).

Die erweiterte Realität (engl. „Augmented Reality“) kennzeichnet eine Ansicht, in der überwiegend Abbildungen einer realen Umgebung existieren. Dahingegen wird eine Ansicht mit mehrheitlichen Elementen aus einer virtuellen Umgebung als erweiterte Virtual Reality (engl. „Augmented Virtuality“) bezeichnet. Abbildung 2.1 zeigt diese Einteilung in das Mixed Reality-Spektrum.

Auf dieser Definition aufbauend wird mit dem Begriff der Mixed Reality die Interaktionsmöglichkeit des Betrachters mit Abbildungen aus der realen und der virtuellen Umgebung sowie die gegenseitige Beeinflussung dieser verbunden: Ein virtuell abgebildeter Quader kann von einem Benutzer beispielsweise aufgenommen und daraufhin auf eine Tischplatte, die in der realen Umgebung existiert, abgelegt werden. Der Benutzer bedingt somit die Translation der virtuellen Abbildung, während sich die virtuelle Umgebung, die der Computer kontinuierlich errechnet, gleichzeitig an die reale Umgebung anpasst, um die Kollision mit der Tischplatte und dem Quader zu errechnen. Diese gegenseitigen Einwirkungen von Benutzer, Computer und Umgebung sind in Abbildung 2.2 zu sehen.

Um Objekte in einem dreidimensionalen Raum zuverlässig und performant zu erkennen müssen mehreren Faktoren erfüllt sein. Kruijff et al. [KSF10] beschreiben fünf Überpunkte, welche mehrere Problembereiche enthalten, die eine erfolgreiche Objekterkennung, und somit eine korrekte Überlagerung negativ beeinflussen:

1. **Umgebung:** Die Beschaffenheit und Struktur der Umgebung mindert die Erkennung von Objekten maßgeblich; so erschweren ungünstige Lichtverhältnisse und wiederkehrende Muster oder Farben, dass ein zu erkennendes Objekt von der Umgebung extrahiert werden kann.
2. **Sensoren und Bildqualität:** Die Bildauflösung, Farbkorrekturen sowie Kontrasteinstellungen können sich negativ auf die Erkennung eines Objekts auswirken. Zudem beeinflusst die Linse und die Bildfrequenz der Kamera die Qualität der Erkennungsalgorithmen.
3. **Augmentation:** Die korrekte Einblendung virtueller Objekte über die reale Umgebung und die kontinuierliche Neuanpassung der Position durch Algorithmen wird durch mehrere Faktoren erschwert. Beispielsweise reicht die Genauigkeit der Sensoren moderner Datenbrillen häufig nicht aus, um eine reale Entität akkurat zu überlagern. Erschwert wird dies zudem, wenn die Entität durch die Umgebung oder die Bewegungen des Benutzers verdeckt wird und die Position von Algorithmen geschätzt werden muss. Abhängig von der Umgebung haben Bilderkennungsalgorithmen Schwierigkeiten Objekte räumlich abzutrennen: Sind die Kanten einer Entität im Vordergrund nicht deutlich zu messen, kann dieses im virtuellen Raum nicht zuverlässig eingefügt werden.

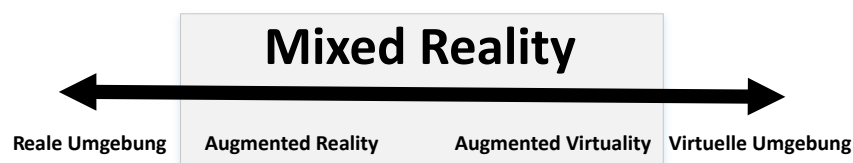


Abbildung 2.1: Mixed Reality-Spektrum nach Milgram und Kishino [MK94]

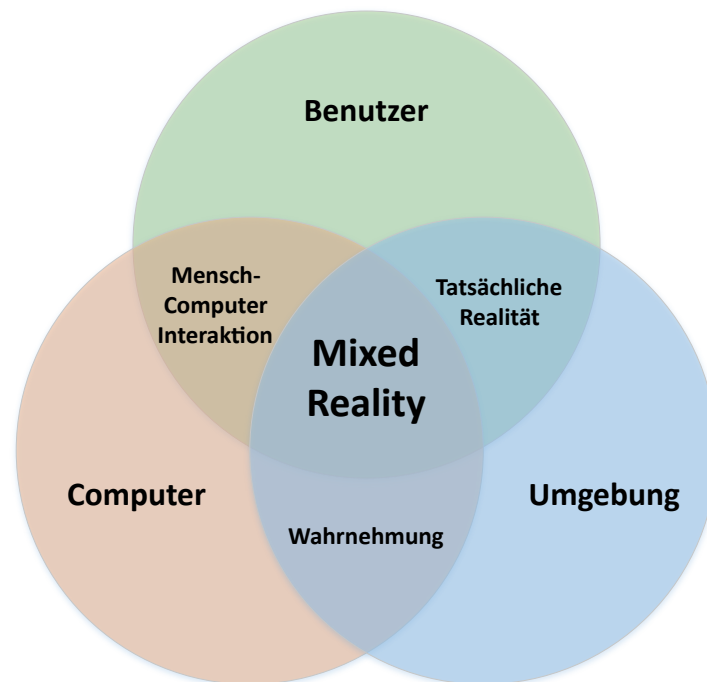


Abbildung 2.2: Mixed Reality-Einflussfaktoren nach Bray et al. [BZS18]

4. **AR-Geräte:** Die Eigenschaften des Displays von Datenbrillen und weiteren AR-Geräten bestimmen eine korrekte Anzeige der virtuellen Objekte in der realen Umgebung. So beeinflussen die stereoskopische Umrechnung, das Sichtfeld sowie die Latenz des Gerätes die Anzeige der errechneten virtuellen Realität.
5. **Benutzer** Die unterschiedlichen Sinneswahrnehmungen individueller Benutzer erschweren es, ein universelles Design zur Anzeige einer zusätzlichen virtuellen Ebene zu entwerfen. Den menschlichen Sinn betreffende Eigenschaften wie die Akkommodation des Auges oder die Tiefenwahrnehmung unterscheiden dabei unterschiedliche Benutzer der eingesetzten Technik.

Zur Bewältigung oder Eindämmung ein oder mehrerer Problemgebiete existieren unterschiedliche Konzepte und Technologien, die sich unter anderem in den den Umsetzungen der Datenbrillen wiederfinden.

Aus technischer Sicht wird zwischen zwei Gerätetypen unterschieden, die den Bereich der erweiterten Realität bzw. der Erweiterten Virtual Reality abzudecken versuchen. Immersive Brillen, wie das Acer Mixed Reality Headset¹, schirmen die Sicht auf die reale Umgebung vollständig ab, erlauben aber dennoch eine Interaktion mit dieser durch extra dafür angebrachte Sensoren: So können beispielsweise Videokameras die Umgebung erkennen und die Position des Benutzers im Raum feststellen, sowie dessen Kopfbewegungen in die virtuelle Umgebung übersetzen. Zusätzlich kann mittels Bluetooth eine Verbindung zu tragbaren Controllern aufgebaut werden, die von den Sensoren

¹<https://www.acer.com/ac/de/DE/content/series/wmr>

des Headsets daraufhin im Raum erkannt werden und deren Translation in die virtuelle Umgebung übertragen wird. Somit werden sie in den Bereich der Erweiterten Virtual Reality eingeteilt [BZS18].

Holographische Geräte hingegen erlauben die freie Sicht auf die reale Umgebung. Ein vor den Augen angebrachtes Display projiziert dabei virtuelle Elemente in den Raum, der von den angebrachten Sensoren kontinuierlich ausgemessen wird. Diese Geräte sind Vertreter der Erweiterten Realität [BZS18]. Ein Beispiel für ein holographisches Gerät ist die Microsoft HoloLens² (vgl. Abschnitt 3.2).

Mixed Reality stellt das übergeordnete Anwendungs- und Forschungsfeld dieser Arbeit dar. Die verwendeten Technologien und Konzepte sind somit in dieses Themenfeld eingebettet. Einen weiterführenden Einblick in die Teilgebiete des Forschungsfelds der Mixed Reality gibt das „International Symposium on Mixed and Augmented Reality“³ (kurz ISMAR), eine akademische Konferenz im Bereich der Mixed Reality. Die Trends der ISMAR-Veröffentlichungen im Bereich Mixed Reality der Jahre 2008 bis 2018 führen die Schlagwörter *Tracking-Techniken* und *Rekonstruktion* an [KBB+18]. Das Schlagwort *Tracking-Techniken* bezeichnet dabei Methoden zum Erkennen einer physischen Entität mit Hilfe von Sensoren und die Berechnung der Position des Sichtfelds. Der Begriff *Rekonstruktion* bündelt Methoden zur automatischen Generierung von virtuellen Entitäten, basierend auf Bildmaterial oder anderen Datenquellen, welche aus der physischen Umgebung gewonnen werden. Diesen beiden Teilgebieten lässt sich der Inhalt dieser Arbeit zuweisen.

2.3 Objekterkennung

Bereits 1997 beschrieb Azuma [Azu97] in seiner Veröffentlichung „A survey of augmented reality“ die fundamentale Herausforderung, virtuelle und physische Entitäten korrekt zu überlappen. Er verdeutlichte die Notwendigkeit der Bewältigung dieses Problems, indem er ein Beispiel aus der Medizin herausgriff: Operiert ein Chirurg mit Hilfe von Mixed Reality-Technologie, muss die Überlagerung zu jedem Zeitpunkt korrekt dargestellt werden – andererseits ist ein Einsatz der Technologie nicht vorstellbar.

Die Kategorie Tracking ist unter den ISMAR-Veröffentlichungen im Bereich der Mixed Reality von 2008 bis heute mit 16,2% die einflussreichste bezüglich der Anzahl an Veröffentlichungen [KBB+18]. Der Fokus liegt dabei auf Themengebieten wie der Lokalisation des Benutzers, dem Mapping der Umgebung, der virtuellen Rekonstruktion von Entitäten sowie Hybriden Tracking-Ansätzen [KBB+18].

Als Tracking-System wird ein Gerät bezeichnet, das Position und Orientierung einer Entität bestimmen und interpretieren kann [CKF09]. Costanza et al. [CKF09] unterteilt dabei in aktive und passive Tracking-Systeme: Als passiv wird ein System bezeichnet, das Objekte allein durch Sensoren erkennen und unterscheiden kann – eine spezielle Hardware muss nicht an den Objekten installiert werden. Aktive Tracking-Systeme kommunizieren hingegen mit Sensoren, die an den zu erkennenden Objekten angebracht sind.

²<https://www.microsoft.com/hololens>

³<http://www.ismar.net/>

Eine Herausforderung bei der Entwicklung von Tracking-System ist dabei, alle sechs Freiheitsgrade, welche durch eine Kopfbewegung ermöglicht werden, in nahezu Echtzeit auf das virtuelle Abbild zu übertragen: drei Translationen sowie drei Rotationen (links bzw. rechts, vor bzw. zurück, hoch bzw. runter).

Trotz des vorherrschenden Interesses, existieren in der Objekterkennung Nutzungsgrenzen, die Raum für Forschungs- und Industriebestrebungen schaffen. Zhou et al. [ZDB08] beschreiben in ihrer Veröffentlichung „Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR“ eine Zusammenfassung von Forschungsarbeiten im Bereich der Augmented Reality, in welcher sie auf folgende Grenzen vorherrschender Tracking-Techniken hinweisen:

1. Mit steigender Komplexität der Umgebung, in der das Objekt erkannt werden soll, sinkt die Erkennungsrate. So stellen sich bewegende Objekte und wechselnde Lichtverhältnisse eine große Herausforderung dar.
2. Das Tracking außerhalb von Gebäuden ist schwer zu bewerkstelligen, da eine Festlegung von Merkmalspunkten, welche sich präzise unterscheiden lassen, durch die natürliche Begebenheit häufig erschwert ist.
3. Hybride Tracking-Systeme sind aufgrund verschiedener Merkmale wie Größe, Gewicht und Energiebedarf zumeist nicht effizient einsetzbar.

Den genannten Problempunkten lässt sich bislang technologisch und konzeptionell schwer entgegenwirken. Aus diesem Grund müssen sie bei einem Einsatz oder Entwurf von Tracking-Systemen stets einbezogen werden.

In diesem Abschnitt werden vier grundsätzliche Techniken vorgestellt, die eine Objekterkennung durch unterschiedliche Methoden und Konzepte vollziehen. Die GPS-gestützte Objekterkennung (Abschnitt 2.3.1) bietet eine global einsetzbare Technik an, die jedoch ungenaue Ergebnisse erzielt. Eine Erkennung mittels Referenzmarkern (Abschnitt 2.3.2) bietet eine genauere Erkennung, jedoch muss die Umgebung im Vorfeld manipuliert werden. Systeme, die beliebige Muster aus der Umgebung erkennen können sind deutlich rechenintensiver, jedoch bieten sie ein hohes Maß an Flexibilität. Dies wird als *Natural Feature Tracking*, kurz NFT bezeichnet (Abschnitt 2.3.3). Schließlich besteht die Möglichkeit, Objekte durch elektromagnetische Wellen in der Umgebung zu Orten (Abschnitt 2.3.4).

2.3.1 Global Positioning System

Das *Global Positioning System* (GPS) basiert auf Satellitenkommunikation und wird häufig innerhalb von Navigationssoftware eingesetzt. Mit Hilfe von Koordinaten lässt sich die Position eines GPS-Sensors bestimmen. Ist eine Datenbrille sowie eine zu überlagernde Entität mit einem solchen Sensor ausgestattet, lässt sich deren Abstand zueinander messen und die Entität kann relativ zu der Position der Datenbrille eingeblendet werden.

Ein großer Nachteil dieser Technik ist, dass GPS-Signale ungenaue Ergebnisse liefern, wenn sich Objekte zwischen Satellit und Empfänger befinden [CKF09]. Gerade in Gebäuden weichen die Werte somit oft stark von den tatsächlichen Positionen ab [CKF09]. Da eine Überlagerung stets auf oder sehr nah der physischen Entität eingeblendet werden sollte, eignet sich der Einsatz des GPS-Konzepts für diese Arbeit nicht.

2.3.2 Planare Referenzmarker

Referenzmarker sind präparierte Markierungen auf einem Objekt oder in seiner nahen Umgebung mit dem Ziel, ein virtuelles Bild akkurat im Sichtfeld des Benutzers zu platzieren. Dabei ist entscheidend, dass sich der Referenzmarker ausreichend von seiner Umgebung abhebt und aus unterschiedlichen Betrachtungswinkeln erkennbar bleibt. Etablierte Markersysteme, wie QR-Codes⁴, sind aus diesem Grund für eine Nutzung in der Mixed Reality ungeeignet [Fia05].

Folgender Ausschnitt an Metriken kann angewendet werden, um eine Markerererkennung anhand ihrer Effektivität zu messen [Fia05]:

- **Falsch-positive Fehlerrate:** Beschreibt die Rate, mit der Marker erkannt werden, obwohl diese nicht in der realen Umgebung platziert worden sind.
- **Falsch-negative Fehlerrate:** Beschreibt die Rate, mit der Marker nicht erkannt werden, obwohl sie in der realen Umgebung platziert wurden.
- **Verwechslungsrate unter existierenden Markern:** Beschreibt die Rate, mit der platzierte Marker untereinander verwechselt werden.

Ein Beispiel eines Mixed Reality Referenzmarkersystems ist ARTag (siehe Abbildung 2.3 erste Reihe, dritte Spalte), welches von Fiala [Fia05] veröffentlicht wurde. ARTag setzt ein 10 x 10 großes Feld an Quadraten ein, wobei jedes Feld die Farbe schwarz oder weiß annehmen kann. Der Rand des Markers ist einfarbig und 2 Felder breit.

Der Algorithmus zur Erkennung des Referenzmarkers scannt das Sichtfeld des Benutzers zunächst nach dem Rand des Markers ab. Wird dadurch ein (oder mehrere) potentieller Marker gefunden, wird ihr Innenbereich in vier Matrizen digital verarbeitet – eine Matrix für jede mögliche Rotation des Referenzmarkers. Um zu verhindern, dass bei diesem Verarbeitungsschritt Duplikate entstehen können, sind die möglichen Kombinationen, aus denen der Benutzer einen Marker erstellen kann vorab gefiltert. Da für jedes Feld zwei Farben möglich sind wurden mit Hilfe der Hamming-Distanz [Ham50] weitere Kombinationen gefiltert, die zu einer höheren Verwechslungsrate beitragen würden. ARTag erlaubt damit insgesamt 2002 mögliche Referenzmarker zu erstellen, die eine Verwechslungsrate untereinander, sowie mit ihrer Umgebung von unter 0,0039% ermöglichen, was in etwa einem von 26 000 Fällen entspricht. Kann der Referenzmarker korrekt erkannt werden, liefert der Algorithmus die ihm zugewiesene Identifikationsnummer zurück und das verknüpfte 3D-Modell kann auf dem Marker eingeblendet werden. Zwar benötigt das Tracking durch Referenzmarker eine Manipulation der Umgebung durch das Hinzufügen der Marker selbst, dafür beschränkt sich die Technik nicht auf den exklusiven Gebrauch in Gebäuden oder außerhalb [CKF09].

Garrido-Jurado et al. [GMMM14] entwickelte im Jahr 2014 ein Referenzmarkersystem mit dem Namen ArUco, das im Folgenden als eine Alternative zu ARTag vorgestellt wird. Die Entwickler basieren ihre Arbeit dabei auf zwei Grundproblematiken bei der Erkennung von Markern im Raum:

⁴<https://www.qrcode.com/en/>

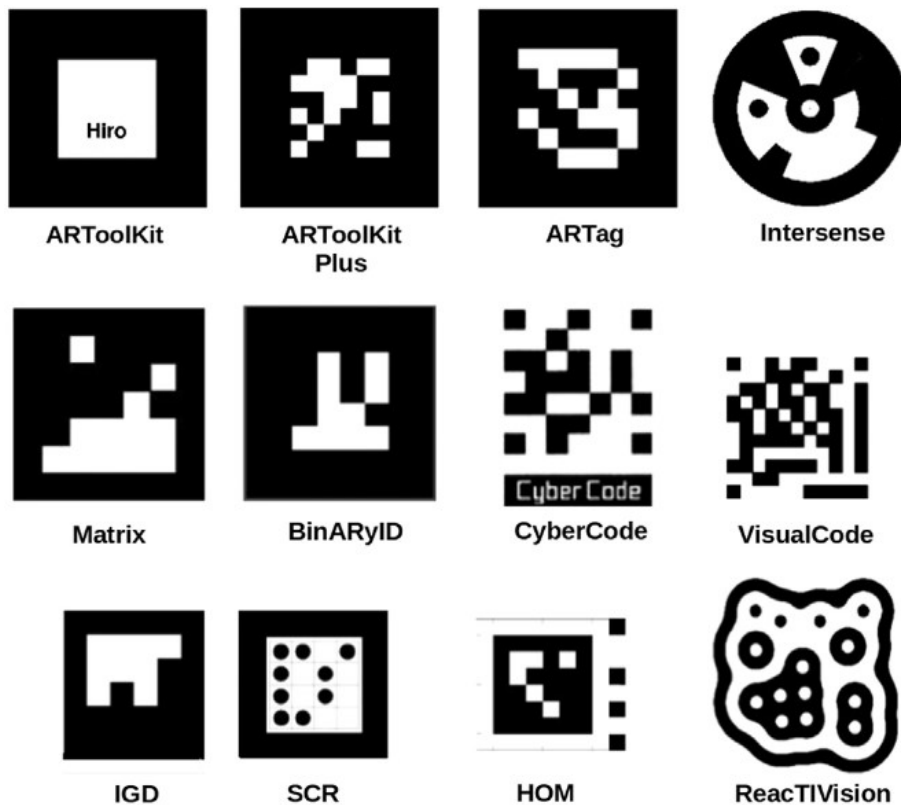


Abbildung 2.3: Auswahl planarer Markersysteme nach Fiala [Fia05]

1. Die Anzahl der benötigten Marker kann das erlaubte Limit (vgl. ARTag mit maximal 2002 Möglichkeiten) übersteigen. Sollte das Limit jedoch unterschritten werden, sollen die eingesetzten Marker maximal unterscheidbar sein, um eine optimale Erkennung zu ermöglichen.
2. Die virtuelle Überlagerung liegt räumlich stets vor den realen Umgebung; Tätigt der Benutzer beispielsweise eine Handbewegung über einen Marker hinweg, sollte das virtuelle Bild hinter der Hand positioniert werden.

Es existieren mehrere Faktoren, die bei dem Entwurf eines Markersystems berücksichtigen werden müssen. Zum einen müssen die falsch-positiv sowie die falsch-negativ Raten bei der Erkennung niedrig gehalten werden und zum anderen sollen die Marker untereinander verwechslungsfrei erkannt werden und von der Umgebung unterscheidbar gestaltet sein. Unter Berücksichtigung dieser Voraussetzung soll das System eine möglichst große Menge an Markern bereitstellen können.

Garrido-Jurado et al. [GMMM14] benutzen eine automatische Methode zur Erstellung der vorgesehenen Anzahl an Markern. Der Algorithmus beginnt dabei mit einer leeren Menge und füllt diese inkrementell mit neuen Markern auf, welche über einen stochastischen Prozess erstellt werden. Dabei werden die quadratischen Anordnungen, die das Muster des Markers bilden, so gewählt, dass deren Hamming-Distanz [Ham50] zu den bestehenden Markern einen gewissen Grenzwert übersteigt. Falls dieser Grenzwert für einen Marker nicht erreicht wird, wird er verworfen und

ein neuer Marker wird generiert. Da mit wachsender Anzahl der Marker der Grenzwert zu häufig unterschritten wird, senkt der Algorithmus diesen nach zu vielen verworfenen Kombinationen. Der Ablauf ist in Algorithmus 2.1 zu sehen.

Algorithmus 2.1 ArUco Algorithmus zur Auswahl von Markern [GMMM14]

```

 $\mathcal{D} \leftarrow \emptyset$  # Reset dictionary
 $\tau \leftarrow \tau^0$  # Initialize target distance
 $\varrho \leftarrow 0$  # Reset unproductive iteration counter
while  $\mathcal{D}$  has not desired size do
  Generate a new marker  $m$ 
  if distance of  $m$  to elements in  $\mathcal{D}$  is  $\geq \tau$  then
     $\mathcal{D} \leftarrow \mathcal{D} \cup m$  # Add to dictionary
     $\varrho \leftarrow 0$ 
  else
     $\varrho \leftarrow \varrho + 1$  # It was unproductive
    # Maximum unproductive iteration reached?
    if  $\varrho = \psi$  then
       $\tau \leftarrow \tau - 1$  # Decrease target distance
       $\varrho \leftarrow 0$ 
    end if
  end if
end while

```

Zur Erkennung der Marker wird das Bild zunächst in Graustufen umgewandelt. Die prägnanten Konturen des Bildes werden mit Hilfe eines Schwellenwertverfahrens extrahiert. Auf die Erkennung von Ecken wurde an dieser Stelle verzichtet, da die Umsetzung dieser Technik nicht in Echtzeit zu realisieren ist. Zur genaueren Ermittlung der Markerkonturen wird der Suzuki und Abe [Suz+85], als auch des Douglas-Peucker [DP73] Algorithmus eingesetzt, wodurch lediglich relevante Konturen der Marker zur Berechnung weiterverarbeitet werden. Die Innenfläche der Konturen wird schließlich perspektivisch angepasst und ausgewertet, indem die schwarzen und weißen Quadrate in eine Matrix transformiert werden. Unter Berücksichtigung möglicher Rotationen der erkannten Marker wird nun geprüft, ob es sich um einen Marker aus der definierten Menge \mathcal{D} handelt. Eine Verbesserung der Performanz wird an dieser Stelle erreicht, indem die Menge als Binärbaum aufgespannt wird. Schließlich wird die Ausrichtung des Markers bezüglich der Kamera errechnet, um eine Überlagerung korrekt im Sichtfeld zu platzieren.

Damit, trotz einer teilweisen Verdeckung, möglichst zuverlässig bestimmt werden kann, um welchen Marker es sich handelt, wird ein Marker-Brett eingesetzt. Dieses setzt sich aus einem schachbrettartigen Muster unterschiedlicher Farben, kombiniert mit ArUco-Markern in den einzelnen Quadranten des Musters zusammen. Der Aufbau des Bretts ist in Abbildung 3.1 zu sehen. Durch die Verteilung der Marker, sowie der Zuweisung von Farben pro Quadrat kann eine Verdeckungsmaske errechnet werden, die es erlaubt Teile der virtuellen Überlagerung entsprechend der Verdeckung auszublenden.

Ein Ansatz von Nakazato et al. [NKY05] erlaubt die Erkennung von für das menschliche Auge unsichtbaren Referenzmarkern unter dem Einsatz einer Infrarotkamera und mehreren Infrarot-LEDs, die synchron an- und ausgeschaltet werden. Der Marker unterteilt sich in mehrere quadratische

und durchsichtige Retroreflektoren, welche das Signal der Infrarot-LEDs reflektieren und so eine Erkennung mit Hilfe der Infrarotkamera erlauben. Die Genauigkeit der Schätzung von Kameraposition und -ausrichtung ist hierbei mit den Messergebnissen visueller Referenzmarkersysteme vergleichbar.

Das Konzept der Marker-basierten Überlagerung ist in den Mixed Reality Anwendungsgebieten etabliert und reich an technischen Vertretern (vgl. Kapitel 3). Es eignet sich somit für einen Vergleich mit weiteren Überlagerungskonzepten und wird in den folgenden Kapiteln im Kontext dieser Arbeit behandelt.

2.3.3 NFT

Techniken wie der Einsatz von Referenzmarkern setzen voraus, dass die zu überlagernde Umgebung angepasst wird und die Markertexturen bekannt sind [NY99]. Damit die virtuelle Entität möglichst genau auf deren physisches Abbild projiziert wird, muss der Marker also statisch und exakt positioniert werden.

NFT steht für *Natural Feature Tracking* und beschreibt Computer Vision-Algorithmen, die in nahezu Echtzeit aus Sensorbildern Muster extrahieren, wobei die Position des erkannten Musters vorab nicht bekannt ist. Shi und Tomasi [ST93] beschreiben als dessen Grundlage die Korrelation von aufeinander folgenden Frames mit Hilfe von Gleichung (2.1).

$$I(x, y, t + \tau) = I(x - \xi(x, y, t, \tau), y - \eta(x, y, t, \tau)). \quad (2.1)$$

Die Funktion $I(x, y, t)$ steht dabei für eine Bildersequenz, wobei die Variablen x und y die Dimensionen darstellen und t die Variable der Zeit. Ein später in der Zeit ($t + \tau$) vorhandenes Bild steht mit den vorhergehenden in Relation, wenn jeder Bildpunkt um einen gewissen Betrag bewegt wird. $\delta = (\xi, \eta)$ bezeichnet dabei die Bewegungsgeschwindigkeit eines Punktes $z = (x, y)$ im Zeitraum t und $t + \tau$ und beschreibt somit seine Verschiebung. Verdeckungen und Reflektionen sind in dieser Gleichung nicht berücksichtigt und verfälschen das Ergebnis. Ebenso ist das Nachverfolgen einzelner Bildpunkte häufig erschwert, da sie ihren Nachbarpunkten häufig ähneln, weswegen Shi und Tomasi [ST93] die Berechnung von kleinen Bildbereichen vorschlagen.

Im Folgenden wird ein NFT-Ansatz von Neumann und You [NY99] vorgestellt, der sowohl 2D-Bildbereiche als auch Bildpunkte aus dem Benutzersichtfeld auswertet, um das Tracking eines Objektes zu ermöglichen. Bildbereiche als Muster sind prinzipiell leichter zu errechnen, deren Erkennung nimmt allerdings ab, falls der Blickwinkel sich ändert, Teile des Bereichs verdeckt sind oder die Beleuchtung die Bildqualität zu stark beeinflusst. Eine Berechnung auf Grundlage von Bildpunkten hat den Vorteil, dass Bewegungsmessungen pixelgenau durchgeführt werden können. Mit einer Menge an korrekt erkannten Bildpunkten ist es somit möglich, eine Odometrie der Datenbrille durchzuführen. Durch diese Genauigkeit bedingt, nimmt die Erkennungsrate in komplexen Umgebungen unter ausgeprägten Kopfbewegungen allerdings deutlich ab.

Die Tracking-Technik nach Neumann und You [NY99] ist dabei in mehrere Phasen unterteilt. Zunächst wird der optische Fluss [BB95] eines Bilds errechnet und geeignete Bildpunkte als Merkmale extrahiert. Dieser Vorgang gestaltet sich dynamisch: Fallen erkannte Merkmale unter einen gewissen Grenzwert, werden sie aus der Menge entnommen; Neue Bildpunkte werden dabei weiterhin hinzugefügt. Danach werden nicht überlappende Bildbereiche festgelegt, die eine

ausreichende Anzahl an Bildpunkten, welche bereits als Merkmale erkannt wurden, beinhalten. Je mehr Merkmale dabei im Verhältnis zu der Gesamtzahl an Pixeln in dem Bildbereich existieren, desto höher ist die Qualität des Bereichs. Um Bildfehler wie Verzerrungen zu erkennen, ist der angewendete Algorithmus so angepasst, dass Iterationen mit niedriger Erkennungsrate gefiltert werden. Dies geschieht unter der Annahme, dass sich die erkannten Punkte und Regionen zueinander planar bewegen. In einer Umgebung, in der wenige oder keine planaren Flächen existieren, ist die angewandte Technik somit weniger wirkungsvoll.

Ebenso werden Ergebnisse aus vorherigen Messungen in neue Messungen einbezogen, was eine Optimierung bei der Erkennung von Merkmalen bewirkt. Das Ergebnis einer Erkennung von regionalen und Punktmerkmalen ist in Abbildung 2.4 zu sehen.

Lepetit und Fua [LF06] erwirkten eine Performanzsteigerung des grundsätzlichen NFT-Ansatzes, indem sie rechenintensive Aufgaben vorab in einer Lernphase unter dem Einsatz von *Randomized Trees* errechnen. Um Punkte, die sich als Merkmale eignen, zu extrahieren, wird auf jeden Pixel m ein diskretisierter Kreis mit dem Mittelpunkt m gelegt. Danach werden alle Pixel aus der Menge an möglichen Merkmalen entfernt, welche einen ähnlichen Grauton besitzen, wie ihr diametral entgegengesetzter Pixel. Die verbliebenen Pixel werden über einen Laplace-Operator gefiltert, wobei die Pixel verbleiben, dessen Grauton maximal unterschiedlich zu dem Grauton des Mittelpunkts ist. Somit ist es möglich, sowohl Geschwindigkeit als auch Verlässlichkeit des Algorithmus zur Laufzeit zu steigern [LF04]. Die so berechneten Merkmale werde als Trainingsdaten für die Lernphase verwendet.



Abbildung 2.4: NFT-Algorithmus nach Neumann und You [NY99] angewandt auf einen einzelnen Frame [NY99]

Ein leistungsstarker Algorithmus zur Erkennung von Ecken in Bildern trägt den Namen FAST („*Features from accelerated segment test*“) und wurde von Rosten und Drummond [RD06] im Jahre 2006 veröffentlicht.

Die Idee des Algorithmus ist dabei, einen Kreis von sechzehn Pixeln um einen ausgewählten Punkt auszuwählen und deren Kontrast zu untersuchen. Falls zwölf Pixel in Folge heller bzw. dunkler als der Referenzpunkt sind, wird dieser Punkt als Ecke zugeordnet. Um die Genauigkeit zu erhöhen, kann ein Grenzwert t bei dem Vergleich des Kontrasts definiert werden, der aussagt, bis zu welcher Differenz ein Punkt als heller oder dunkler gezählt wird. Die Folge von genau zwölf Pixeln ist deswegen gewählt, da ein performanter Test effizient Punkte herausfiltern kann, die im Bild keine Ecken darstellen. Dabei werden lediglich die 4 Pixel der Kompassrichtungen (1, 5, 9 und 13) untersucht: Sind mindestens drei der Pixel heller oder dunkler als der Kreismittelpunkt, handelt es sich um eine Ecke. Somit kann im optimalen Fall bereits nach zwei Prüfungen errechnet werden, dass es sich bei diesem Punkt um keine Ecke handelt. Um den Algorithmus für weniger als 12 Pixel in Folge effizient anwenden zu können und ihn unabhängig von der Wahl und Anordnung der Pixel des Kreises zu gestalten, wird maschinelles Lernen eingesetzt, was sich in zwei Phasen einteilt.

In der ersten Phase wird der Algorithmus mit n Pixeln in Folge (statt den bisherigen 12) auf domänenspezifische Trainingsbilder angewandt. Dabei werden alle 16 Pixel des Kreises getestet und für jede Stelle $x \in \{1..16\}$ kann der Pixel an dieser Stelle ($p \rightarrow x$) einen von drei Zuständen annehmen:

$$S_{p \rightarrow x} = \begin{cases} d, I_{p \rightarrow x} \leq I_p - t & (\text{dunkler}) \\ s, I_p - t < I_{p \rightarrow x} < I_p + t & (\text{gleich}) \\ b, I_p + t \leq I_{p \rightarrow x} & (\text{heller}) \end{cases} \quad (2.2)$$

Unter der Auswahl einer Position x wird nun $S_{p \rightarrow x}$ für alle $p \in P$ berechnet, wobei P die Menge aller Pixel in den Trainingsbildern darstellt. Die Phase endet somit mit einer Einteilung in drei Teilmengen P_d , P_s und P_b .

Die zweite Phase wählt nun mit Hilfe eines Algorithmus [Qui86] ein x , das, gemessen an der Entropie einer booleschen Variable K_p , welche aussagt, ob p eine Ecke darstellt oder nicht, den größten Informationsgehalt liefert. Dieser Vorgang wird rekursiv auf allen drei Teilmengen wiederholt, bis der Informationsgehalt einer Teilmenge null beträgt – womit alle p in dieser Teilmenge den gleichen Wert für K_p annehmen. Mit Hilfe der Berechnung kann so ein Entscheidungsbaum aufgespannt werden, welcher wiederum als eine Reihe verschachtelter if-Abfragen in Programmiercode umzuwandeln ist und als Erkennung für Merkmalspunkte in Bildern angewendet werden kann.

Unter Einsatz dieses Konzepts ist der FAST-Algorithmus in der Lage sehr schnell Erkennungspunkte aus Bildern zu extrahieren. Ein Nachteil ist dabei jedoch, dass die Verlässlichkeit der Erkennung unter starkem Bildrauschen rapide abnimmt.

Der 2004 entworfene Computer Vision-Algorithmus SIFT („*Scale Invariant Feature Transform*“) wurde von Lowe [Low04] entworfen und wird im Folgenden vorgestellt. Der Algorithmus teilt sich in fünf Phasen ein.

In der ersten Phase werden, unter dem Einsatz der Gauß-Funktion, Extrempunkte als Merkmale aus den Bildern extrahiert. Zunächst wird das Bild in unterschiedlichen Weichzeichnungen generiert. Daraufhin wird das Bild in seiner Größe halbiert und die Weichzeichnung erneut angewendet. Um herausstechende Bildmerkmale herauszufiltern, bieten sich Ecken innerhalb des Bildes an. Diese ließen sich aus den Bildinformationen mit Hilfe des Laplace-Operators extrahieren, welcher jedoch

rechenintensiver ist als der Ansatz, den Lowe [Low04] vorschlägt: Statt die erste und die zweite Ableitung zu errechnen, um Extrempunkte zu erhalten, wird die Differenz von drei aufeinander folgenden, weichgezeichneten Bildern pro Bildpunkt im Vergleich zu den benachbarten Punkten auf den verglichenen Bildern errechnet. Der Vorgang ist in Abbildung 2.5 zu sehen. Der Bildpunkt wird als Merkmal gewählt, falls er im Vergleich zu den 26 weiteren Bildpunkten maximal hoch oder tief ist. Ist diese Berechnung abgeschlossen, wird der Algorithmus für den nächsten Satz an Bildern der halben Größe wiederholt. Das Ergebnis ähnelt dabei dem des Laplace-Operators angewendet auf diese Bilder, ist aber deutlich ressourcensparender zu berechnen. Da der Algorithmus jedoch keine zuverlässigen Ergebnisse präsentiert, falls die Extremwerte zu nah beieinander liegen, werden die Bilder zusätzlich in Rotation und Skalierung randomisiert. Ebenso wird 1% Bildrauschen hinzugefügt. Die so extrahierten Extrempunkte werden mit den bereits vorhandenen Punkten abgeglichen.

Die zweite Phase stellt die gefundenen Merkmale in Kontext und entfernt Ergebnisse die einen niedrigen Kontrast aufweisen oder nachteilig entlang von Kanten positioniert sind. Dazu wird zunächst ein Ansatz von Brown und Lowe [BL02] verwendet, der mit Hilfe der Taylor-Reihe ein Gleichungssystem erzeugt, das sich mit niedrigem Aufwand lösen lässt. Punkte, die nach dieser Auswertung unter einen festgelegten Grenzwert fallen, weisen einen zu niedrigen Kontrast auf und werden verworfen. Ebenso wird die, unter dem Einsatz der Gauß-Funktion typische, Ansammlung von erkannten Punkten an Kanten weiter verbessert. Dies ist notwendig, um Merkmale herauszufiltern, die schon bei leichtem Bildrauschen unzuverlässig erkannt werden können. Für diesen Schritt wird die Hauptkrümmung eines jeden Punktes über eine Hesse-Matrix und die Ableitungen über die Differenz der benachbarten Punkte errechnet. Da sich die Eigenwerte der Hesse-Matrix proportional zu der Hauptkrümmung verhalten, muss nicht jeder Eigenwert explizit berechnet werden, was die Performanz weiter steigert.

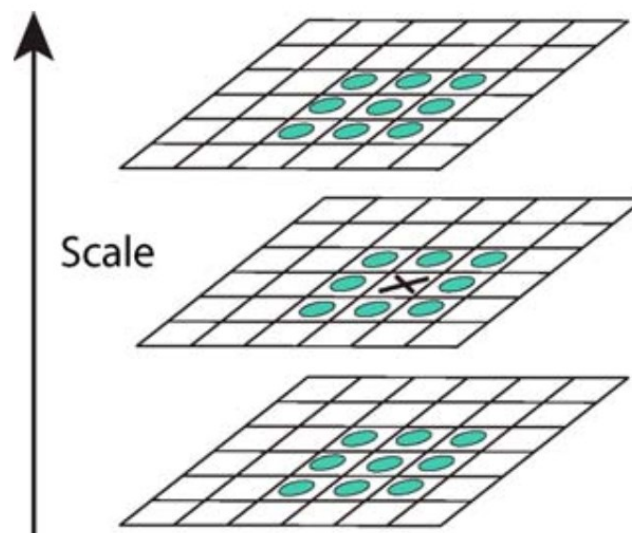


Abbildung 2.5: Anzahl der Vergleiche des SIFT-Algorithmus. Der Bildpunkt X wird mit allen grünen Bildpunkten verglichen. Insgesamt sind dies 26 Vergleiche. [Low04]

In der dritten Phase wird den erkannten Merkmalen eine gewichtete Richtung zugewiesen, um so zusätzlich zu der Invarianz der Skalierung eine Invarianz der Rotation herzustellen. Dazu werden die gewichteten Ausrichtungen der benachbarten Bildpunkte ausgewertet und der Winkel berechnet, welcher in diesem Bereich am stärksten bezüglich des Gradienten wirkt (Die Rotation wird in 36 Abstufungen gemessen, also in einem Abstand von zehn Grad pro vorhandener Richtung). Sollte sich mehr als einer der 36 Vektoren bezüglich seiner Länge von den restlichen Vektoren abheben, wird ein zusätzlicher Bildpunkt als Merkmal mit dieser Ausrichtung in die Liste der vorhandenen Bildmerkmale eingefügt.

Die letzte Phase behandelt die Erstellung eines Deskriptors für jeden Merkmalpunkt mit Invarianzen zu Belichtungen und Rotationen. Dafür wird ein 16×16 Feld um einen Merkmalpunkt in ein 4×4 Feld transformiert, in dem für jedes Feld ein *Histogram of oriented gradients* berechnet wurde, ähnlich jenem, welches in Phase drei beschrieben wurde. Der Unterschied ist, dass die Anzahl der Richtungen für ein einzelnes Histogramm von sechszehn auf acht reduziert ist und der Einfluss auf die Vektorlänge abnimmt, je weiter ein Gradient von dem Merkmalpunkt entfernt ist. Es entstehen $4 * 4 * 8 = 128$ Werte, die normalisiert werden und so den Merkmalvektor ergeben. Um eine Invarianz bezüglich der Rotation zu erhalten, wird von jeder gewichteten Richtung der Gradienten die Richtung des Merkmalpunkts subtrahiert. Somit ist die Orientierung der Gradienten relativ zu der des Merkmalpunkts. Um eine Invarianz bezüglich der Beleuchtung zu erreichen, wird ein Grenzwert von 0,2 für jeden der 128 Werte genutzt.

Zur Nutzung des SIFT-Algorithmus als Objekterkenner, werden erkannte Merkmale mit einer Datenbank aus Trainingsdaten unter dem Einsatz eines Nächste-Nachbarn-Ansatzes mittels des euklidischen Abstands abgeglichen. Zur Verbesserung der Erkennung werden dabei gefundene Paare verworfen, falls das Verhältnis ihres Abstands zu dem des übernächsten Nachbarn einen gewissen Grenzwert überschreitet.

Ein weiterer rotations- und skalierungsinvarianter Erkennungsalgorithmus wurde von Bay et al. [BETV08] im Jahre 2008 vorgestellt und nennt sich „*Speeded-Up Robust Features*“, kurz SURF. SURF ist in Teilen eine Abwandlung des SIFT-Algorithmus, die Unterschiede werden im Folgenden erläutert.

Statt einen Gaußschen Weichzeichner als Filter für die Bilder zu verwenden setzt SURF auf Integralbilder, was die Berechnung deutlich beschleunigt, da für die Berechnung der Pixelsumme nur vier Speicherzugriffe und drei Operationen notwendig sind. Mit jedem weiteren Bild in der Reihe steigt die Größe der Rechtecke, welche für die Integralbilder verwendet werden an, beginnend mit einem 9×9 Quadrat.

Die Erkennung von Merkmalpunkten basiert auf einer Hesse-Matrix, wobei deren Determinante als Maßeinheit genutzt wird. Punkte werden demnach an den Stellen gewählt, an denen die Determinante maximal ist. Nach einem Ansatz von Lindeberg [Lin98] wird die Determinante der Hesse-Matrix ebenfalls genutzt, um die Skalierung zu wählen. Die Maxima der Hesse-Matrix werden bezüglich der Skalierung und des Bildraums zudem interpoliert, da die Filterabstände zwischen den einzelnen Bildreihen jeweils hoch ausfallen.

Zur Bestimmung der Orientierung von Merkmalspunkten, wird zunächst die Ausrichtung des Punktes p evaluiert, basierend auf dem Haar-Wavelet⁵ aus einem Kreis von Pixeln mit Mittelpunkt p . Die Ausrichtung des Punktes lässt sich durch die Einzelberechnung der Summe aller Haar-Wavelet Ergebnisse in Bereichen von $\frac{\pi}{3}$ annähern. Ein solcher Bereich ist in Abbildung 2.6 durch die grau markierte Fläche gekennzeichnet. Die blauen Punkte stellen die Haar-Wavelet Rückmeldung dar, wobei deren Positionierung von der horizontalen bzw. vertikalen Richtung des Wavelet abhängen. Die Hauptausrichtung ergibt sich aus dem längsten Vektor aller somit berechneter Bereiche.

Der Deskriptor wird extrahiert, indem zunächst eine Fläche um den Punkt quadratisch in 4x4 Felder aufgeteilt wird, mit dem Merkmalspunkt als Mittelpunkt und einer Ausrichtung nach der Orientierung, welche im vorherigen Schritt berechnet wurde. Für jede dieser Regionen wird nun das Haar-Wavelet berechnet, indem die Region zusätzlich in 5x5 Felder unterteilt wird und für alle fünf Felder ein Vektor bestimmt wird, der sich, wie im vorherigen Schritt, nach der Haar-Wavelet Rückmeldung richtet. Bevor die Vektoren zu einem Gesamtergebnis summiert werden, gewichtet sie der Algorithmus zunächst unter Einsatz der Gaußschen Funktion mit dem Merkmalspunkt als Mittelpunkt. Dies steigert die Robustheit und wirkt Fehlern bei der Lokalisation entgegen. Um eine Invarianz bezüglich des Kontrastes herzustellen, kann der summierte Vektor schließlich als Einheitsvektor dargestellt werden.

Auf Grundlage des Bag-of-words Modells⁶, angewendet auf die Computer Vision-Domäne, und einen Naive Bayes-Klassifikator, werden Trainingsdaten aus Referenzbildern extrahiert und mit Hilfe von Tests evaluiert. Dies ermöglicht es, mit dem SURF-Algorithmus Objekterkennungen durchzuführen.

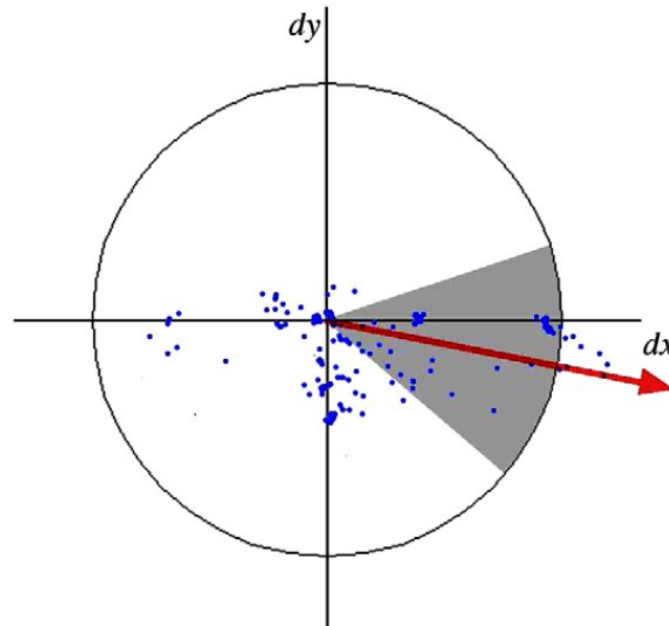


Abbildung 2.6: Bestimmung der Ausrichtung eines Merkmalspunkts in SURF [BETV08]

⁵unstetige, alternierende Zeit-Frequenz-Funktion

⁶Ermittlung der Frequenz von Wörtern in einem gegebenen Text, unter Vernachlässigung der genauen Position und Reihenfolge

Das Konzept der Mustererkennung aufgrund von Bildeigenschaften bildet die Grundlage für die Erkennung von Markern und der anschließenden Überlagerung in der Mixed Reality. Darüber hinaus eröffnet dieses Konzept die Möglichkeit, physische Entitäten aufgrund ihrer reinen Struktur und Beschaffenheit zu erkennen, ohne das Einblenden von Markern oder ähnlichen visuellen Hilfsmitteln. Dieser Ansatz ist in unterschiedlichen SDKs technisch umgesetzt und eignet sich für den durchzuführenden Vergleich als Überlagerungskonzept.

2.3.4 RFID

RFID steht für „*Radio-Frequency IDentification*“ und bezeichnet eine Kommunikations-Technologie mittels elektromagnetischer Wellen [Fin10]. Ein Transponder wird dabei an das Objekt, das erkannt werden soll angebracht und kann von einem Lesegerät kontaktlos ausgelesen werden.

Ähnlich dem Einsatz von GPS, erlaubt der Einsatz von RFID-Technologien keine zuverlässige Erkennung mit hoher Genauigkeit, da diese von äußeren Einflüssen gemindert wird und die technischen Grenzen für diesen Anwendungsfall rasch erreicht sind [CFA+11]. So muss die genutzte Frequenz beispielsweise stets der Umgebung angepasst sein, um eine Kollision mit weiteren Kanälen zu vermeiden [Fin10]. Um eine Erkennung mit Hilfe von RFID-Technologie zu ermöglichen, muss das Objekt darüber hinaus um einen RFID-Chip erweiterbar sein, was eine Vielzahl von Anwendungsfällen erschwert. Aus diesem Grund und der fehlenden Präzision bei der Überlagerung von physischen Entitäten, eignet sich ein RFID-Ansatz nicht für den Vergleich mit den übrigen Konzepten.

Die vorgestellten Algorithmen und Überlagerungsansätze schaffen ein grundlegendes Wissen über die Arbeitsweise moderner Mixed Reality-Applikationen. Durch die Aufarbeitung konnten zwei Konzepte für den durchzuführenden Vergleich als geeignet eingestuft werden, die übrigen beiden wurden aufgrund ihrer fehlenden oder zu gering ausgeprägten Eigenschaft bezüglich einer möglichst präzisen Überlagerung von virtuellen Objekten vernachlässigt. Um die Entwicklung der vorgestellten Ansätze zu unterstützen, existieren unterschiedliche SDKs, welche die hier genannten Techniken und Konzepte aufgreifen und implementieren. Ein Ausschnitt dieser technologischen Vertreter wird in Kapitel 3 beschrieben.

3 Stand der Technik

Dieses Kapitel beschreibt bestehende Technologien und deren Einsatzgebiete im Bereich der erweiterten Realität, damit eine Auswahl an Umsetzungen der Überlagerungskonzepte für den durchzuführenden Vergleich herangezogen werden kann. Die technische Realisierung dieser Konzepte ist dabei in zahlreichen SDKs und Bibliotheken zusammengefasst. In diesem Kapitel werden vier dieser Umsetzungen vorgestellt (vgl. Abschnitt 3.1). Um die Implementierungen auszuführen und dem Benutzer immersiv darzustellen, benötigt es entsprechende Geräte. Zwei Vertreter von Augmented Reality-Datenbrillen werden in Abschnitt 3.2 beschrieben. In der Realität mangelt es derzeit an weitreichenden Anwendungsfällen der Mixed Reality-Technologie, sieht man von der Unterhaltungsbranche ab [Fit11]. Innerhalb des Forschungsgebiets wurden jedoch bereits mehrere mögliche Einsatzfelder veröffentlicht und diskutiert. Vier Beispiele aus unterschiedlichen Branchen werden in Abschnitt 3.3 vorgestellt. Abschließend werden drei verwandte Arbeiten benannt und deren Abgrenzung bzw. Gemeinsamkeit zu diesem Werk erläutert (Abschnitt 3.4).

3.1 Augmented Reality SDKs

Sowohl in der Industrie als auch der Forschung wurden Frameworks entwickelt, die als Grundlage für die Interaktion in der Mixed Reality dienen. Dabei ähneln sich die von den Herstellern eingesetzten Computer Vision-Techniken oft, deren Umfang fällt jedoch unterschiedlich aus.

Im Folgenden werden vier Vertreter von Mixed Reality Frameworks vorgestellt und im Hinblick auf das Angebot an Objekterkennungsansätzen hin untersucht: ARCore ist ein SDK des Unternehmens Google, das eine Markererkennung unterstützt (vgl. Abschnitt 3.1.1). Das SDK ARKit stammt von Apple und stellt dem Benutzer zwei Objekterkennungstechniken zur Verfügung (vgl. Abschnitt 3.1.2). OpenCV ist eine Open Source-Bibliothek, die eine Vielzahl an Computer Vision-Algorithmen bereitstellt – Teile davon können auch zur Objekterkennung genutzt werden (vgl. Abschnitt 3.1.3). Das SDK Vuforia stellt ebenfalls mehrere Techniken für unterschiedliche Einsatzgebiete zur Erkennung von Objekten bereit (vgl. Abschnitt 3.1.4).

3.1.1 ARCore

ARCore ist Googles SDK zur Unterstützung der Programmierung von Augmented Reality [Goo18]. ARCore stellt eine Erkennung von statisch positionierten 2D-Bildern zur Verfügung, die vorab in einer Datenbank als Referenz hinterlegt werden. Das Limit beträgt hierbei 1000 Referenzbilder. ARCore kann bis zu 20 Bilder gleichzeitig in der Umgebung erkennen, allerdings sind diese pro Bild auf eine Instanz beschränkt. ARCore informiert den Nutzer über die Qualität der hinterlegten Bilder bezüglich ihrer Tracking-Fähigkeit.

3.1.2 ARKit

ARKit ist das Virtual Reality SDK des Unternehmens Apple. Es bietet zwei Ansätze für die Erkennung von Objekten im Raum, die im Folgenden beschrieben sind [App18]:

1. **Image Detection:** ARKit erlaubt das Erkennen von 2D-Bildern im Raum durch das Hinterlegen von Referenzbildern sowie der tatsächlichen Größe des physischen Objekts in einer Datenbank. Die Größe des 2D-Bilds wird hierbei verwendet, um dessen Abstand zur Kamera korrekt zu visualisieren. ARKit informiert den Nutzer über die Qualität der hinterlegten Bilder bezüglich ihrer Tracking-Fähigkeit.

Translationen und Rotationen des physischen Bilds zur Laufzeit beeinflussen die virtuelle Abbildung nicht.

2. **Object Detection:** Das SDK stellt neben der Erkennung von 2D-Bildern auch eine Erkennung von 3D-Objekten bereit. Dazu erfasst der Benutzer zunächst mit Hilfe einer App das physische Abbild, welches von ARKit erkannt werden soll. Für diesen Vorgang wird das Objekt zunächst auf einer möglichst freien Fläche positioniert, so dass der Algorithmus das Objekt erkennen kann. Ein Rahmen wird daraufhin eingeblendet, der die von dem Algorithmus erkannten Grenzen anzeigt. Dieser Rahmen kann von dem Benutzer in seiner Größe und Position nachträglich verändert werden. Ist der Rahmen korrekt an das Objekt angebracht kann der Scan-Vorgang gestartet werden. Dafür bewegt der Nutzer den Kamerasensor um das Objekt, bis der Algorithmus ausreichend Informationen gesammelt hat. Schließlich kann der Ursprung auf das Objekt platziert werden und die Referenzdaten zur weiteren Benutzung exportiert werden. Neben der bereitgestellten App zum Scannen von Objekten stellt ARKit auch Schnittstellen zur Verfügung, um eigene Scanner in eine Entwicklung zu integrieren.

3.1.3 OpenCV

OpenCV ist eine Open Source-Bibliothek und beinhaltet über 2500 Computer Vision Algorithmen [Ope18]; unter diesen Algorithmen befinden sich sowohl Funktionen zur Markererkennung als auch zur 2D- und 3D-Objekterkennung.

Die Markererkennung in OpenCV basiert auf der ArUco Library [GMMM14] von Garrido-Jurado et al. [GMMM14] (vgl. Abschnitt 2.3.3). Das Vorgehen gliedert sich dabei zunächst in das Erkennen von Referenzmarkern in Bildern und die Schätzung der Kamerahaltung durch einzelne Marker im Raum. Des weiteren stellt OpenCV Algorithmen zur Erkennung von ChArUco-Brettern [Ope15] zur Verfügung, die aufgrund ihrer visuellen Beschaffenheit (Eine Kombination aus Schachbrettmuster und ArUco-Markern, siehe Abbildung 3.1) eine sehr hohe Robustheit und Verlässlichkeit erwirken und demnach gut für Kalibrierungen geeignet sind. Zur 2D-Objekterkennung kommt ein Algorithmus zum Einsatz, der, ähnlich dem SURF-Algorithmus (siehe Abschnitt 2.3.3), Haar-Wavelets als Erkennungsgrundlage benutzt [LM02]. Als 3D-Objekterkennung wird ein Convolutional Neural Network [KSH12]-basierter Lernalgorithmus verwendet, der 2D-Bilder mit Metadaten über deren Position und Klasse als Grundlage nutzt.

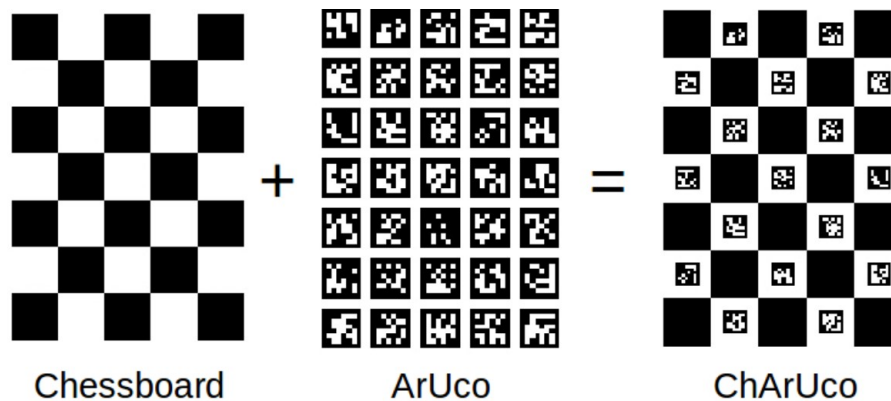


Abbildung 3.1: ChArUco-Brett [Ope15]

3.1.4 Vuforia

Vuforia von PTC Inc. entwickelte mehrere Tracking-Ansätze zur Erkennung von Objekten im Raum, die im Folgenden vorgestellt werden [PTC18].

Um eine Erkennung zur Laufzeit zu ermöglichen, müssen Referenzdaten in einer Datenbank hinterlegt werden, auf die Vuforia zur Laufzeit zugreifen kann. Für diese Referenzdaten wird zudem eine Einstufung ihrer Tracking-Fähigkeit vorgenommen, um den Benutzer über deren Qualität zu informieren.

1. **Cylinder Targets:** Dieser Ansatz ist speziell für zylindrische Objekte geeignet. Bilder können dabei auf der Mantelfläche sowie der Ober- und Unterseite erkannt werden.

Um ein Objekt für den Vuforia Target Manager zu registrieren, muss die Seitenlänge, sowie der Durchmesser der Ober- und Unterseite des Zylinders bekannt sein und ein zu erkennendes Bild der Zylinderoberfläche existieren.

Das Format ist JPG oder PNG mit maximal 2 Megabyte. Der Benutzer kann dabei zwischen RGB-Farben oder Graustufen entscheiden.

2. **Image Targets:** Das Tracking von Bildern setzt auf die NFT-Technik (Abschnitt 2.3.3) und erlaubt auch dann eine Erkennung, falls nur ein Teil des Bildes im Sichtfeld erkennbar ist.

Als Parameter muss der Benutzer bei der Erstellung die tatsächliche Größe des physischen Bildes angeben.

Das Format ist JPG oder PNG mit maximal 2 Megabyte. Der Benutzer kann dabei zwischen RGB-Farben oder Graustufen entscheiden.

Um das Tracking zusätzlich zu verbessern, kann die Position des Sichtfelds in die Berechnung einbezogen werden. Dies erlaubt es dem Benutzer den Blick von dem Bild zu lösen und dennoch das virtuelle Modell zu sehen.

3. **Model Targets:** Um diese Technik zu nutzen, muss ein 3D-Modell des physischen Objekts, welches von dem Tracker erkannt werden soll, digital vorliegen. Grundlage für die Erkennung ist dabei, dass das Objekt unverändert zu dem digitalen Abbild bleibt und eine erkennbare Oberflächenstruktur aufweist (keine reflektierenden Oberflächen); Ebenso sollte das Objekt

stationär verbleiben. Mit Hilfe des von Vuforia bereitgestellten *Model Target Generator*¹ lassen sich nun mehrere Betrachtungspunkte hinzufügen, die als Grundlage für eine Erkennung der physischen Entität dienen. Um eine Erkennung aus mehreren Betrachtungspunkten ohne vorherige Zuweisung zu ermöglichen, muss das Modell zunächst über einen Cloud-Dienst Trainingsdaten generieren.

Im Vergleich zu Cylinder und Image Targets wird das Koordinatensystem nicht konventionell platziert, sondern richtet sich ausschließlich nach dem vorgegebenen CAD-Modell.

4. **Multi-Targets:** Um mehrere Image Targets als ein Objekt zu erkennen, existiert ein Multi-Targets Tracker. Jede Seite des Objekts kann dabei gleichzeitig erkannt werden, da die Positionen der Einzelbilder zueinander in Relation stehen. Das gesamte Objekt kann somit erkannt werden, sobald ein einziges Bild der Gruppe erkannt wurde. Ein Nachteil ist dabei, dass eine Veränderung des Objekts zu einer fehlerhaften Darstellung führt.

Der Benutzer kann innerhalb eines separaten *Target Managers* auf einen vorgefertigten Quader zugreifen und seinen sechs Seiten ein Bild zuordnen, um den Erstellungsprozess zu vereinfachen. Dabei gibt er lediglich die Höhe, Breite und Dicke des Quaders an, sowie die tatsächliche Größe des physischen Abbilds.

Das Format ist JPG oder PNG mit maximal 2 Megabyte. Der Benutzer kann dabei zwischen RGB-Farben oder Graustufen entscheiden.

5. **Object Targets:** Mit Hilfe der Vuforia Object Scanner App, lässt sich ein Objekt in einer präparierten Umgebung über einen Kamerasensor erfassen und dessen Oberflächen-Struktur als auch -Textur abspeichern. Daraufhin ist es auf Grundlage dieser Daten möglich eine generelle Erkennung des Objekts durchzuführen.

Zunächst wird dafür ein vorgefertigtes Messbild unter dem zu erkennenden physischen Objekt platziert. Daraufhin wird die Szene von unterschiedlichen Sichtwinkeln mit dem Vuforia Object Scanner erfasst. Abbildung 3.2 zeigt den erfolgreichen Scan-Vorgang eines Spielzeugautos des Vuforia Object Scanners.

6. **VuMark:** VuMarks ist Vuforias Erstellungsempfehlung planarer Referenzmarker (vgl. Abschnitt 2.3.2), wobei Firmen Marker erstellen können, die zum einen als Logo dienen und zum anderen als Marker erkannt werden.

Bei jeder Tracking-Technik wird eine ausreichende und flächendeckende Beleuchtung vorausgesetzt. Zudem stellt jede/s Bibliothek/SDK eine Dokumentation mit Tipps und Hinweisen zur Verbesserung der Tracking-Fähigkeit zur Verfügung. Die Zusammenfassung der bestehenden SDKs und Bibliotheken zeigt, dass eine Vielzahl von technischen Vertretern für die Umsetzung von Konzepten der Mixed Reality bereitstehen. Somit stellen alle vorgestellten Technologien Implementierungen für Überlagerungskonzepte zur Verfügung; Unterschiede sind aufgrund der oft proprietären Software nicht detailliert bestimmbar. Deshalb richtet sich die Auswahl eines geeigneten SDKs nach einer leichten Bedienbarkeit, der Portabilität und der Plattformunabhängigkeit. Welche technischen Vertreter für die Überlagerungskonzepte gewählt werden, wird in Kapitel 4 elaboriert.

¹<https://library.vuforia.com/articles/Solution/model-target-generator-user-guide.html>

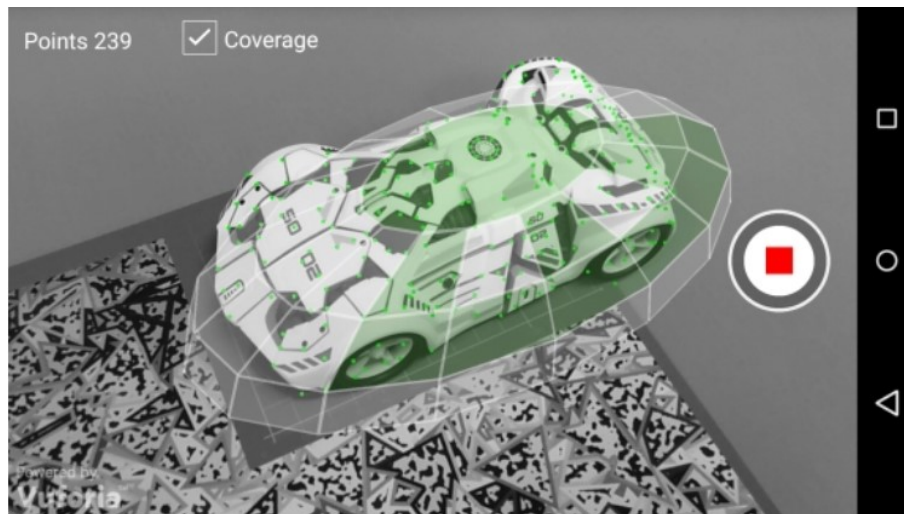


Abbildung 3.2: Erfassung mittels des Vuforia Object Scanner [PTC18]

3.2 Datenbrillen

Mixed Reality-Datenbrillen existieren in vielfältigen Ausführungen. Dabei unterscheiden sich die Geräte besonders in der Art der Immersion (vgl. Abschnitt 2.2). Da für diese Arbeit eine Überlagerung von physischen Entitäten vorgesehen ist, wird ein Ausschnitt jener Datenbrillen vorgestellt, die eine reale Umgebung um virtuelle Hologramme im Sichtfeld des Benutzers ergänzen. Zwei Geräte, welche eine solche Immersion herstellen sind die Magic Leap One (vgl. Abschnitt 3.2.1) und die Microsoft HoloLens (vgl. Abschnitt 3.2.2).

3.2.1 Magic Leap One

Die Datenbrille Magic Leap One wurde von Magic Leap Inc. entwickelt. Eine Spezifikation auf der Entwicklerwebseite² enthält Informationen zu der eingesetzten Hardware. Die Datenbrille ist über zwei Kabel mit einer weiteren Hardware-Komponente verbunden, die am Körper angebracht wird. Zudem wird die Datenbrille mit einem Controller ausgeliefert, der alle sechs Freiheitsgrade unterstützt. Über die acht angebrachten Sensoren macht der Hersteller keine genaueren Angaben. Der Prozessor basiert auf einer Nvidia Tegra X2 SoC mit zwei Denver 2.0 64-bit Kernen und vier ARM Cortex A57 64-bit Kernen. Die integrierte Grafikeinheit ist eine auf Pascal basierende GPU mit 256 CUDA Kernen. Die Datenbrille ist des Weiteren mit 8 GB RAM ausgestattet. Der Speicherplatz auf der Magic Leap One beträgt 128 GB. Das Betriebssystem basiert auf Android und wurde für die Datenbrille modifiziert.

Die Entwicklung für die Magic Leap One wird durch das Lumin SDK³ vereinfacht. Mit Hilfe des SDKs können Kopfbewegungen und Gesten erkannt und weiterverarbeitet werden. Ebenso ist es möglich über Bildverarbeitungsalgorithmen die Umgebung zu verarbeiten und abzuspeichern,

²<https://www.magicleap.com/magic-leap-one>

³<https://creator.magicleap.com/learn/guides/creator-portal>

sodass Hologramme an ihrer Position verbleiben, obwohl sie das Sichtfeld des Benutzers verlassen. Schließlich können Audioquellen im Raum positioniert werden und so an den Benutzer übertragen werden, dass die Ursprungsrichtung hörbar wird.

3.2.2 Microsoft HoloLens

Die folgenden technischen Details zu der Datenbrille von Microsoft sind von Zeller und Bray [ZB18] veröffentlicht worden.

Die HoloLens (siehe Abbildung 3.3) ist ein kabelloses holographisches Headset. Sie besteht aus vier Kamerasensoren, welche die Umgebungsanalyse der Brille durchführen. Zur besseren Erkennung der Umgebung besitzt die HoloLens ebenso eine Tiefenkamera. Um die Sensordaten auszuwerten, beispielsweise zur Erkennung der Kopfneigung, wurde eine inertielle Messeinheit verbaut. Weitere Sensoren sind eine Fotografie- bzw. Videokamera, zur direkten Aufnahme des Sichtfelds (die virtuelle und reale Umgebung wird simultan erfasst und abgespeichert), vier Mikrofone, um die Spracherkennung zu steuern, sowie ein Umgebungslichtsensor. Das Display der Datenbrille besteht aus zwei durchsichtigen Wellenleitern und zwei HD-16:9-Licht-Generatoren. Um Hologramme in einer korrekten Distanz darzustellen, setzt die HoloLens eine automatische Pupillen-Abstands-Kalibrierung ein [BZ18]. Die holographische Auflösung beträgt insgesamt 2,3 Mio. Lichtpunkte, die holographische Dichte über $2,5 \times 10^3$ Radianten (Lichtpunkte pro Radiant). Der Prozessor basiert auf einer Intel 32-Bit-Architektur. Zudem kommt ein speziell angefertigter holographischer Prozessor von Microsoft zum Einsatz. Der Arbeitsspeicher unterteilt sich in 64 GB Flashspeicher und 2 GB RAM. Das Betriebssystem der HoloLens ist Windows 10.



Abbildung 3.3: Microsoft HoloLens [Mic18]

Um Entwicklern die Programmierung von Applikationen für die HoloLens-Datenbrille zu vereinfachen, entwarf Microsoft das quelloffene Mixed Reality Toolkit⁴ für die Entwicklungsumgebung Unity. Das Toolkit befindet sich zu diesem Zeitpunkt noch in der Entwicklungsphase, erlaubt aber bereits die Nutzung zahlreicher Schnittstellen für die Datenbrille. Der folgende Abschnitt basiert auf der Dokumentation des Toolkits [Mic19] und erläutert einige der Funktionen, die den Entwicklern zur Verfügung gestellt werden.

⁴<https://github.com/Microsoft/MixedRealityToolkit-Unity>

Ein Großteil der bereitgestellten Funktionen besteht aus GUI-Elementen, die der Entwickler leicht in seine eigene Applikation einbetten kann und deren Interaktionen bereits für den Einsatz mit der HoloLens-Datenbrille abgestimmt sind. So können vorgefertigte Buttons in das Sichtfeld des Benutzers eingebettet werden, die ohne eine weitere Implementierung die Gestensteuerung unterstützen und verarbeiten. Ebenfalls ist es möglich 3D-Objekte mit Skripten zu versehen, die es dem Benutzer ermöglichen das Objekt zu manipulieren und so eine Translation, Rotation oder Skalierung mittels der Gestensteuerung erlauben. 3D-Objekte können zudem mit Skripten versehen werden, die mit Hilfe der Umgebungserkennung der Datenbrille Oberflächen identifizieren und das Objekt automatisch an der Umgebung ausrichten oder diese im Falle einer Kollision abstoßen. Schließlich kann der Entwickler Hilfswerkzeuge zur Verwendung der Sprachsteuerung einsetzen, welche die Gestensteuerung zur Navigation unterstützt.

Da die Veröffentlichung der Datenbrille Magic Leap One nur kurze Zeit zurückliegt und Drittsoftware das Gerät häufig noch nicht unterstützt, stellt die Microsoft HoloLens, mit einem eigenen SDK und einer hohen Zahl an verfügbaren SDKs von Drittherstellern, für den durchzuführenden Versuch eine erleichterte Umsetzung und Bedienung der technischen Implementierungen aller Überlagerungskonzepte zu Verfügung.

3.3 Einsatzgebiete

Die Anwendungsfälle der Mixed Reality in der tatsächlichen Realität sind ein Gebiet von großem Interesse; so zumindest belegen es die Veröffentlichungen der ersten und zweiten Dekade der ISMAR Konferenz [KBB+18]. Nach den Zahlen gehört das Forschungsgebiet *AR Application* zu den fünf beliebtesten Forschungsthemen, bemessen an der Anzahl der Zitate pro Jahr von Veröffentlichungen mit dem Schwerpunkt Anwendungsfälle. Dennoch schaffen es Mixed Reality-Technologien nur in seltenen Fällen tatsächlich in den Arbeitsalltag [Fit11]. In diesem Abschnitt werden vier Anwendungsfälle beschrieben, die einen Einsatz von Mixed Reality-Technologie empfehlen.

Ein Beispiel aus der Industrie erläutert das Überlagern einer Fabrikanlage mit virtuellen Maschinen zur Vermeidung von Fehlern bei der Planung (vgl. Abschnitt 3.3.1). Zudem wird ein Anwendungsfall beschrieben, bei dem 3D-Modelle auf Bildaufnahmen von Anlagen überlagert werden, um einen direkten Abgleich der beiden Datensätze zur Wartung zu ermöglichen. Weitaus kritischere Fehler sollen unter dem Einsatz von Mixed Reality im Bereich der Medizin vermieden werden, indem einem Chirurgen grafische Hilfestellungen während des Eingriffs auf eine Datenbrille übermittelt werden (vgl. Abschnitt 3.3.2). Im Bereich der Unterhaltungsbranche wird ein Ansatz mit wissenschaftlichem Kontext vorgestellt, der ein Jump & Run-Spiel beschreibt, das die reale Umgebung auf Plattformen hin untersucht, auf die der Spieler seine Figur leiten muss (vgl. Abschnitt 3.3.3).

3.3.1 Industrie

Planung von Anlagen

Um Planungsfehler zu vermeiden und so Kosten einzusparen, stellen Pentenrieder et al. [PBDM07] einen Mixed Reality-gestützten Ansatz bei der Planung von Industrieanlagen vor. Die Grundidee besteht darin visuelle Überlagerungen mit Hilfe von Datenbrillen einzublenden, anstelle komplette digitale Fabrikhallen zu entwerfen, welche häufig veraltete Informationen beinhalten oder denen es an 3D-Modellen mangelt. Dem entgegen steht jedoch der Standpunkt, dass Mixed Reality-Technologien nicht robust, verlässlich, benutzerfreundlich und ergonomisch genug sind und sich somit für den regelmäßigen Gebrauch nicht eignen. Pentenrieder et al. [PBDM07] entwarfen deshalb drei Kriterien, die bei der Erstellung einer AR-gestützten Planungsanwendung zu erfüllen sind.

1. **Usability:** Sowohl die eingesetzte Software als auch Hardware muss so einfach wie möglich nutzbar sein. Funktionen, die nicht benötigt werden sollten nicht zur Verfügung stehen, um Fehler oder Verwechslungen zu vermeiden und die Komplexität der Anwendung zu minimieren.

Um eine benutzerfreundliche Umgebung zu erzeugen, setzen Pentenrieder et al. [PBDM07] auf Marker-basierte Objekterkennung. Diese ist besonders leicht umzusetzen, da nur die gedruckten Marker und eine Kamera notwendig sind, um das Tracking zu realisieren. Wichtig ist hierbei, dass die Fabrikanlage ausreichend beleuchtet ist und die Marker sichtbar platziert werden. Jedem Marker kann daraufhin ein 3D-Modell zugewiesen werden, das angezeigt werden soll, falls der Marker von der Kamera erkannt worden ist. Die Planungssoftware enthält über diese automatische Zuweisung hinaus manuelle Translationsmöglichkeiten, um die 3D-Modelle nachträglich im Raum zu bewegen.

2. **Analysefunktionen:** Für den Einsatz von Planungssoftware sollten Funktionen wie Abstandsmesser oder Kollisionsabfragen zur Verfügung gestellt werden. Deren Bedienung muss dabei verständlich dokumentiert und die Ergebnisse präsentierbar sein.

Um eine Analyse der virtuellen Komponenten und der Umgebung zu ermöglichen, können die 3D-Modelle vier Zustände annehmen: *Sichtbar*, *Unsichtbar*, *Drahtmodell* und *Verdeckung*. *Sichtbar* und *Unsichtbar* blenden das Modell jeweils ein oder aus. *Drahtmodell* entfernt die Seitentexturen des Modells und blendet an deren Stelle ein Drahtmodell ein, um die Einbettung in die physische Umgebung besser zu erkennen. *Verdeckung* blendet das Modell ebenfalls aus, verdeckt jedoch weiterhin übrige 3D-Modelle in der Umgebung. Eine weitere Funktion erlaubt das Ausblenden von Objekten über eine Clippingebene, die sich im Raum entlang der drei Koordinatenachsen platzieren lässt. Die Clipping-Ebene lässt sich ebenfalls benutzen, um eine Kollision mit anderen Objekten zu messen. Die Planungssoftware enthält darüber hinaus einen 3D-Distanzmesser. Außerdem ist ein Genauigkeitsmesser integriert, der im Folgenden beschrieben ist.

3. **Genauigkeit:** Im Bereich der Planung müssen Überlagerungen präzise auf die reale Welt übertragen werden. Um dieser nichtfunktionalen Anforderung gerecht zu werden, müssen angepasste Tracking-Techniken verwendet werden und die eingesetzten Geräte korrekt kalibriert sein. Die Ergebnisse der Planungsphase müssen ebenfalls auf ihre Qualität hin überprüft werden.

Die Genauigkeit in der Überlagerung von virtuellen und physischen hängt von mehreren Einflussfaktoren ab: Die Kalibrierung der Kamera, die 6DOF der Kamera bezüglich des Markers, die 6DOF-Verschiebung bezüglich der Marker und dem physischen Koordinatensystem, sowie die Tessellation der virtuellen Geometrie. Vereint man diese Faktoren, lässt sich aus ihnen ein Gesamtbild zeichnen, wodurch eine Aussage über die Genauigkeit der Darstellung in Form eines konkreten Wertes getroffen werden kann. Dieses Maß basiert auf dem *13005. Leitfaden zur Angabe der Unsicherheit beim Messen (GUM)* und kann mathematisch berechnet werden. Der Fehlerbereich der Überlagerung wird dem Anwender schließlich visuell in Form von Ellipsen dargestellt.

Wartung

Die Wartung von Maschinen und Anlagen kann eine herausfordernde Aufgabe darstellen, insbesondere dann, wenn die Baupläne nicht mit dem installierten Objekt übereinstimmen und es nicht möglich ist nach einem definierten Schema zu verfahren. Dabei muss ein Mangel nicht zwangsläufig negative Auswirkungen oder Probleme am Objekt selbst hervorrufen, es kann die Wartung aber dennoch erschweren [GSN09]. Ein Beispiel ist die Markierung von Schaltern und Drehreglern durch Lackierungen. Fehlt diese Markierung bei einigen Schaltflächen, kann der Mechaniker nicht mehr zuverlässig entscheiden, wie bei einem Wartungsschritt zu verfahren ist. Eine Möglichkeit wäre es, spezifische CAD-Modelle für jede installierte Anlage oder Maschine anzufertigen, um individuelle Baumängel zu berücksichtigen – dies ist jedoch sehr zeitaufwendig [GSN09].

Georgel et al. [GSN09] stellt als Lösungsvorschlag einen Ansatz vor, der Bilder des Wartungsobjekts aus unterschiedlichen Sichtwinkeln mit den CAD-Daten überlagert und eine 3D-Navigation erlaubt. Die grafische Oberfläche ist in Abbildung 3.4 zu sehen. Die Baumstruktur (a) liefert eine Übersicht der vorhandenen Komponenten und erlaubt dem Benutzer das ein- und ausblenden dieser. Abschnitt (b) zeigt eine Übersicht aller aufgenommenen Ansichten des physischen Objekts, wobei der Benutzer mit Hilfe der Schaltflächen in (c) die Szenerie Verschieben kann, oder an Stellen hinein- bzw. herauszoomen kann. Die Möglichkeiten der Translationen errechnet ein Algorithmus mit Hilfe von virtuellen 3D-Punkten, welche als Schnittpunkte des Sichtfelds und 3D-Strukturen im Bild festgelegt werden. So können die Bilder als Nachbaransichten in Relation gestellt werden, abhängig von der relativen Position der aktuellen Sicht und der erkannten 3D-Punkte.

Eine im Rahmen der Veröffentlichung durchgeführte Studie zeigte, dass der Algorithmus Nachbaransichten zu großen Teilen gleich einordnet wie die Probanden.

3.3.2 Medizin

Nach Wang et al. [WWL+16] stellen Operationen im Inneren des menschlichen Körpers, bei denen ein Chirurg kein direktes Sichtfeld zu seinem Werkzeug hat eine Herausforderung dar, da der behandelnde Chirurg gezwungen ist, einen externen Bildschirm als Hilfestellung zu benutzen. Durch diesen ständigen Sichtwechsel wird die Hand-Augen-Koordination des Chirurgen beeinträchtigt – die Gefahr eine ungewollte Handbewegung zu vollziehen und den Patienten damit lebensgefährlich zu verletzen steigt. Der in der Medizin etablierte Ansatz ist der Einsatz von Fluoroskopie, um den Arzt bei der Operation zu unterstützen, jedoch erleidet der Patient dabei eine hohe Strahlenbelastung und die Technik ist fehleranfällig, da Fettleibigkeit oder Darmgase die Auswertung verfälschen

3 Stand der Technik



Abbildung 3.4: Visual Inspection and Documentation GUI [GSN09]

können. Ein Beispiel einer solchen Operation ist das Einsetzen einer Schraube in das Iliosakralgelenk eines Patienten. Die Folge der beschriebenen Einschränkungen ist, dass 2 bis 15% der Schrauben fehlerhaft positioniert werden.

Um den Chirurgen bei der Operation zu unterstützen und die Fehlerquote zu verringern, stellt Wang et al. [WWL+16] einen AR-Ansatz vor, bei dem eine Datenbrille eingesetzt wird und der Körper des Patienten, die Kopfbewegung des Chirurgen sowie das eingesetzte Werkzeug mit Hilfe eines Infrarot Tracking-System erkannt und auf das Sichtfeld des Chirurgen abgebildet wird. Dafür werden Referenzvorrichtungen am Becken des Patienten, dem chirurgischen Werkzeug und der Datenbrille selbst positioniert. Um die relative Position der Vorrichtung bezüglich des zu erkennenden Werkzeugs zu erhalten, folgt zunächst eine Kalibrierung der Sensoren. Somit kann eine Translation des Werkzeugs in Echtzeit auf das virtuelle Ebenbild übersetzt werden. Damit eine Überlagerung von virtuellen Komponenten auf die physische Umgebung ermöglicht werden kann, wird das virtuelle und das tatsächliche Koordinatensystem zunächst erfasst und abgeglichen. Virtuell wird dazu eine Bilderkennungstechnik eingesetzt, während in der physischen Umgebung eine Sonde zum Einsatz kommt. Die Erfassung der Infrarotmarker findet daraufhin über eine Singulärwertzerlegung statt.

Um die Kopfbewegung des Chirurgen zu kalibrieren, werden die Messwerte der Infrarotmarker an der Datenbrille und am Körper des Patienten unter dem Einsatz einer Transformationsmatrix ausgewertet, sodass die virtuellen Bilder im Sichtfeld des Chirurgen den 6DOF der Kopfbewegung folgen.

Sind die Geräte kalibriert, wird dem Chirurgen in Echtzeit die vorgesehene Bahn der Schraube gemeinsam mit einem virtuellen Abbild des relevanten Teils des Beckens und den umliegenden Blutgefäßen in das Sichtfeld eingeblendet. Ebenfalls kann der Chirurg ein 3D-Modell seines

Werkzeugs sehen, welches sich synchron zu den Translationen der tatsächlichen Handbewegung bewegt. In dieser Ansicht können Elemente beliebig ein- und ausgeblendet werden, um eine bessere Übersicht zu gewährleisten. Die Modelle des Beckens und der Blutgefäße stammen dabei aus Daten von vorherigen Computertomographiebildern; Das Becken und die Schraube wurden im Vorfeld digital rekonstruiert.

Eine Studie zeigte, dass alle zwölf Testfällen an Leichnamen erfolgreich und hinsichtlich der Soll-Position der Schraube akkurat verliefen.

3.3.3 Unterhaltungsbranche

Mixed Reality gewinnt in der Computerspielindustrie immer mehr an Bedeutung: auf der Spieleplattform Steam⁵ existieren bereits über 150 Spieletitel, die auf Mixed Reality-Technologien basieren [Val18].

Chekhlov et al. [CGCM07] stellen einen Spiele-basierten Ansatz vor, mittels AR-Technologie stetig neue planare Flächen im Raum zu erkennen, um einen Spieler über diese Oberfläche hinweg zu bewegen.

Statt auf den Einsatz von Referenzmarkern, welche eine vorherige Manipulation des Spielfelds erforderlich machen, setzen Chekhlov et al. [CGCM07] auf einen SLAM („*Simultaneous localization and mapping*“)-Ansatz. Dabei verarbeitet ein Algorithmus die von der Kamera erfasste Umgebung, während gleichzeitig neue Merkmalpunkte in Richtung der Kamerabewegung extrahiert werden. Die Messung wird dabei mit einem FAST-Algorithmus (vgl. Abschnitt 2.3.3) durchgeführt, der Merkmalpunkte und Deskriptoren erkennt, welche jenen des SIFT-Algorithmus (vgl. Abschnitt 2.3.3) ähneln. Die Genauigkeit von Überlagerungen der virtuellen Ebenen und den physischen planaren Flächen wird in diesem Fall vernachlässigt, stattdessen wird lediglich ein Rechteck über die Stelle der erkannten Merkmalpunkte gelegt. Da die Größe der virtuellen Ebene von den erkannten Merkmalpunkten abhängt, kann sie sich im Verlauf der Kamerabewegung hinsichtlich ihrer Größe verändern.

Das Spiel *Ninja on a Plane* visualisiert einen grünen Ninja der planare Ebenen, die in der physischen Umgebung existieren, herabspringt. Dabei kann der Ninja abstürzen, falls die Ebenen in horizontaler und/oder vertikaler Richtung zu weit auseinander liegen. Ein Zeitlimit zwingt den Spieler dazu, so schnell wie möglich planare Ebenen in Form von Büchern oder ähnlichen quadratischen Flächen herzustellen, damit der Ninja eine unter ihm liegende Zielplattform erreicht.

Die Framerate des Spiels lag zwischen 15-20 FPS, bei einem Limit von bis zu 50 Merkmalpunkten. Bei zu schnellen Kopfbewegungen konnte die Erkennung nicht mehr zuverlässig durchgeführt werden. Die Genauigkeit der Überlagerung von planaren Flächen in der physischen Umgebung wurde negativ beeinflusst, wenn der Spieler das Sichtfeld an eine Stelle mit wenigen Merkmalpunkten bewegte. Ebenso kam es zu Ungenauigkeiten, falls der Spieler das Sichtfeld komplett von dem bereits verarbeiteten Spielfeld abwendete. Zwar besitzt das entworfene System Mechanismen, um sich nach einem kurzzeitigen Abbruch der durchgängigen Objekterkennung wieder zu initialisieren, dennoch kann dies zu Fehlern führen, falls die Kameraposition nicht mehr korrekt approximiert wird. Die Entwickler empfehlen für diesen Fall eine Zurücksetzung des Spiels.

⁵<https://store.steampowered.com/>

Gerade im Bereich der Industrie und der Wartung wurden Anwendungsfälle beschrieben, welche den Fokus dieser Arbeit darstellen. In den Anlagen der Produktion kann die vorherige Evaluation eines angepassten Überlagerungskonzepts eine verbesserte Planung hinsichtlich der Automatisierung und eine Wahrnehmung möglicher Wechselwirkungen, wie eine eingeschränkte Präzision der Überlagerung, bewirken. Der Vergleich von Überlagerungskonzepten beschränkt sich jedoch nicht ausschließlich auf die genannten Einsatzgebiete, was in folgendem Abschnitt durch die Beschreibung verwandter Arbeiten ersichtlich wird.

3.4 Verwandte Arbeiten

Im Forschungsbereich der Augmented Reality unter dem Stichwort Registration existieren unterschiedliche Veröffentlichungen welche dieser Arbeit ähneln und als Kernaspekt einen Vergleich bestehender Konzepte oder Technologien anstreben.

Ein Beispiel einer Vergleichsstudie stammt von Ceseracciu et al. [CSC14] und behandelt den Vergleich Marker-basierter und markerloser Bewegungserfassungstechnologie mit einem Anwendungsbereich innerhalb der Medizin. Der Aufbau besteht aus acht unterschiedlich positionierten Kameras im Raum, welche so ausgerichtet sind, dass sie in Paaren auf je einen Marker an der Wand zeigen. Eine Versuchsperson läuft in mehreren Durchläufen eine vorgegebene Route entlang des Raumes. Dabei wurden die Winkelgelenke der unteren Extremitäten der Versuchsperson gemessen und die Abweichung der vom Algorithmus erkannten Werte und der tatsächlichen Werte festgehalten. Im ersten Durchlauf wird dabei ein markerloser Ansatz genutzt, bei dem ein Iterative Closest Point-Algorithmus [BM92] zum Einsatz kommt. Im zweiten Durchlauf wird ein Marker-basierter Ansatz nach Leardini et al. [LSP+07] angewendet, bei dem sowohl mehrere reflektierende Marker an der Versuchsperson angebracht werden, als auch in der Umgebung. Die Auswertung der Studie ergab, dass ein markerloses System der etablierten Markertechnologie im Hinblick auf die Präzision unterliegt. Dies wurde insbesondere durch die Prämisse gestützt, dass das System im medizinischen Kontext eingesetzt werden soll.

Dieser Studie ähnlich ist ein Experiment von Patrizi et al. [PPV16], welche eine markerlose und eine Marker-basierte Bewegungserfassungstechnologie gegenüberstellt. Dabei basiert die markerlose Technologie auf der kostengünstigen Microsoft Kinect⁶, während für das Marker-basierte System das high-end Produkt BTS SMART⁷ zum Einsatz kommt. Der Versuch fokussiert sich bei der Studie auf die Erkennung der Bewegungen von Personen bei typischen industriellen Arbeitsprozessen und wurde mit fünf Probanden durchgeführt. Ziel des Versuchsaufbaus ist in diesem Fall das Erkennen potenziell gesundheitsschädlicher Bewegungsabläufe. Die Ergebnisse der Versuchsreihe zeigten, dass die Erkennung der markerlosen Bewegungserfassungstechnologie der Erkennung der Marker-basierten annähernd ähnelte. Eine Verbesserung der Erkennung mit Hilfe der markerlosen Technologie konnte hergestellt werden, indem ein Filter starke Abweichungen aus der Sensormessung entfernt.

⁶<https://developer.microsoft.com/en-us/windows/kinect>

⁷<https://www.btsbioengineering.com/products/smart-dx>

Eine Vergleichsstudie von Amin und Govilkar [AG15] stellt mehrere Augmented Reality SDKs, darunter auch Vuforia und ARToolKit, gegenüber. Als Vergleichsgrundlage wurde dabei zunächst die Lizenzart des jeweiligen SDKs und die unterstützten Plattformen gewählt. Des Weiteren wurde die Funktionalität der SDKs geprüft, darunter die Möglichkeit der Erstellung von Markern und die Tracking-Fähigkeiten. Ebenso wurde verglichen, ob das SDK die Möglichkeit der Überlagerung von virtuellen Objekten erlaubt und wie groß dabei der Funktionsumfang ist. So wurde ausgewertet, ob sowohl 3D-, als auch 2D-Objekte überlagert werden können. Schließlich wurden besondere Vorzüge und Einschränkungen der SDKs festgehalten. Die Autoren stellten in der Vergleichsstudie lediglich die Ergebnisse vor, eine Rangordnung oder Diskussion über die Vergleichsergebnisse wurde nicht durchgeführt.

Die Studien von Ceseracciu et al. [CSC14] und Patrizi et al. [PPV16] legten den alleinigen Fokus auf das Anwendungsgebiet der Medizin. Dabei wurden ausschließlich stationäre Tracking-Systeme verwendet, die Überprüfung von mobilen Geräten, wie Datenbrillen oder Smartphones, wurde nicht durchgeführt. Die Vergleichsstudie von Amin und Govilkar [AG15] führte einen gezielten Vergleich der Technologien im Bereich der Mixed Reality durch. Deren Qualität unter dem Einsatz von unterschiedlichen Konzepten wurde dabei nicht festgestellt. Stattdessen bietet die Studie eine Übersicht der Gesamtfunktionalität aller überprüfter SDKs. Um vertieftes Wissen über die Funktionen der SDKs zu erhalten, müsste ein Funktionsvergleich durchgeführt und evaluiert werden.

Durch die Technologien und Forschungsergebnisse, die in diesem Kapitel vorgestellt wurden, ist es in Kapitel 4 möglich einen Versuch zu entwerfen, der unterschiedliche Überlagerungskonzepte auf Basis von Automatisierungsstufen vergleicht. Dazu wird auf die technischen Umsetzungen aus diesem Kapitel zurückgegriffen. Da der Fokus auf den Anwendungsfall in der Industrie liegt, fließen die Erkenntnisse über mögliche Einsatzgebiete, speziell den industriellen, aus diesem Kapitel mit in den Entwurf des Versuchsaufbau ein. Ferner soll im folgenden Kapitel vermittelt werden, wie sich diese Arbeit in die bestehenden Werke einbettet und was sie von den verwandten Arbeiten abgrenzt.

4 Untersuchungsaufbau und -durchführung

Die in dieser Arbeit durchgeführte Untersuchung von Überlagerungskonzepten (vgl. Vorgang einer Überlagerung in Abbildung 1.1) soll ein möglichst breites Spektrum an Konzepten vergleichen, die sich für eine Überlagerung von Entitäten in der Mixed Reality eignen. Dabei sollen die Überlagerungskonzepte möglichst unterschiedliche Techniken anwenden, um das gewünschte Resultat der korrekten Überlagerung zu erreichen. Je mehr Ansätze mit in den Vergleich einbezogen werden, desto mehr Auswahlmöglichkeiten existieren bei der Konzeptwahl in der Prozessintegration. Bezüglich des Prozesses ist einzugrenzen, dass der Versuch auf dem Einsatzgebiet der Produktion basiert. Dazu wird bewusst ein Testobjekt für den Versuchs verwendet, das sich nach Kruijff et al. [KSF10] schlecht für einen Überlagerungsvorgang eignet, jedoch einer typischen Entität in dem Anwendungsfall entspricht. Auch die Messgrößen werden mit dem Fokus auf die Produktion ausgewählt, wobei insbesondere der Vergleich auf Grundlage der Automatisierungsstufen hervorzuheben ist.

Basierend auf den in Kapitel 2 zusammengefassten Grundlagen und der Einbeziehung des aktuellen Stands der Technik (vgl. Kapitel 3), beschreibt dieses Kapitel alle Bedingungen, Konzepte, Objekte und Messungen, welche für den Vergleich der Überlagerungstechniken notwendig sind. Dazu werden die in dem Vergleich berücksichtigten Überlagerungskonzepte zunächst vorgestellt und deren technischer Vertreter benannt (vgl. Abschnitt 4.1). Zudem werden Metriken definiert, die Messgrößen auf eine Skala zur Bewertung der Eigenschaften abbildet (vgl. Abschnitt 4.2). Aufbauend auf die so definierten Vergleichsobjekte und Metriken wird das Vorgehen strukturiert, indem weitere grundlegende Rahmenbedingungen festgelegt werden (vgl. Abschnitt 4.3).

4.1 Beschreibung der Überlagerungskonzepte

Für den Vergleich verschiedener Überlagerungstechniken wurden drei Konzepte erarbeitet. Zwei der Konzepte wurden bereits durch die Zusammenfassung der Grundlagen in Kapitel 2 erkannt. Dabei handelt es sich zum einen um das Marker-basierte Überlagerungskonzept (vgl. Abschnitt 4.1.2) und zum anderen um das Modell-basierte Konzept (vgl. Abschnitt 4.1.3). Ein dritter Ansatz wird zudem in Abschnitt 4.1.1 vorgestellt. Die Wahl der Überlagerungskonzepte begründet sich sowohl in einer ausreichend hohe Varianz untereinander als auch durch das Bestehen technischer Vertreter, die frei zugänglich sind. Eine Beschreibung der Konzepte und der technischen Vertreter wird im Folgenden vorgestellt.

4.1.1 Manuelle Positionierung

Das Überlagern von Elementen auf einer Projektionsfläche mittels Zeigegeräten wird als *Drag and Drop* bezeichnet. Ähnlich diesem Verfahren, ist es mittels Gestensteuerung oder tragbaren Eingabegeräten möglich, virtuelle Elemente im Sichtfeld einer Datenbrille zu verschieben. Beispielsweise führt der Benutzer eine Verschiebung mittels Gestensteuerung durch, indem er seine Hand vor die Sensoren der Datenbrille bewegt und Zeigefinger und Daumen so zusammenführt, als würde er einen physischen Gegenstand in der Luft greifen. Das virtuelle Element folgt den Bewegungen der Hand nun so lange, bis der Benutzer Zeigefinger und Daumen wieder auseinanderführt oder die Sensoren der Datenbrille die Handbewegung nicht mehr aufnehmen können. Um weitere Objektmanipulationen zu ermöglichen, kann der Benutzer das Objekt zusätzlich mit der zweiten Hand greifen und die Hände auseinander bzw. zueinander bewegen, um das Objekt zu skalieren, oder sie kreisförmig bewegen, um eine Rotation durchzuführen. Mittels der räumlichen Berechnungen der Datenbrille ist es daraufhin möglich das Objekt der physischen Entität so zu überlagern, dass es nicht mit dem Sichtfeld des Benutzers bewegt wird, sondern an der ursprünglichen Position verbleibt; der Benutzer kann das virtuelle Objekt nun von allen Seiten aus betrachten.

Das in Abschnitt 3.2 vorgestellte SDK von Microsoft, welches in die Laufzeit- und Entwicklungsumgebung Unity¹ integrierbar ist, stellt als Teil seiner Eingabekomponente die Möglichkeit der Manipulation von Hologrammen zur Verfügung, was die Skalierung, Rotation und Translation mittels Gestensteuerung beinhaltet. Aufgrund der einfachen und nativen Einsatzfähigkeit auf der HoloLens-Datenbrille, welche als Versuchsgerät zum Einsatz kommt, wird diese technische Umsetzung als Vertreter des Konzepts der manuellen Positionierung gewählt. Um die Gestensteuerung auf einem vorgegebenen Modell zu aktivieren, wird diesem ein „Manipulation Handler“ als Komponente hinzugefügt (vgl. Einbettung der Komponente in die Entwicklungsplattform Unity, Abbildung 4.1). Dabei können unterschiedliche Parameter gewählt werden, die Einfluss auf die Steuerung des Modells nehmen: Beispielsweise können mehrere Kombinationen aus Rotation, Skalierung und Translation als Manipulationsgrad gewählt werden oder der Einsatz von einer bzw. beiden Hände festgelegt werden. Der Collider (in Abbildung 4.1 grün umkreist) des Modells legt dabei den Bereich fest, der als Auslöser für eine Manipulationsgeste verwendet wird.

4.1.2 Markerpositionierung

Das virtuelle Überlagern mittels Referenzmarker basiert auf den Konzepten und Technologien, die in Abschnitt 2.3.2 und Abschnitt 3.1 vorgestellt wurden. Virtuelle Objekte werden dabei mit den Referenzmarkern eines Markersystems über eine Software oder Onlineplattform verknüpft. Dazu müssen die verwendeten Referenzmarker zunächst automatisch oder manuell generiert werden. Die gedruckten Referenzmarker werden daraufhin an die physische Entität angebracht, sodass die Sensoren der Datenbrille diese möglichst verdeckungsfrei erkennen können. Mittels Objekterkennungsalgorithmen (vgl. Abschnitt 2.3) können die Marker daraufhin im Sichtfeld der Datenbrille erkannt werden und die virtuelle Entität relativ dazu eingeblendet werden. Solange der Referenzmarker im Sichtfeld des Benutzers sichtbar ist, verbleibt das virtuelle Objekt daraufhin der physischen Entität überlagert und unbeeinflusst von der Kopfbewegung des Benutzers.

¹<https://unity3d.com>

Technische Realisierungen von Markersystemen werden von mehreren Entwicklern bereitgestellt (vgl. Querschnitt in Abschnitt 3.1). Das SDK Vuforia von PTC Inc. wurde bereits in unterschiedlichen Veröffentlichungen als Grundlage der Markererkennung eingesetzt und erprobt [XL14][LCWN15][PZ17]. Zusätzlich ist das SDK seit 2017 in der Laufzeit- und Entwicklungsumgebung Unity enthalten [Aug17] und bietet explizite Konfigurationen für den Einsatz mit der HoloLens-Datenbrille. Vuforia stellt zudem eine Vielzahl an Features bereit und unterstützt mehrere Plattformen, vergleicht man das SDK mit seinen Alternativen (vgl. Abschnitt 3.4). Vuforia überwiegt somit den übrigen technischen Realisierungen der Markerpositionierung aus Abschnitt 3.1 und wird als Vertreter des Konzepts für diesen Versuch gewählt. Abbildung 4.2 zeigt den für den Vergleich entworfenen Referenzmarker. Dieser wurde nach dem Leitfaden² des Vuforia SDKs gestaltet. Der erstellte Marker kann mit Hilfe des SDKs in die Entwicklungsumgebung Unity importiert werden und um ein Modell ergänzt werden, das, abhängig von der Markerposition, bei einer Erkennung des Markers automatisch eingeblendet wird.

4.1.3 Modellbasierte Positionierung

Eine weitere Möglichkeit der Überlagerung physischer Entitäten besteht durch das Erkennen des Objekts im Raum mit Hilfe eines Abgleichs der Oberflächenstruktur und -textur von identischen, virtuell hinterlegten 3D-Modellen. Diese müssen zunächst mittels eines 3D-Computergrafik- und Animationsprogramms als Grundlage dieser Methode erstellt werden. Um eine optimale Erkennung zu ermöglichen, muss das virtuelle Modell der physischen Entität in Form, Textur und Dimension maximal ähneln. Da das virtuelle Modell als Grundlage der Überlagerung dient, besteht zudem die Notwendigkeit einer Optimierung durch eine Reduktion der vorhandenen Polygone; die Kapazität der

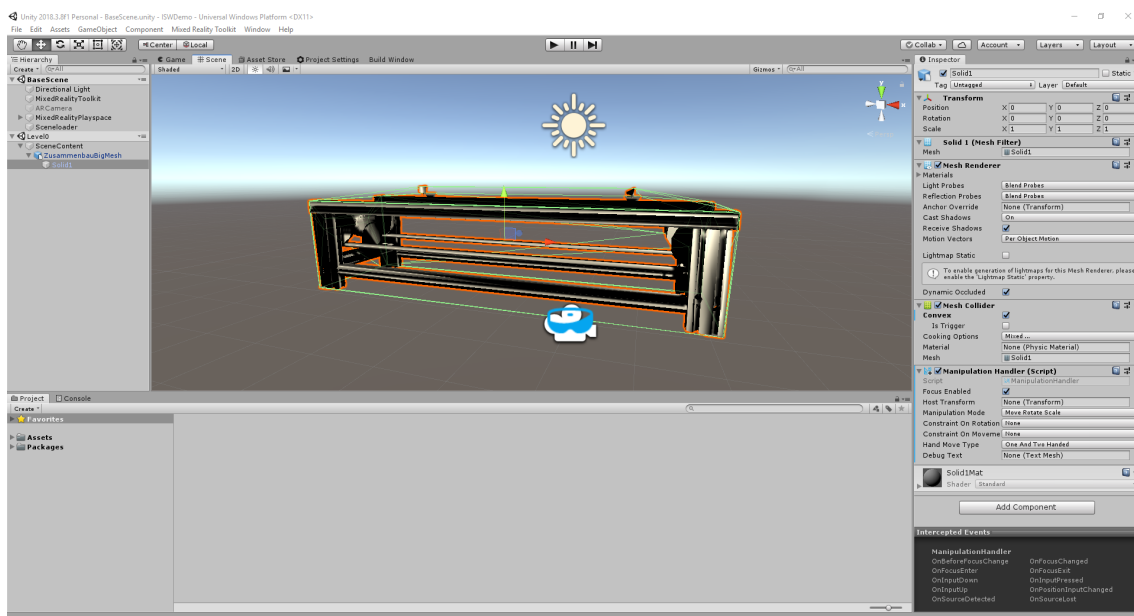


Abbildung 4.1: Technische Umsetzung der manuellen Positionierung

²<https://library.vuforia.com/articles/Solution/Optimizing-Target-Detection-and-Tracking-Stability>

Verarbeitungsalgorithmen und die Leistung der Datenbrillen sind Ressourcen-bedingt limitiert. Steht dem Benutzer kein 3D-Modell des Objekts zur Verfügung, besitzen einige technische Realisierungen einen Objekt-Scanner, der es erlaubt, sensorische Aufnahmen von unterschiedlichen Positionen aus zu erfassen und somit ein virtuelles 3D-Abbild der physischen Entität zu erzeugen. Auf Grundlage dieses Modells ist es daraufhin möglich, ein Objekt in der reellen Umgebung zu überlagern, falls es eine annähernd ähnliche geometrische Form und Oberflächentextur wie das virtuelle Modell besitzt. Dafür muss der Benutzer das Sichtfeld so auf die Entität ausrichten, dass die Ansicht des Objekts einer hinterlegten Ansicht des virtuellen Modells gleicht. Als Hilfestellung bieten einige technische Vertreter dabei eine 2D- oder 3D-Einblendung des Modells im Sichtfeld des Benutzers an. Objekterkennungsalgorithmen (vgl. Abschnitt 2.3) suchen die Umgebung kontinuierlich nach einer physischen Entität ab, die dem virtuellen Modell ähnelt. Wurde die Entität mit einer ausreichend hohen Sicherheit erfasst, wird das virtuelle Modell an dessen Stelle überlagert. Solange die Erkennung des Algorithmus erfolgreich ist, kann sich der Benutzer daraufhin im Raum bewegen und das virtuelle Objekt von unterschiedlichen Seiten aus relativ zu der physischen Entität betrachten.

Das in der Laufzeit- und Entwicklungsumgebung Unity enthaltene SDK Vuforia bietet eine technische Realisierung dieses Konzepts durch das Erfassen eines bereits erstellten 3D-Modells mit Hilfe einer eigenständigen Software (vgl. Abbildung 4.3) [PTC19]. Dazu muss das 3D-Modell zunächst importiert und daraufhin von mehreren Blickwinkeln aus erfasst werden, um den Algorithmus zur automatischen Erkennung zu trainieren. Die erstellte Datenbank kann daraufhin in Unity importiert werden und als Grundlage für eine *Model Target-Komponente* verwendet werden.

Weitere Versuchsobjekte, wie beispielsweise Mischformen der drei vorgestellten Konzepte oder technisch schwer bzw. nicht realisierbare Konzepte, wurden hier vernachlässigt. Sie können jedoch als Motivation für weiterführende Arbeiten dienen (vgl. Abschnitt 6.2). Die Umsetzungen der

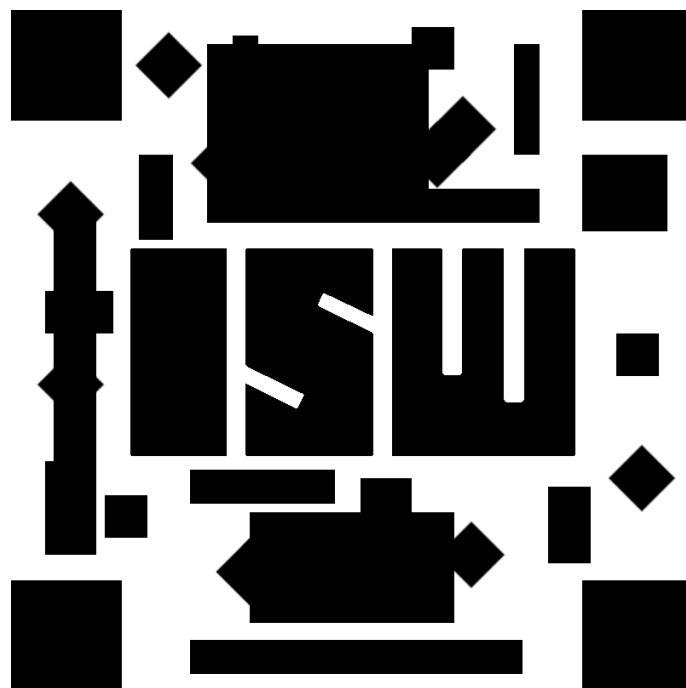


Abbildung 4.2: Referenzmarker des Versuchsaufbaus

Überlagerungskonzepte behandeln ausschließlich die Objekterkennung in einer dreidimensionalen Umgebung, auch wenn einige Konzepte auf die zweidimensionale Objekterkennung übertragbar sind. Die Beispiel-Programmierung der Konzepte beschränkt sich auf eine Benutzung mit der Datenbrille Microsoft HoloLens; weitere Mixed Reality-Geräte, wie Smartphones, werden nicht berücksichtigt. Die entstandenen Projektdateien basieren, wie bereits vorgestellt, auf der Spiele-Engine Unity, die eine Programmierung auf der Datenbrille ermöglicht; Die technischen Umsetzungen sind somit für den Einsatz auf ebenjener optimiert. Zwar erlaubt die Engine die Portierung der Programmierung auf unterschiedliche Plattformen, die Implementierungen dieser Arbeit sind jedoch exklusiv für den Einsatz auf der Universal Windows Platform (UWP) [RSW+18] konzipiert. Um Eigenschaften der definierten Überlagerungskonzepte quantifiziert messbar zu gestalten, müssen diese, dem Anwendungsfall in der Produktion entsprechend, ausgewählt und um eine Metrik ergänzt werden.

4.2 Messgrößen und Metriken

Als Hauptbestandteil der Messgrößen werden die für die generelle Mensch-Computer Interaktion entworfenen Automatisierungsgrade nach Parasuraman et al. [PSW00] (vgl. Abschnitt 2.1) im Folgenden dem Anwendungsfall entsprechend abgewandelt. Dies bedeutet insbesondere, die einzelnen Phasen auf die hard- und softwaregestützte Objekterkennung physischer Entitäten und der Überlagerung digitaler Abbildungen zu beschränken.

- **Informationsaufnahme:** Die Automatisierung der Informationsverarbeitung bezieht sich auf die Vorverarbeitung der Umgebung, der Erkennungssoftware und der virtuellen Modelle. So führt das notwendige Eingreifen des Benutzers zum Platzen von zusätzlichen Objekten

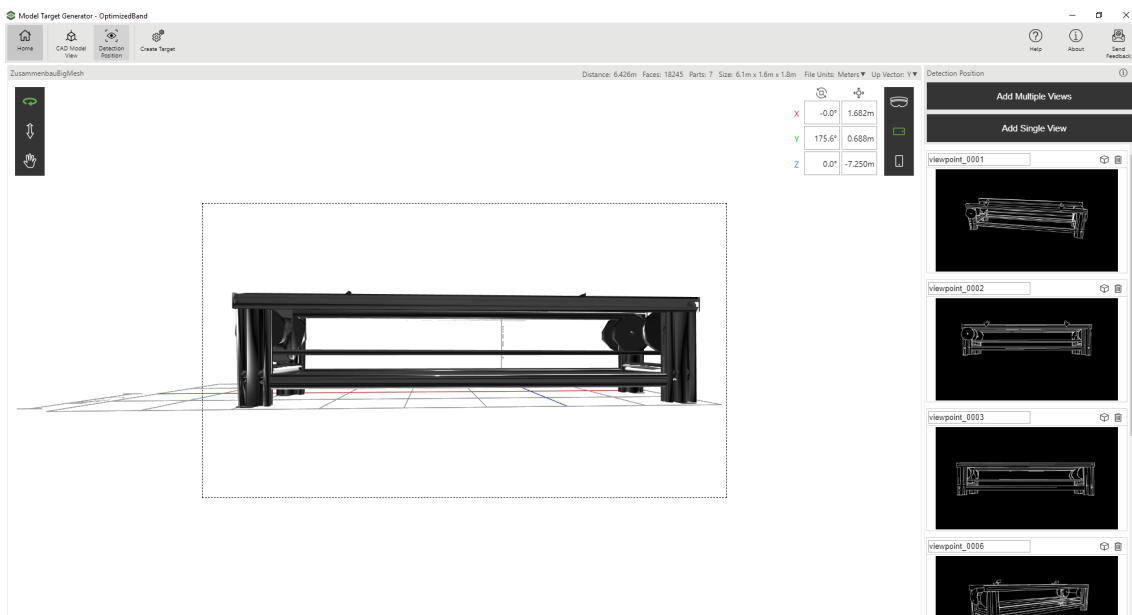


Abbildung 4.3: Technische Umsetzung der modellbasierten Positionierung






im Raum oder die manuelle Konfiguration der eingesetzten Software zu einer Minderung der Bewertung dieser Phase. Ebenso wird das Ergebnis negativ beeinflusst, falls virtuelle Abbildungen nicht automatisch erstellt oder manipuliert werden.

- **Informationsverarbeitung:** Die Bewertung der Informationsverarbeitungsphase richtet sich nach dem Grad der Automatisierung des Überlagerungskonzepts bezüglich der selbstständigen Erkennung von definierten physischen Entitäten. Je mehr Eingriffe durch den Benutzer notwendig sind, um eine Entität zu erkennen, desto geringer fällt die Bewertung aus. Ein Überlagerungskonzept, das keinen Algorithmus zur Erkennung einsetzt, erhält die niedrigste Bewertung.
- **Entscheidungsfindung:** Der Erkennung des Objekts folgt die Wahl der virtuellen Entität. Die Bewertung dieser Phase bemisst sich nach der Fähigkeit eines Konzepts, für eine erkannte physische Entität automatisch das dem Kontext entsprechend korrekte Modell auszuwählen. Ein Algorithmus, der ohne Benutzereingriff das gewünschte Modell auswählt, erhält somit eine höhere Bewertung als ein Algorithmus, der nur ein festgelegtes Modell auswählt oder bei dem ein Benutzereingriff notwendig ist.
- **Ausführung der Aktion:** Die Ausführung der Aktion bezieht sich auf die tatsächliche Überlagerung der physischen und virtuellen Entität. Der Grad der Automatisierung misst sich in diesem Fall daran, wie ausgeprägt der Benutzer in die Ausführung involviert wird. Muss der Benutzer die virtuelle Entität selbstständig an den Ort der physischen Entität bewegen, liegt keine Automatisierung der Phase vor. Bewegt ein Algorithmus hingegen das Objekt an die entsprechende Position, richtet sie die Bewertung danach, wie sehr der Benutzer in diesen Vorgang involviert ist.

Um die so definierten Phasen des Überlagerungsvorgangs nach einer Skala auf den Grad der Automatisierung hin bewerten zu können, wird die Einteilung nach Sheridan und Verplank [SV78] (vgl. Abschnitt 2.1) angewendet. Die Metrik überträgt die Automatisierungsgrade auf eine Skala von Kreisdiagrammen in fünf Abstufungen („*Harvey Balls*“) [Roa06]. Somit steht der Wert 1 (am kleinsten) für keine Füllung, der Wert 2 für eine viertel Füllung, der Wert 3 für eine halbe Füllung, der Wert 4 für eine dreiviertel Füllung und der Wert 5 (am größten) für eine komplette Füllung des Harvey Balls. Tabelle 4.1 zeigt die definierte Metrik der Automatisierungsstufen.

Da die Überlagerungskonzepte unterschiedliche Vorgehensweisen verfolgen, können weitere Eigenschaften gemessen werden. Insbesondere jene Eigenschaften sind hierbei von Interesse, die in der Produktion von hoher Relevanz sein können. So müssen Teile der Produktion beispielsweise möglichst präzise überlagert werden, um Fehler zu vermeiden. Zu diesem Zweck wird die Überlagerungsgenauigkeit gemessen. Da die Teile häufig an unterschiedlichen Stationen bearbeitet werden, könnten sie sich in Bewegung befinden, weswegen der Bedarf bestünde, dass ein virtuelles Objekt auf der physischen Entität verbleibt, wurde es initial auf diese projiziert. Dazu wird die Stabilität der Überlagerung gemessen. Je nach Anwendungsfall kann es von Relevanz sein, wie hoch der Ressourcen- oder Zeitverbrauch der jeweiligen Überlagerungstechnik ist. Aus diesem Grund werden diese beiden Messgrößen ebenfalls in den Vergleich aufgenommen. Die so ergründeten Eigenschaften werden im folgenden beschrieben und um eine Metrik ergänzt, die ebenfalls auf Grundlage von Harvey Balls entworfen wird.

Tabelle 4.1: Zuweisung der Automatisierungsstufen

Automatisierungsstufe	Skala
Stufe 1 & 2	
Stufe 3 & 4	
Stufe 5 & 6	
Stufe 7 & 8	
Stufe 9 & 10	

- **Überlagerungsgenauigkeit:** Die Überlagerungsgenauigkeit liefert Rückschlüsse über das Maß an Präzision, mit der eine virtuelle Komponente der physischen Entität überlagert wird. Dazu wird die Abweichung zwischen der virtuellen Überlagerung und der physischen Entität, ausgehend von vier Fixpunkten, gemessen. Für das Versuchsobjekt werden die vier Ecken der beiden oberen Führungsstangen verwendet; das abschließende Ende links hinten, links vorne, rechts hinten und rechts vorne. Für jeden Fixpunkt wird der Abstand zwischen dem virtuellen und dem physischen Punkt in Millimetern gemessen, indem eine Aufnahme des Sichtfelds durch die Datenbrille ausgewertet wird. Die Möglichkeit der Aufnahme wird von der Datenbrille HoloLens bereitgestellt und liefert ein Bild der Kamerasensoren und der derzeit eingeblendeten Hologramme. Der Durchschnitt dieser vier Werte liefert daraufhin einen Vergleichswert. Die Vergleichswerte aller vier Positionen werden schließlich erneut verwendet, um den Durchschnitt zu errechnen, der die finale Abweichung darstellt.
- **Stabilität:** Die Stabilität der Überlagerungstechniken bemisst sich an der Präzision, mit der eine bereits überlagerte virtuelle Abbildung auf der physischen Entität verweilt, wenn man diese aus unterschiedlichen Winkeln und Entfernungen betrachtet. Dazu wird die Datenbrille nach der Erkennung in der ursprünglichen Position auf die übrigen Positionen bewegt und jeweils eine Aufnahme des Sichtfelds erstellt. Tabelle 4.3 zeigt die Zuweisung der durchschnittlichen Abweichung und der Harvey Balls.
- **Ressourcenverbrauch:** Die benötigte Menge an Ressourcen, wie die Verwendung des Arbeitsspeichers oder der CPU, ist für den Einsatz auf den Datenbrillen der aktuellen Generation von gesondertem Interesse, da die Gesamtmenge an verfügbaren Ressourcen auf diesen Geräten vergleichsweise beschränkt ist. Der tatsächliche Verbrauch von Ressourcen unter Ausführung der jeweiligen Überlagerungstechnik, wird mit Hilfe eines Entwicklerportals gemessen, das eine kontinuierliche Ressourcenübersicht der eingesetzten Datenbrille HoloLens bereitstellt. Gemessen wird der Durchschnitt der CPU- und Arbeitsspeicher-Auslastung während der Ausführung der jeweiligen Überlagerungstechnik. Für die Ergebnisse von CPU und Arbeitsspeicher wird der Mittelwert auf zwei Nachkommastellen genau bestimmt, wobei

Tabelle 4.2: Zuweisung der Überlagerungsgenauigkeit











Durchschnittliche Abweichung	Skala
> 6 Millimeter	
4-6 Millimeter	
2-4 Millimeter	
1-2 Millimeter	
< 1 Millimeter	

Tabelle 4.3: Zuweisung der Stabilität

Durchschnittliche Abweichung	Skala
> 20 Millimeter	
15-20 Millimeter	
10-15 Millimeter	
5-10 Millimeter	
< 5 Millimeter	

stets aufgerundet wird. Tabelle 4.4 zeigt die Werte, welche für die durchschnittliche Ressourcenauslastungen der CPU verwendet werden. Tabelle 4.5 zeigt die Werte, welche für die durchschnittliche Ressourcenauslastungen des Arbeitsspeichers verwendet werden.

- **Zeitaufwand:** Das Messen der benötigten Zeit zum Erkennen und Überlagern einer Entität liefert keine Erkenntnis über den Zeitaufwand zur Vorbereitung der Umgebung und der virtuellen Modelle. Zu diesem Zweck wird die Vorbereitungszeit einer jeden Überlagerungstechnik gemessen. Da die Erstellung des CAD-Modells für jede Technik notwendig ist, wird die dafür aufgewendete Zeit nicht mit eingerechnet. Tabelle 4.6 zeigt die Metrik der Gesamtzeit und der Abbildung mit Hilfe der Harvey Balls.

Tabelle 4.4: Zuweisung des Ressourcenverbrauchs (CPU)
















Durchschnittliche Ressourcenauslastung	Skala
80-100%	
60-80%	
40-60%	
20-40%	
0-20%	

Tabelle 4.5: Zuweisung des Ressourcenverbrauchs (Arbeitsspeicher)

Durchschnittliche Ressourcenauslastung	Skala
80-100%	
60-80%	
40-60%	
20-40%	
0-20%	

Mit Hilfe der Implementierungen der drei Überlagerungskonzepte und der Festlegung von Metriken zur Einteilung der Ansätze ist es im folgenden Arbeitsschritt möglich, einen Versuchsaufbau zu entwerfen, der diese Umsetzungen gegenüberstellt und die Messung von Automatisierungsstufen (vgl. Abschnitt 2.1) erlaubt, sowie die korrelierenden Eigenschaften untersucht.

Tabelle 4.6: Zuweisung des Zeitaufwands

Vorbereitungszeit	Skala
> 60 Minuten	
45-60 Minuten	
30-45 Minuten	
15-30 Minuten	
< 15 Minuten	

4.3 Vorgehen

Um einen quantifizierbaren, reproduzierbaren Versuch zu gewährleisten, muss das Vorgehen sorgfältig strukturiert werden. Der Versuchsaufbau muss reproduzierbar gestaltet werden und die Möglichkeit der Auswertung der Daten sichergestellt werden. Der folgende Abschnitt beschreibt dieses Vorgehen, indem die zu vollziehenden Ausführungsschritte und der Aufbau des Versuchs beschrieben werden.

Da das Einsatzgebiet der Überlagerungskonzepte in dieser Arbeit mit dem Schwerpunkt auf Anlagen in der Produktion ausgerichtet ist, wird als physische Entität, die jede Überlagerungstechnik erkennen und um ihr virtuelles Modell ergänzen soll, ein Segment eines Förderbands zum Transport von Einzelteilen verwendet (vgl. Abbildung 4.4). Das Segment besteht überwiegend aus reflektierendem Metall. Dies erschwert die Objekterkennung zwar erheblich³, im Umfeld des Anwendungsfalls der Produktion stellt dies jedoch eine gängige Beschaffenheit dar, was das Versuchsobjekt trotz erschwerter Randbedingungen als geeignet qualifiziert.

Um falsch-positive und falsch-negative Erkennungsfehler zu minimieren, wird die Umgebung des Versuchsobjekts farblich vereinheitlicht und von Reflektionsquellen befreit. Als Lichtquelle wird künstliches Licht verwendet, das an der Decke über dem Versuchsobjekt angebracht ist.

Der Überlagerungsvorgang einer jeden Überlagerungstechnik wird aus vier verschiedenen Positionen wiederholt. Dabei unterscheiden sich die Positionen in der Entfernung der Datenbrille zum Versuchsobjekt und der Perspektive auf der vertikalen Achse. Somit können mögliche Fehlerquellen ausgeschlossen werden, die bedingt durch einen bestimmten Betrachtungswinkel oder eine gewisse Entfernung zum Objekt entstehen können. Ebenso wird durch eine unterschiedliche Positionierung ein klareres Ergebnis der Qualität einer jeweiligen Überlagerungstechnik erzielt, da mögliche Schwächen erst aus gewissen Winkeln oder Entfernungen ersichtlich werden. Schließlich dient

³<https://library.vuforia.com/articles/Solution/Optimizing-Target-Detection-and-Tracking-Stability>

eine Definition mehrerer Positionen der Messung der Stabilität: Wurde eine initiale Überlagerung vollzogen, wird die Datenbrille an die übrigen Positionen bewegt, um das Verhalten der Überlagerung unter Bewegung zu bemessen. Folgenden Positionen werden für diesen Versuch ausgewählt:

- **Front:** Das Versuchsobjekt wird aus einer frontalen Position in einer Entfernung von einem Meter betrachtet.
- **Seite:** Das Versuchsobjekt wird von der rechten Seite aus einer Entfernung von einem Meter in einer Höhe von 1,80 Metern aus betrachtet. Eine zusätzliche Messung der linken Seite entfällt unter der Annahme, dass die Positionen ein gleiches Verhalten der Überlagerungstechniken bewirken.
- **Nah:** Das Versuchsobjekt wird in einer Entfernung von einem halben Meter aus einer Höhe von 1,80 Metern aus betrachtet.
- **Fern:** Das Versuchsobjekt wird in einer Entfernung von 2 Metern aus einer Höhe von 1,80 Metern aus betrachtet.

Bei der Wahl der Positionen wurde auf eine möglichst große Varianz der Winkel und Entfernungen geachtet. Der Versuchsaufbau ist in Abbildung 4.4 zu sehen. Die Markierungen im unteren Bereich der Abbildung zeigen die Position Nah und Front.

Die Versuchsreihe wird in elf Phasen untergliedert, welche für jede Überlagerungstechnik und jede der vier Positionen wiederholt werden. Somit wird sichergestellt, dass jede Messung gleich vollzogen wird und keine Messfehler aufgrund einer unterschiedlichen Reihenfolge entstehen. Ebenso dient die exakte Definition des Vorgehens der Reproduzierbarkeit des Versuchs. Folgende Phasen werden für alle Überlagerungstechniken in jeder Position wiederholt, was eine Gesamtzahl von zwölf Durchläufen ergibt:

1. **Starten der Zeitmessung:** Zur Berechnung des gesamten Zeitaufwands für die Vorbereitung der Überlagerungstechnik und der Umgebung wird die Dauer gemessen.
2. **Überlagerungstechnikspezifische Vorbereitungen:** Überlagerungstechniken, die eine gesonderte Vorbereitung benötigten werden in dieser Phase für den Versuch präpariert. Falls es sich bei der Vorbereitung um einen einmaligen Vorgang handelt, der für die übrigen Positionen wiederverwendet werden kann, wird die benötigte Zeit nur für diesen einen Vorgang gemessen.
3. **Vorbereiten der Umgebung:** Falls Überlagerungstechniken eine Manipulation der Umgebung erfordern, werden sämtliche Vorbereitungen in dieser Phase vollzogen. Muss die Umgebung nur ein Mal angepasst werden, um die Überlagerungstechnik von allen Positionen aus auszuführen, wird die Zeit ebenfalls nur für diesen einen Vorgang gemessen.
4. **Stoppen der Zeitmessung:** Sind alle Vorbereitungen abgeschlossen wird die Zeit gestoppt und dokumentiert.
5. **Positionierung der Datenbrille:** In dieser Phase wird die Datenbrille an die aktuell zu messende Position bewegt und in den Ausgangszustand zurückgesetzt. Dies bedeutet, dass alle laufenden Applikationen und angezeigte Hologramme entfernt werden, um die Messung für jede Wiederholung unverfälscht durchführen zu können.

6. **Starten des Performance Tracing:** Die HoloLens-Datenbrille erlaubt eine umfangreiche Aufzeichnung ausgeführter Prozesse inklusive des Ressourcenverbrauchs. Die Ablaufverfolgung kann mit Hilfe einer Web-Plattform gestartet und gestoppt werden. Um einen vollständigen Ablauf zu erhalten wird das Performance Tracing bereits vor der Ausführung der Applikation gestartet.
7. **Starten der Testapplikation:** Die Testapplikation wird mit der zu messenden Überlagerungstechnik auf der HoloLens-Datenbrille ausgeführt.
8. **Durchführung der Erkennung und Überlagerung:** Die Applikation wird dreißig Sekunden lang ausgeführt, um der Berechnung der Algorithmen genügend Zeit einzuräumen.

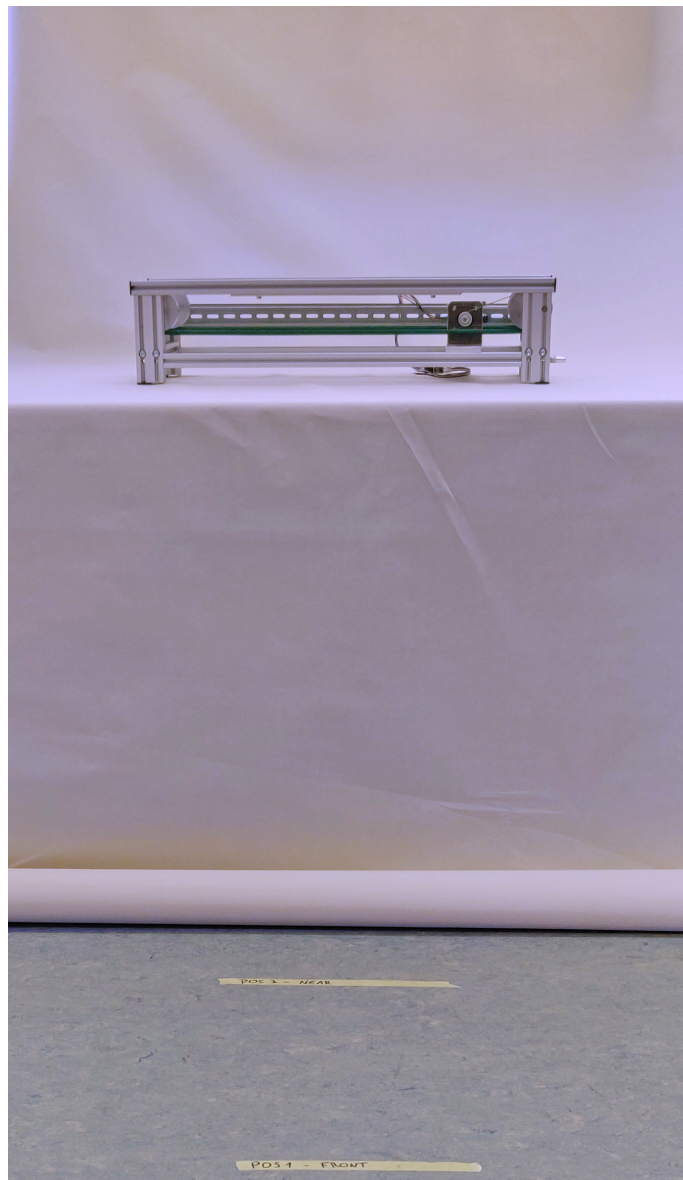


Abbildung 4.4: Aufnahme des Versuchsaufbaus

9. **Stoppen des Performance Tracing:** Die Ablaufverfolgung wird über die Web-Plattform nach Ablauf der dreißig Sekunden gestoppt.
10. **Aufnahme der Szenerie zur Bestimmung der Überlagerungsgenauigkeit:** Die Aufnahmefunktion der HoloLens-Datenbrille wird in dieser Phase dazu genutzt, ein Bild von der physischen Entität und der virtuellen Überlagerung aufzunehmen. Diese Aufnahme wird dazu genutzt, um die Abweichung der Fixpunkte von der physischen und der virtuellen Überlagerung zu messen.
11. **Aufnahmen der Szenerie aus den verbleibenden Positionen, zur Bestimmung der Stabilität:** Um die Stabilität der Überlagerungstechniken zu messen, wird die Datenbrille an die übrigen drei Positionen verschoben und jeweils ein Bild der Überlagerung aufgenommen. Diese Bilder werden ebenfalls dazu genutzt, die Abweichung der Fixpunkte von der physischen und der virtuellen Entität zu messen.

Mit Hilfe der Ergründung der zu messenden Überlagerungskonzepte, einer Einteilung der Messgrößen mit Hilfe von Metriken unter dem Einsatz von Harve Balls und der Beschreibung des Vorgehens sind alle Grundvoraussetzungen geschaffen, den angestrebten Versuch durchzuführen. Sämtliche Ergebnisse, die welche durch diesen entstanden, sind in Kapitel 5 zusammengefasst.

5 Evaluation

Das folgende Kapitel beinhaltet die Evaluation der Versuchsreihe, die in Kapitel 4 beschrieben wurde. Dazu werden die Ergebnisse der Messung jeder Messgröße einzeln vorgestellt (vgl. Automatisierung in der Informationsaufnahme in Abschnitt 5.1, Automatisierung in der Informationsverarbeitung in Abschnitt 5.2, Automatisierung bei der Entscheidungsfindung in Abschnitt 5.3, Automatisierung bei der Ausführung der Aktion in Abschnitt 5.4, Überlagerungsgenauigkeit in Abschnitt 5.5, Auslastung der CPU in Abschnitt 5.7, Nutzung von Arbeitsspeicher in Abschnitt 5.8, Stabilität der Überlagerung in Abschnitt 5.6 und benötigte Gesamtzeit in Abschnitt 5.9). Abschließend werden die Messergebnisse in Korrelation gestellt, um eine Gesamtübersicht herzustellen (vgl. Abschnitt 5.10).

5.1 Automatisierung Informationsaufnahme

Die zu überprüfenden Konzepte stellten in der Phase der Informationsaufnahme keine Form der Automatisierung bereit. Jede der drei Überlagerungstechniken musste jeweils eigenhändig in die Testapplikation eingepflegt werden. Dazu wurde für jede Technik ein entsprechendes 3D-Modell zur virtuellen Überlagerung in die Implementierung eingefügt. Das Marker-basierte Überlagerungskonzept erfordert darüber hinaus die Erstellung relevanter Marker und deren Verknüpfung mit den gewünschten 3D-Modellen. Bestehende Algorithmen benötigten eine Referenz, um die zu erkennenden Objekte im Raum zu erfassen. Die Auswahl einer Überlagerung folgert, dass in der Implementierungsphase das vorgesehene Modell eingefügt werden muss. Das so bereitgestellte Modell wird daraufhin für den Vorgang der Überlagerung als Referenz benutzt, eine nachträgliche Anpassung ist nur dann möglich, wenn auch weitere Modelle hinterlegt werden. Existieren jedoch mehrere Modelle, muss ein Algorithmus entscheiden, welches Modell als Referenz genutzt werden soll. Ohne vorherige Evaluation der Umgebung und der Bezugsobjekte, kann ein Algorithmus somit keine Aussage darüber treffen, welche Überlagerungsmodelle von Relevanz sein könnten. Eine vollautomatisierte Lösung würde hingegen ein für den vorgesehenen Anwendungsfall relevantes virtuelles Modell selbstständig auswählen. Dazu würde der Algorithmus Informationen aus der Umgebung aufnehmen, die zu Beginn des Überlagerungsvorgangs vorliegen. Mögliche Automatisierungsvorgänge wären beispielsweise eine Vorverarbeitung der Umgebung, um den korrekten Kontext und so passende Modelle auszuwählen.

Somit trifft für alle drei Überlagerungskonzepte der Automatisierungsgrad Stufe eins zu, der besagt, dass die Maschine keine Unterstützung anbietet und der Benutzer sämtliche Aktionen selbstständig ausführen muss. Im Bereich der Produktion könnten Lernalgorithmen bekannte Muster aufgreifen, um eine Vorauswahl der möglichen Überlagerungen selbstständig auszuwählen. Durch die stetige Weiterentwicklung von Anlagen in der Produktion stellt diese Vorgehensweise jedoch eine große Herausforderung dar. Eine Möglichkeit zur Bewältigung dieser Hürde wäre der Einsatz von maschinellem Lernen. Algorithmen könnten die bestehende Umgebung auf bekannte Muster

hin untersuchen, um eine Vorauswahl relevanter Modelle an den Benutzer zu übermitteln. Kann der Algorithmus den Anwendungsfall präzise genug erkennen, wird für jede Vorauswahl das korrekte Modell bestimmt, um eine weiterführende Überlagerung zu ermöglichen.

5.2 Automatisierung Informationsverarbeitung

Der Ansatz der manuellen Positionierung verwendet keine Automatisierung bei der Erkennung der Entität; der Benutzer muss somit selbst die Aktion durchführen und mittels seiner Perzeption das physische Objekt wahrnehmen und auswählen. Damit fällt das manuelle Überlagerungskonzept unter Stufe eins der Automatisierungsstufen. Im Gegensatz dazu erkennen Algorithmen im Marker- und Modell-basierten Konzept die physische Entität vollkommen selbstständig und ohne Information oder Eingriffsnotwendigkeit des Benutzers. Dies spiegelt eine Einteilung in Stufe zehn der Automatisierungsstufen wider.

Der Definition einer manuellen Vorgehensweise entsprechend, liefert ein solcher Überlagerungsvorgang keine Unterstützung bei der Erkennung von Entitäten in der Mixed Reality Umgebung. Die Erkennung der Entität hängt vollständig von der Wahrnehmung des Benutzers ab. Ein Marker- oder Modell-basierter Vorgang hingegen überprüft das vorliegende Sichtfeld des Benutzers selbstständig auf die vorab definierten Modelle; entweder durch das Erkennen eines angefügten Markers oder durch die Struktur der physischen Entität selbst. Die eingesetzten Vorgehensweisen wurden in Kapitel 2 erläutert. Um den manuellen Überlagerungsvorgang in dieser Phase automatisierbar zu gestalten, müsste ein Algorithmus die Perzeption des Benutzers ersetzen. Dies widerspricht jedoch der Vorgehensweise einer manuellen Überlagerung, weshalb in dieser Phase eine deutliche Unterscheidung wahrzunehmen ist. Die Wahl des korrekten Überlagerungskonzepts in dieser Phase ist anwendungsspezifisch. Kann ein Algorithmus mit einer hohen Wahrscheinlichkeit die korrekte physische Entität zur Überlagerung auswählen, ist ein Benutzereingriff nicht notwendig. Liegen jedoch falsch-positive oder falsch-negative Verwechslungsmöglichkeiten vor, muss diese Phase zusätzlich durch den Benutzer auf Korrektheit hin überwacht werden.

5.3 Automatisierung bei der Entscheidungsfindung

Da die manuelle Positionierung bereits bei der Informationsverarbeitung von der Perzeption des Benutzers abhängt, führt dies zwangsläufig dazu, dass die Auswahl des geeigneten, virtuellen Modells ebenfalls eine Interaktion des Benutzers erfordert. Wenn ein Algorithmus zur Unterstützung dieser Phase genutzt werden würde, gebe es durch die fehlende Automatisierung der vorangehenden Phasen keine Daten, die in dieser Phase ausgewertet werden könnten. Somit wird auch diese Phase in Stufe eins der Automatisierungsstufen eingeteilt. Was die Marker- und die Modell-basierte Erkennung betrifft, übernimmt in beiden Fällen ein Algorithmus die Auswahl des korrekten Modells. Dabei erhalten zwar beide Konzepte eine Einteilung in Stufe zehn der Automatisierungsstufen, jedoch besteht ein Vorzug in der Marker-basierten Erkennung: Da der Benutzer jeweils unterschiedliche Marker an einer Entität anbringen kann, lässt sich das Modell für eine einzige Entität variabel verändern. Somit können für ein und dieselbe physische Entität unterschiedliche animierte virtuelle

Modelle verwendet werden. Diese Eigenschaft lässt sich bei dem Modell-basierten Konzept nicht anwenden, da der Algorithmus das Modell anhand seiner Struktur erkennt und für strukturgleiche physische Entitäten immer dasselbe virtuelle Modell auswählt.

Wie in der vorherigen Phase beschrieben, setzt ein manueller Ansatz vollständig auf die Perzeption des Benutzers. Wurde eine physische Entität durch den Benutzer selbst erkannt, folgt darauf die Entscheidung des Benutzers, welche Überlagerung für diese Entität geeignet ist. Ein Algorithmus agiert in dieser Phase ähnlich; der Erkennung einer vordefinierten Entität folgt eine programmierte Überlagerung. Während der Benutzer kontextbedingt unterscheiden kann, wird ein Algorithmus jedoch stets nach der ihm vorgegebenen Logik ausführen. Diesbezüglich muss nach Vorlage des Anwendungsfalls entschieden werden. Sind die Entscheidungen, die in dieser Phase getroffen werden kritisch für den gesamten Prozess, sollte ein Algorithmus nur dann entscheiden, wenn die Korrektheit der Ausführung auch sichergestellt ist.

5.4 Automatisierung bei der Ausführung der Aktion

Die Ausführung der Aktion erfolgt bei der manuellen Positionierung durch den Benutzer mittels Gestensteuerung, was Stufe eins der Automatisierungsstufen entspricht. Im Fall des Marker- und Modell-basierten Ansatzes, führt der Algorithmus die Aktion vollautomatisch und ohne Eingriff oder vorherige Information des Benutzers durch, was Stufe zehn der Automatisierungsstufen entspricht. Eine manuelle Überlagerung hängt dabei immer von den Fähigkeiten des Benutzers ab, also mit der Vertrautheit der Benutzung der eingesetzten Software. Ebenso kann ein Algorithmus die Aktion nur dann befriedigend ausführen, wenn genügend Daten vorliegen, um eine zuverlässige Überlagerung der Entität zu ermöglichen. Die Auswahl des Überlagerungskonzepts in dieser Phase hängt von der Relevanz ab, wie sehr ein Ergebnis von der tatsächlichen Umgebung abweichen darf. Wird eine Überlagerung von einem Menschen betrachtet, lässt sich eine gewisse Abweichung kompensieren, da der Benutzer unter Einsatz seiner Perzeption Abweichungen erkennen kann. Wird die Überlagerung jedoch durch einen Algorithmus vollzogen und anschließend verarbeitet, muss ein gewisser Grad an Präzision vorausgesetzt werden. In wie weit diese Präzision für jede Überlagerungstechnik zutrifft wird durch die folgenden Eigenschaften bestimmt.

5.5 Überlagerungsgenauigkeit

Die durchschnittliche Genauigkeit der Überlagerung aller eingesetzten Techniken liegt über der Untergrenze von 6 Millimetern. Somit war im überwiegenden Teil der Positionen eine deutlich sichtbare Abweichung zwischen physischer Entität und virtueller Überlagerung messbar. Ein Beispiel eines Messergebnisses ist in Abbildung 5.1 zu sehen. Die Abbildung zeigt das Sichtfeld des Benutzers durch die Datenbrille aus der nahen Messposition. Die nachträglich eingezeichneten roten Hilfslinien stellen die Abweichung zwischen den Referenzpunkten der physischen Entität und der virtuellen Überlagerung dar. Die Messergebnisse der Überlagerungsgenauigkeit aller Positionen sind in Tabelle 5.1 zu sehen. Minima sind in grün, Maxima in rot dargestellt. Festzuhalten ist des Weiteren, dass sich die Marker-basierte Überlagerung nur auf eine Position bezieht, da der Marker von allen übrigen Positionen aus innerhalb der festgelegten Zeit nicht erkannt werden konnte. Die manuelle Positionierung schnitt innerhalb der festgelegten Zeit mit einer

Abweichung von durchschnittlich 7 Millimetern am genauesten ab. Die Werte der Modell-basierten Überlagerung entsprachen durchschnittlich 36 Millimeter und die Marker-basierte Überlagerung wies eine Abweichung von durchschnittlich 58 Millimetern auf. Eine Übersicht ist in Abbildung 5.2 zu sehen. Die Standardabweichung fiel mit 4,98 für die manuelle Überlagerungstechnik am geringsten aus. Der Marker-basierte Ansatz weist eine höhere Standardabweichung von 14,86 auf, die Modell-basierte Überlagerungstechnik besitzt mit 26,67 den höchsten Wert. Die Werte sind in Abbildung 5.3 gegenübergestellt. Eine ausführliche Übersicht der Messergebnisse ist in Abbildung 5.4 als Kastengrafik zu sehen.

Um die Tragfähigkeit dieser Messergebnisse zu bewerten, muss der entsprechende Anwendungsfall betrachtet werden. Ist eine präzise Überlagerung nicht notwendig, da lediglich die Zuweisung genügt, nicht jedoch ein punktgenauer Abgleich, kann die fehlende Genauigkeit vernachlässigt werden. Muss jedoch ein Algorithmus die Ergebnisse der Messung weiter verarbeiten, ist ein höherer Grad an Präzision notwendig. Die hohen Abweichungen könnten mit dem für den Anwendungsfall gewählten Versuchsobjekt zusammenhängen, um weiterführende Ergebnisse zu erhalten, müssten weitere Objekte in einem Überlagerungsvorgang gemessen werden.

5.6 Stabilität

Wie die Überlagerungsgenauigkeit, unterschreitet auch die Stabilität der Überlagerungstechniken die festgelegte Metrik. Die Ergebnisse zeigen ebenfalls eine deutlich sichtbare Abweichung zwischen physischer Entität und virtueller Überlagerung aus Sicht der übrigen Positionen. Die Messergebnisse der Stabilität sind in Tabelle 5.2 zu sehen. Diese sind in ihre ursprüngliche Position unterteilt: wurde ein Messdurchlauf aus der Position Front durchgeführt, befinden sich die Messergebnisse für die Positionen Nah, Seite und Fern unterhalb der Beschriftung *Ursprung Front*. Minima sind

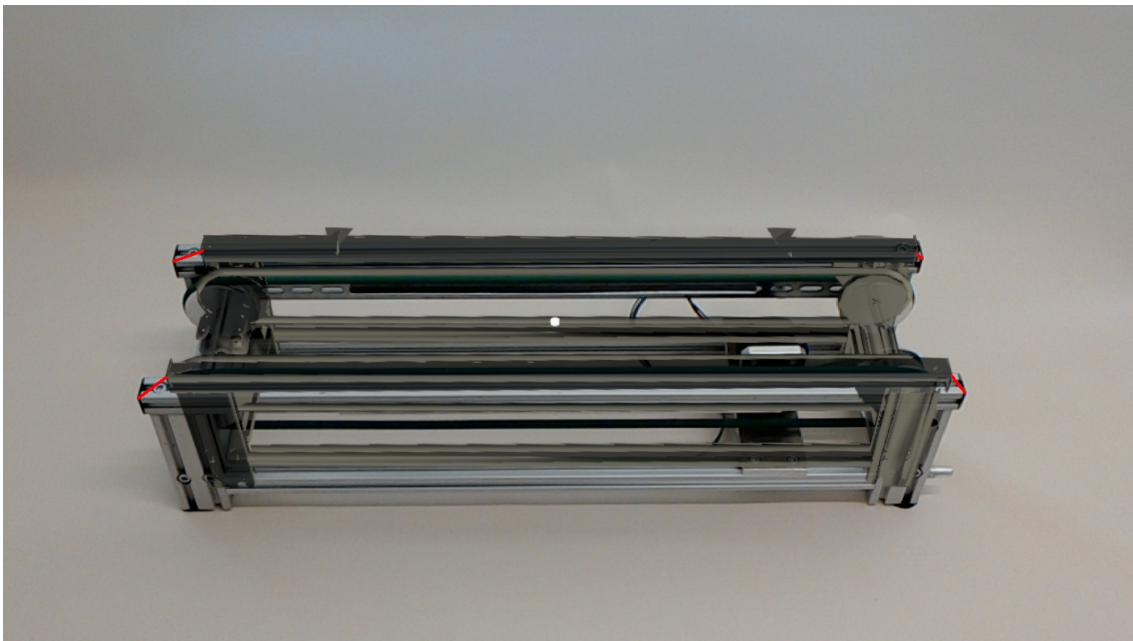


Abbildung 5.1: Aufnahme der manuellen Positionierung aus der Position nah

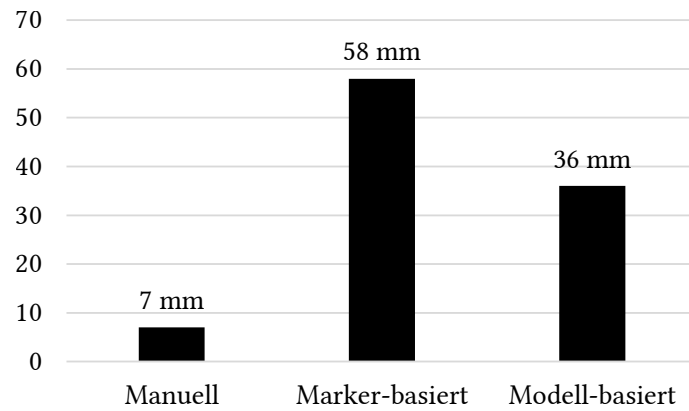


Abbildung 5.2: Durchschnittliche Abweichung der Überlagerung in Millimetern

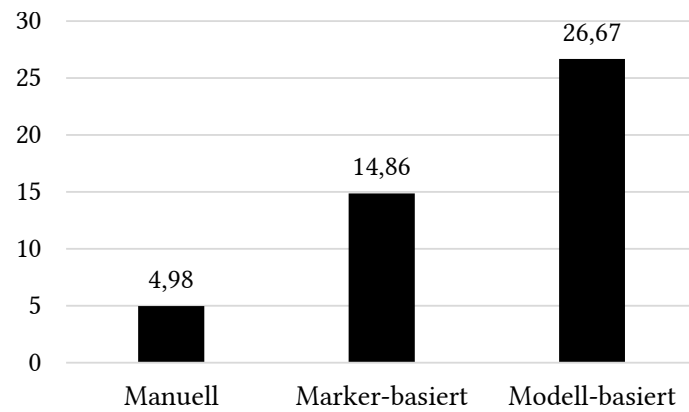


Abbildung 5.3: Standardabweichung der Überlagerung

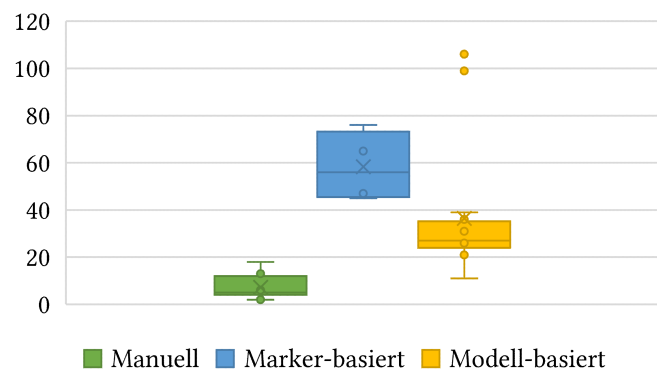


Abbildung 5.4: Kastengrafik der Abweichung der Überlagerung in Millimetern

in grün, Maxima in rot dargestellt. Wie bereits beschrieben bezieht sich die Marker-basierte Überlagerung nur auf eine Position, da der Marker von allen übrigen Positionen aus innerhalb der festgelegten Zeit nicht erkannt werden konnte. Die manuelle Positionierung schnitt mit einer Abweichung von durchschnittlich 26 Millimetern am genauesten ab. Dies besagt, dass eine virtuelle Überlagerung unter Einsatz der manuellen Überlagerungstechnik am präzisesten auf der physischen Entität verbleibt, wenn der Benutzer eine neue Position einnimmt. Die Werte der Modell-basierten Überlagerung entsprachen durchschnittlich 39 Millimeter und die Marker-basierte Überlagerung wies eine Abweichung von durchschnittlich 30 Millimetern auf. Eine Übersicht ist in Abbildung 5.5 zu sehen. Die Standardabweichung fiel mit 26,4 für die manuelle Überlagerungstechnik am geringsten aus. Der Marker-basierte Ansatz weist eine höhere Standardabweichung von 18,8 auf, die Modell-basierte Überlagerungstechnik besitzt mit 27,28 den höchsten Wert. Die Werte sind in Abbildung 5.6 gegenübergestellt. Eine vollständige Übersicht der Ergebnisse ist in Abbildung 5.7 zu sehen.

Tabelle 5.1: Einzelergebnisse der Messung zur Überlagerungsgenauigkeit

Position/Referenzpunkt	Manuell	Marker-basiert	Modell-basiert
Front links hinten	6 mm	65 mm	24 mm
Front links vorne	8 mm	76 mm	33 mm
Front rechts hinten	13 mm	45 mm	99 mm
Front rechts vorne	18 mm	47 mm	106 mm
Seite links hinten	2 mm	-	31 mm
Seite links vorne	5 mm	-	24 mm
Seite rechts hinten	4 mm	-	36 mm
Seite rechts vorne	4 mm	-	32 mm
Nah links hinten	13 mm	-	26 mm
Nah links vorne	15 mm	-	22 mm
Nah rechts hinten	3 mm	-	27 mm
Nah rechts vorne	9 mm	-	21 mm
Fern links hinten	2 mm	-	27 mm
Fern links vorne	4 mm	-	11 mm
Fern rechts hinten	5 mm	-	39 mm
Fern rechts vorne	4 mm	-	24 mm

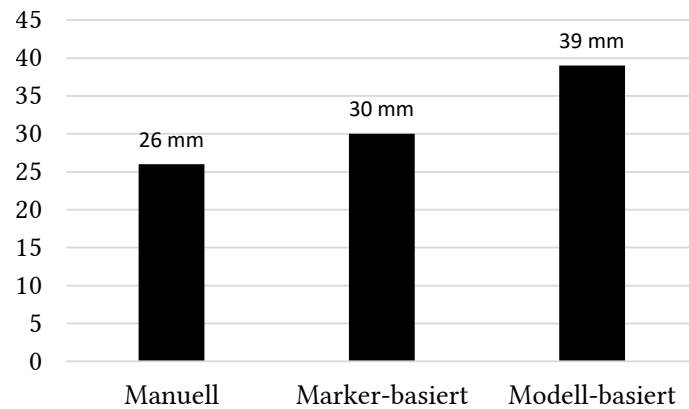


Abbildung 5.5: Durchschnittliche Abweichung aus Sicht der übrigen Positionen in Millimetern

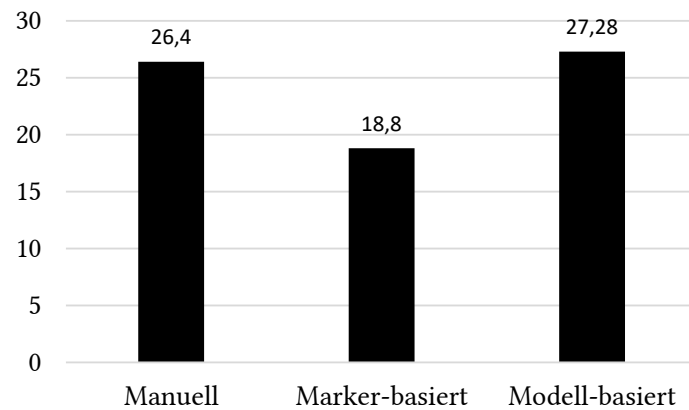


Abbildung 5.6: Standardabweichung der Abweichung aus Sicht der übrigen Positionen

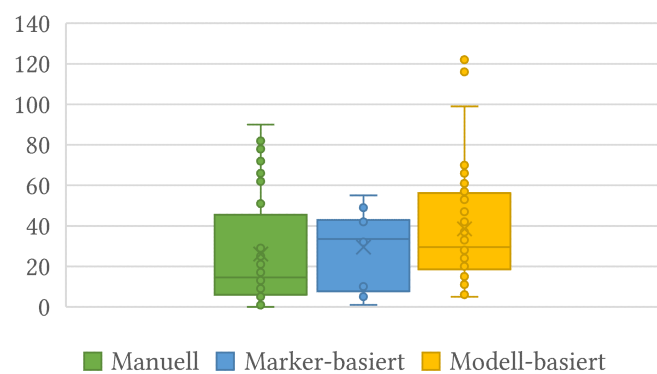


Abbildung 5.7: Kastengrafik der Abweichung aus Sicht der übrigen Positionen in Millimetern

Tabelle 5.2: Einzelergebnisse der Messung zur Stabilität

Position/Referenzpunkt	Manuell	Marker-basiert	Modell-basiert
Ursprung Front			
Fern links hinten	7 mm	7 mm	14 mm
Fern links vorne	5 mm	10 mm	11 mm
Fern rechts hinten	5 mm	5 mm	42 mm
Fern rechts vorne	5 mm	1 mm	37 mm
Seite links hinten	7 mm	34 mm	9 mm
Seite links vorne	6 mm	32 mm	26 mm
Seite rechts hinten	15 mm	43 mm	27 mm
Seite rechts vorne	14 mm	43 mm	44 mm
Nah links hinten	7 mm	49 mm	24 mm
Nah links vorne	6 mm	55 mm	20 mm
Nah rechts hinten	18 mm	33 mm	116 mm
Nah rechts vorne	20 mm	42 mm	122 mm
Ursprung Seite			
Fern links hinten	3 mm	-	31 mm
Fern links vorne	3 mm	-	28 mm
Fern rechts hinten	10 mm	-	24 mm
Fern rechts vorne	10 mm	-	22 mm
Front links hinten	1 mm	-	70 mm
Front links vorne	5 mm	-	64 mm
Front rechts hinten	25 mm	-	66 mm
Front rechts vorne	29 mm	-	53 mm
Nah links hinten	0 mm	-	64 mm
Nah links vorne	6 mm	-	57 mm
Nah rechts hinten	21 mm	-	61 mm
Nah rechts vorne	24 mm	-	62 mm
Ursprung Nah			
Fern links hinten	11 mm	-	6 mm
Fern links vorne	13 mm	-	5 mm
Fern rechts hinten	7 mm	-	7 mm
Fern rechts vorne	9 mm	-	8 mm
Front links hinten	23 mm	-	27 mm
Front links vorne	29 mm	-	25 mm
Front rechts hinten	21 mm	-	18 mm
Front rechts vorne	27 mm	-	20 mm
Seite links hinten	3 mm	-	17 mm
Seite links vorne	6 mm	-	15 mm

Auf nächster Seite fortgeführt

Tabelle 5.2 – Fortsetzung der Tabelle vorheriger Seite

Position/Referenzpunkt	Manuell	Marker-basiert	Modell-basiert
Seite rechts hinten	13 mm	-	15 mm
Seite rechts vorne	17 mm	-	14 mm
Ursprung Fern			
Front links hinten	51 mm	-	69 mm
Front links vorne	62 mm	-	36 mm
Front rechts hinten	54 mm	-	99 mm
Front rechts vorne	67 mm	-	68 mm
Nah links hinten	64 mm	-	25 mm
Nah links vorne	78 mm	-	33 mm
Nah rechts hinten	72 mm	-	54 mm
Nah rechts vorne	90 mm	-	47 mm
Seite links hinten	62 mm	-	36 mm
Seite links vorne	66 mm	-	37 mm
Seite rechts hinten	75 mm	-	47 mm
Seite rechts vorne	82 mm	-	26 mm

Wie bereits der Überlagerungsgenauigkeit zugeschrieben, ist die Bewertung der Stabilität anwendungsspezifisch. Besitzt die präzise Überlagerung von bewegten Objekten im vorgesehenen Einsatzgebiet eine untergeordnete Rolle, kann das Konzept trotz der unzureichenden Ergebnisse als geeignet eingestuft werden.

5.7 Ressourcenverbrauch CPU

Die Ergebnisse bezüglich der Auslastung der CPU während der Messung sind in Tabelle 5.3 zu sehen. Minima sind in grün, Maxima in rot dargestellt. Der durchschnittliche CPU-Verbrauch ist bei der manuellen Überlagerungstechnik mit durchschnittlich 23,13 % am niedrigsten. Die Ergebnisse der Marker-basierten und der Modell-basierten CPU-Auslastung fielen fast doppelt so hoch aus. Der Marker-basierte Ansatz misst eine durchschnittliche CPU-Auslastung von 41,36%, während sie bei dem Modell-basierten Ansatz 43,41% beträgt. Die Durchschnittswerte sind in Abbildung 5.8 zu sehen. Bei allen Ergebnissen fiel die Standardabweichung sehr gering aus, wobei für den Marker-basierten Ansatz keine Standardabweichung vorliegt, da nur eine Position erfolgreich von der Überlagerungstechnik erkannt worden ist. Die Standardabweichung der manuellen Überlagerung betrug 1, die des Modell-basierten Ansatzes fiel mit 1,93 höher aus. Die Werte sind in Abbildung 5.9 gegenübergestellt.

Da die manuelle Überlagerungstechnik auf Bilderkennungsalgorithmen verzichtet, liegt der Verbrauch der CPU-Ressourcen deutlich unter denen der Algorithmus-basierten Techniken. Diese wiederum lieferten keine deutlichen Unterschiede, was schlussfolgern lässt, dass die Konzepte auf

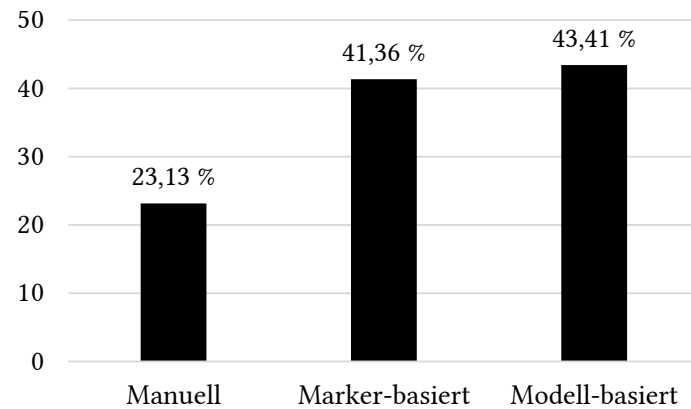


Abbildung 5.8: Durchschnittliche Auslastung der CPU in Prozent

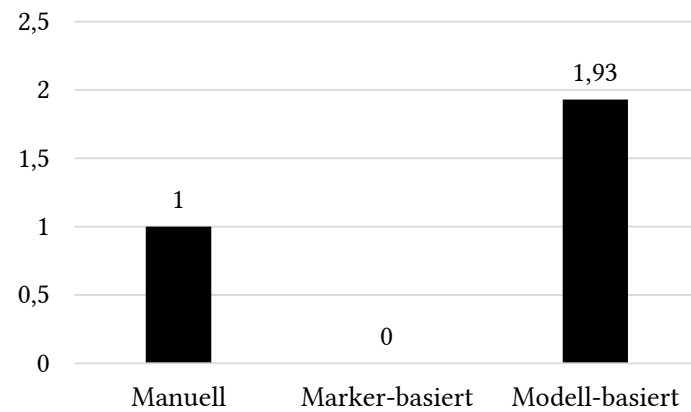


Abbildung 5.9: Standardabweichung der Auslastung der CPU

Tabelle 5.3: Einzelergebnisse der Messung zur Auslastung der CPU

Position	Manuell	Marker-basiert	Modell-basiert
Front	23,47%	41,36%	45,55%
Seite	24,4%	-	41,26%
Nah	22,33%	-	44,41%
Fern	22,33%	-	42,43%

ähnlich rechenintensiven Erkennungsalgorithmen basieren; im Fall der Marker-basierten Überlagerungstechnik muss das Bild eines Markers in den Framebildern erkannt werden, im Fall des Modell-basierten Ansatzes trainierte Muster, die auf das Vorhandensein des Objekts Hinweise geben.

5.8 Ressourcenverbrauch RAM

Die Ergebnisse der Messung des RAM-Verbrauchs zeigen, dass alle Überlagerungstechniken eine niedrige RAM-Auslastung aufweisen. Dabei fiel der Verbrauch der manuellen Überlagerung mit einer durchschnittlichen RAM-Auslastung von 5,55% am niedrigsten aus. Der Marker-basierte Ansatz erreichte eine Durchschnittsauslastung von 6,74%. Am höchsten fiel die Auslastung der Modell-basierten Überlagerungstechnik mit 9,39% aus. Die Einzelergebnisse jeder Position sind in Tabelle 5.4 zu sehen. Eine Gegenüberstellung der Durchschnittswerte ist in Abbildung 5.10 dargestellt. Die Messergebnisse der einzelnen Überlagerungstechniken sind zudem sehr ähnlich, was anhand der Standardabweichung (vgl. Abbildung 5.11) erkennbar ist. Somit weißt die manuelle Überlagerungstechnik eine Standardabweichung von 0,08 und die Modell-basierte eine Standardabweichung von 0,5 auf. Wie in Abschnitt 5.7 bereits erläutert, besitzt der Marker-basierte Ansatz keine Standardabweichung, da eine Erkennung nur von einer Position aus positiv verlief.

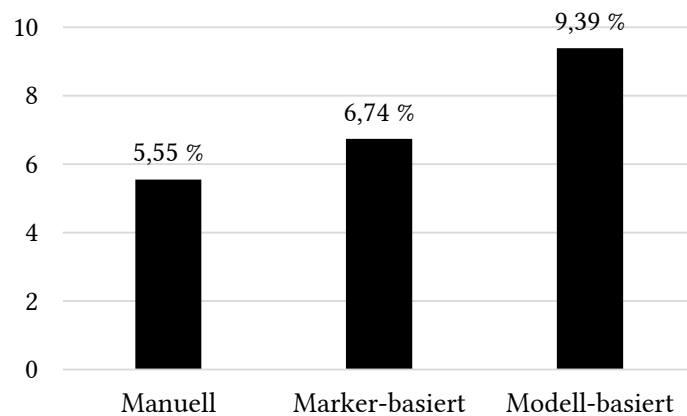


Abbildung 5.10: Durchschnittlicher RAM-Verbrauch in Prozent

Tabelle 5.4: Einzelergebnisse der Messung des RAM-Verbrauchs

Position	Manuell	Marker-basiert	Modell-basiert
Front	5,57%	6,74%	10,13%
Seite	5,63%	-	9,09%
Nah	5,43%	-	9,1%
Fern	5,56%	-	9,22%

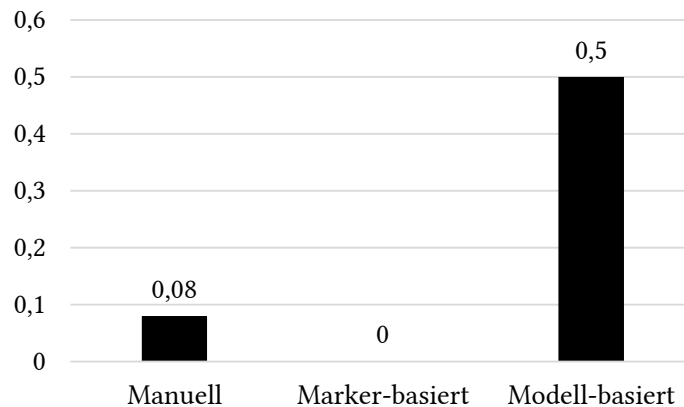


Abbildung 5.11: Standardabweichung des RAM-Verbrauchs

Die Arbeitsspeicher-Auslastung fiel bei allen drei Techniken unter 10% aus. Das niedrige Abschneiden der manuellen Überlagerungstechnik gibt Hinweise darauf, dass außer dem 3D-Modell nur wenige Daten im Hauptspeicher hinterlegt sind. Ebenso verhält es sich mit dem Marker-basierten Ansatz. Dieser fällt unter Umständen etwas höher aus, da Hilfsdaten für die Anwendung des Erkennungsalgorithmus hinterlegt werden. Das die Auslastung unter Anwendung des Modell-basierten Ansatzes am höchsten ausfällt, gibt Rückschlüsse darauf, dass die Trainingsdaten im Arbeitsspeicher abgelegt werden.

5.9 Zeitaufwand

Die Messung des Gesamtzeitaufwands ergab, dass die manuelle Überlagerungstechnik am wenigsten Vorbereitungszeit benötigt. Lediglich das 3D-Modell, das der Benutzer in der Applikation der physischen Entität überlagert, muss in der Entwicklungsumgebung eingefügt werden und um einen Manipulation Handler (vgl. Abschnitt 4.1.1) ergänzt werden. Dieses Vorgehen hat bei der Messung 54 Sekunden in Anspruch genommen. Der Marker-basierte Ansatz benötigte eine deutlich höhere Vorbereitungszeit, da der entsprechende Marker entworfen und ausgedruckt werden muss. Daraufhin wird der Marker an die physische Entität angebracht und in der Entwicklungsumgebung dem Modell zugewiesen. Der gesamte Vorgang dauerte 24 Minuten und 43 Sekunden. Die längste Vorbereitungszeit benötigte der Modell-basierte Ansatz. Das vorhandene 3D-Modell muss zunächst über den Model Target Generator (vgl. Abschnitt 3.1.4) um die vier Betrachtungspunkte ergänzt werden. Daraufhin müssen Trainingsdaten mit Hilfe des Cloud-Dienstes generiert werden, was entsprechend der momentanen Auslastung mehrere Minuten bis Stunden Zeit beansprucht. Schließlich müssen die generierten Trainingsdaten mit dem Modell innerhalb der Entwicklungsumgebung verbunden werden. Bei der durchgeführten Messung dauerte der gesamte Vorgang 49 Minuten und 12 Sekunden. Eine Übersicht der benötigten Vorbereitungszeiten ist in Abbildung 5.12 zu sehen.

Die Überlagerungstechniken lassen sich auch in zeitkritischen Prozessen einsetzen, falls keine stetige Neuerstellung der Erkennungsmuster notwendig ist. Wiederholt sich der Überlagerungsvorgang also stets mit dem selben Überlagerungsobjekt, langt eine einmalige Vorbereitung der Technik. Sind die Objekte jedoch einem ständigen Wandel unterworfen und muss diese Änderung möglichst schnell innerhalb des Prozesses adaptiert werden, eignen sich die Überlagerungstechniken mit langen Vorbereitungszeiten nicht.

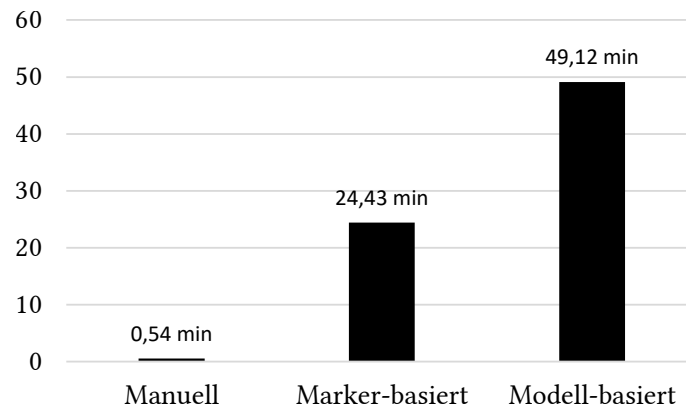


Abbildung 5.12: Insgesamt benötigte Vorbereitungszeit in Minuten

5.10 Gesamtübersicht

Eine Einteilung entsprechend der in Kapitel 4 definierten Metriken aller Überlagerungskonzepte ist in Tabelle 5.5 zu sehen. Auffällige Unterschiede sind zwischen dem manuellen und dem Marker-basierten sowie dem Modell-basierten Überlagerungskonzept ersichtlich. Da der Benutzer die Überlagerung bei dem manuellen Ansatz selbständig durchführt, erhält das Konzept eine Wertung von 1 für alle Automatisierungsphasen. Der Marker- und der Modell-basierte Ansatz handeln in allen Phasen, außer der Informationsaufnahme, komplett automatisiert und erhalten somit die volle Wertung. Weitere Unterschiede konnten im Verbrauch der CPU-Ressourcen beobachtet werden, wobei die Messung des manuellen Ansatz eine niedrigere Auslastung ergab als bei den übrigen Techniken. Auch bezüglich der Vorbereitungszeit konnten Unterschiede erkannt werden. Die manuelle Überlagerungstechnik benötigt, verglichen mit den beiden anderen Techniken, deutlich weniger Zeit. Jedoch existieren bei dem Marker- und Modell-basierten Ansatz ebenfalls deutliche Unterschiede: So benötigt der Modell-Ansatz fast doppelt so viel Zeit, wie für den Marker-basierten Ansatz gemessen wurde. Hinsichtlich der Überlagerungsgenauigkeit, der Stabilität und des RAM-Verbrauchs konnten keine auffälligen Unterschiede beobachtet werden.

Eine manuelle Überlagerung eignet sich im Kontext des Anwendungsfalls der Produktion dann, wenn eine Automatisierung des Prozesses nicht zwangsläufig notwendig ist. In diesem Fall kann die Perzeption des Benutzers, basierend auf den Ergebnissen des Vergleichs, bislang ein präziseres Ergebnis bezüglich der Überlagerungsgenauigkeit liefern als es mit den übrigen Konzepten umzusetzen war. Liegt der Fokus jedoch auf einer Automatisierung des Produktionsprozesses, muss auf ein entsprechendes Überlagerungskonzept zurückgegriffen werden. Dabei sind, wie die Messungen ergaben, jedoch gewisse Wechselwirkungen zu beachten. Zum einen überlagern die automatisierten Konzepte, wie der Marker- oder der Modell-basierte Ansatz die physischen Entitäten nicht präzise genug unter dem Einsatz einer Datenbrille und zum anderen benötigten diese Ansätze häufig mehr Vorbereitungszeit, weswegen sie für zeitkritische Prozesse nicht geeignet sind. Die Ergebnisse dieses Versuchs ermöglichen eine Vereinfachung der Prozessplanung in der Produktion unter Einbeziehung von Überlagerungstechniken mittels AR-Technologien. Erkenntnisse über die Automatisierung von bestimmten Überlagerungskonzepten und derer korrelierender Eigenschaften konnten durch diese Arbeit ergründet werden.

Tabelle 5.5: Analyseergebnisse als Harvey Ball-Diagramm

	Manuell	Marker- basiert	Modell- basiert
Automatisierung Informationsaufnahme			
Automatisierung Informationsverarbeitung			
Automatisierung Entscheidungsfindung			
Automatisierung Ausführung der Aktion			
Überlagerungsgenauigkeit			
Stabilität			
Ressourcenverbrauch CPU			
Ressourcenverbrauch RAM			
Zeitaufwand			

6 Schlussbetrachtung

Dieses Kapitel dient der Zusammenfassung der behandelten Themenfelder und des durchgeführten Versuchs. Dazu werden die Inhalte der Arbeit in Abschnitt 6.1 kurz erläutert. Ebenso wird in Abschnitt 6.2 ein Ausblick auf mögliche zukünftige Arbeiten gegeben und Verbesserungsvorschläge für aufgetretene Probleme erläutert.

6.1 Zusammenfassung

Durch den industriellen Wandel unter dem Leitbegriff der Industrie 4.0, werden eine Vielzahl von Technologien in die Produktionsprozesse integriert. Auch im Bereich der Mixed Reality bestehen weitreichende Forschungsbestrebungen, die Konzepte und technologischen Vertreter in Anwendungsbereichen wie der Medizin, der Unterhaltung, aber auch der Industrie effizient einzusetzen. Während Prozesse bei der Fertigung möglichst automatisiert ablaufen sollen, liegt ein Fokus im Bereich der Mixed Reality darauf, dem Benutzer Hologramme möglichst präzise und unter dem Einsatz von tragbarer Technologie einzublenden.

Zu diesem Zweck elaborierte diese Arbeit drei unterschiedliche Überlagerungskonzepte und verglich diese hinsichtlich ihrer Automatisierung in den unterschiedlichen Phasen einer Objekterkennung. Um weitere Unterschiede zwischen den Konzepten zu erarbeiten, wurden diese zudem auf ihre Genauigkeit, Stabilität der Hologramme unter Neupositionierung, Ressourcenverbrauch und benötigter Vorbereitungszeit hin untersucht. Dazu wurden umfassende Grundlagen zur Schaffung einer Metrik in der Automatisierung recherchiert und dokumentiert. Ebenso wurde bestehendes Wissen im Bereich der Mixed Reality und der Objekterkennung im Allgemeinen erarbeitet, um die Vorgänge beim Einsatz der Überlagerungskonzepte und -techniken verständlich zu gestalten und korrekt anzuwenden. Der Einsatz bestimmter Technologien wurde zudem durch das Elaborieren und Abgleichen des Stands der Technik begründet. Von besonderem Interesse war hierbei die Auswahl eines angepassten Augmented Reality SDKs, wobei die Wahl auf Vuforia und das Microsoft Mixed Reality Toolkit fiel.

Mit der gesammelten Wissensbasis war es im Anschluss möglich, den Versuchsaufbau zu strukturieren. Darin eingeschlossen war die Bestimmung der Überlagerungskonzepte, welche sich ausreichend unterscheiden und technisch angemessen realisierbar sein mussten. Als Ergebnis wurde eine manuelle Überlagerung, ein Marker-basierter Ansatz und ein Modell-basierter Ansatz gewählt. Zu diesen Konzepten wurde jeweils ein technischer Vertreter entwickelt, der schließlich aus mehreren Positionen zur Erkennung eines Förderbands eingesetzt wurde. Da diese Arbeit eine Grundlage zur Auswahl eines Überlagerungskonzept schaffen sollte, wurden die Metriken der Messgrößen unter Anwendung von Harvey Balls lediglich grob eingeordnet.

Kein Überlagerungskonzept stellte eine Automatisierung in der Phase der Informationsaufnahme bereit. Die übrigen Phasen wurden von dem Marker- und Modell-basierten Ansatz vollautomatisch ausgeführt; das manuelle Überlagerungskonzept war hingegen nicht automatisiert einsetzbar. Bezüglich der Überlagerungsgenauigkeit und der Stabilität konnte keine Überlagerungstechnik Messergebnisse erzielen, die eine Überlagerung ohne sichtbare Abweichung ermöglichen. Der Ressourcenverbrauch des Marker- und Modell-basierten Ansatz war gegenüber der manuellen Überlagerungstechnik leicht erhöht, die Auslastung des Arbeitsspeichers hingegen war bei allen drei Techniken ressourcenschonend. Deutliche Unterschiede waren bei der Messung der Vorbereitungszeit zu erkennen. Während der manuelle Ansatz eine verschwindend geringe Vorbereitungszeit benötigte, lag die Vorbereitungszeit für die Marker-basierte Überlagerungstechnik deutlich höher. Die Modell-basierte Technik benötigte die längste Vorbereitungszeit, was auf die zeitintensive Erstellung von Trainingsdaten zurückzuführen ist.

6.2 Ausblick

Der Versuchsaufbau dieser Arbeit basierte auf dem Einsatz der Microsoft HoloLens Datenbrille. Sensoren und deren Benutzung können sich unter Anwendung einer anderen Datenbrille stark unterscheiden. Aus diesem Grund müsste die Versuchsreihe um eine möglichst große Zahl an Geräten erweitert werden, damit ein genaueres Messergebnis der Überlagerungskonzepte und -techniken möglich wird. Auch im Bereich der eingesetzten SDKs existieren mehrere Alternativen, die für einen erweiterten Versuch gegenübergestellt werden müssten.

Das Experiment wurde mit nur einer Person durchgeführt, was hinsichtlich der Messung der Vorbereitungszeit und der Bedienung der Gestensteuerung bei der manuellen Überlagerung einseitige Messergebnisse liefert. Um diesen Ergebnissen statistische Relevanz zu verleihen, müsste der Versuch zusätzlich mit einer Gruppe von Probanden durchgeführt werden, die ein unterschiedliches Maß an Erfahrung mit dem Einsatz von Datenbrillen aufweisen. Diese Vorauswahl könnte durch das erstellen und auswerten von Fragebögen getroffen werden.

In dieser Arbeit wurden zudem nur jene Überlagerungskonzepte berücksichtigt, deren technische Vertreter bereits existieren. Weitere Konzepte, wie der Einsatz von künstlicher Intelligenz zur Auswahl der gewünschten 3D-Modelle auf Datengrundlage der Sensormessungen, könnten zu den Vergleichsobjekten hinzugefügt werden.

Literaturverzeichnis

- [AG15] D. Amin, S. Govilkar. „Comparative study of augmented reality SDKs“. In: *International Journal on Computational Science & Applications* 5.1 (2015), S. 11–26 (zitiert auf S. 45).
- [App18] Apple Inc. *ARKit*. 22. Dez. 2018. URL: <https://developer.apple.com/documentation/arkit> (zitiert auf S. 34).
- [Aug17] Augmenteer. *Welcome to Vuforia!* 13. Sep. 2017. URL: <https://forum.unity.com/threads/welcome-to-vuforia.495198/> (zitiert auf S. 49).
- [Azu97] R. T. Azuma. „A survey of augmented reality“. In: *Presence: Teleoperators & Virtual Environments* 6.4 (1997), S. 355–385 (zitiert auf S. 21).
- [BB95] S. S. Beauchemin, J. L. Barron. „The computation of optical flow“. In: *ACM computing surveys (CSUR)* 27.3 (1995), S. 433–466 (zitiert auf S. 26).
- [BETV08] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. „Speeded-up robust features (SURF)“. In: *Computer vision and image understanding* 110.3 (2008), S. 346–359 (zitiert auf S. 30, 31).
- [BL02] M. Brown, D. G. Lowe. „Invariant features from interest point groups.“ In: *BMVC*. Bd. 4. 2002 (zitiert auf S. 29).
- [BM92] P. J. Besl, N. D. McKay. „Method for registration of 3-D shapes“. In: *Sensor Fusion IV: Control Paradigms and Data Structures*. Bd. 1611. International Society for Optics und Photonics. 1992, S. 586–607 (zitiert auf S. 44).
- [BZ18] B. Bray, M. Zeller. *Calibration*. 21. März 2018. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/calibration> (zitiert auf S. 38).
- [BZS18] B. Bray, M. Zeller, N. Schonning. *What is mixed reality?* 21. März 2018. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/mixed-reality> (zitiert auf S. 20, 21).
- [CFA+11] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, M. Ivkovic. „Augmented reality technologies, systems and applications“. In: *Multimedia tools and applications* 51.1 (2011), S. 341–377 (zitiert auf S. 32).
- [CGCM07] D. Chekhlov, A. P. Gee, A. Calway, W. Mayol-Cuevas. „Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam“. In: *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society. 2007, S. 1–4 (zitiert auf S. 43).
- [CKF09] E. Costanza, A. Kunz, M. Fjeld. „Mixed reality: A survey“. In: *Human machine interaction*. Springer, 2009, S. 47–68 (zitiert auf S. 21–23).

- [CSC14] E. Ceseracciu, Z. Sawacha, C. Cobelli. „Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: proof of concept“. In: *PloS one* 9.3 (2014), e87640 (zitiert auf S. 44, 45).
- [DFG+05] W. Dangelmaier, M. Fischer, J. Gausemeier, M. Grafe, C. Matysczok, B. Mueck. „Virtual and augmented reality support for discrete manufacturing system simulation“. In: *Computers in Industry* 56.4 (2005), S. 371–383 (zitiert auf S. 13).
- [DP73] D. H. Douglas, T. K. Peucker. „Algorithms for the reduction of the number of points required to represent a digitized line or its caricature“. In: *Cartographica: The International Journal for Geographic Information and Geovisualization* 10.2 (1973), S. 112–122 (zitiert auf S. 25).
- [ENV99] D. ENV. 13005. *Leitfaden zur Angabe der Unsicherheit beim Messen (GUM)*. 1999 (zitiert auf S. 41).
- [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. 23, 24).
- [Fin10] K. Finkenzeller. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons, 2010 (zitiert auf S. 32).
- [Fit11] P. Fite-Georgel. „Is there a reality in industrial augmented reality?“ In: *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*. IEEE. 2011, S. 201–210 (zitiert auf S. 13, 33, 39).
- [GMMM14] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, M. J. Marín-Jiménez. „Automatic generation and detection of highly reliable fiducial markers under occlusion“. In: *Pattern Recognition* 47.6 (2014), S. 2280–2292 (zitiert auf S. 23–25, 34).
- [Goo18] Google LLC. *Recognize and Augment Images*. 14. Aug. 2018. URL: <https://developers.google.com/ar/develop/java/augmented-images/> (zitiert auf S. 33).
- [GSN09] P. F. Georgel, P. Schroeder, N. Navab. „Navigation tools for viewing augmented cad models“. In: *IEEE Computer graphics and applications* 29.6 (2009), S. 65–73 (zitiert auf S. 41, 42).
- [Ham50] R. W. Hamming. „Error detecting and error correcting codes“. In: *Bell System technical journal* 29.2 (1950), S. 147–160 (zitiert auf S. 23, 24).
- [KBB+18] K. Kim, M. Billingham, G. Bruder, H. B.-L. Duh, G. F. Welch. „Revisiting Trends in Augmented Reality Research: A Review of the 2nd Decade of ISMAR (2008–2017)“. In: *IEEE transactions on visualization and computer graphics* 24.11 (2018), S. 2947–2962 (zitiert auf S. 21, 39).
- [Keh17] P. Kehren. *Microsoft HoloLens wird Teil der neuen BMW X2 Kampagne*. 10. Nov. 2017. URL: <https://news.microsoft.com/de-de/mixed-reality-bmw/> (zitiert auf S. 13).
- [Keh18] P. Kehren. *Hologramme für Unternehmen: Neue Partner verstärken Mixed Reality Partner Programm*. 15. Feb. 2018. URL: <https://news.microsoft.com/de-de/neue-partner-verstarken-mixed-reality-partner-programm/> (zitiert auf S. 13).

- [KSF10] E. Kruijff, J. E. Swan, S. Feiner. „Perceptual issues in augmented reality revisited“. In: *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*. IEEE. 2010, S. 3–12 (zitiert auf S. 19, 47).
- [KSH12] A. Krizhevsky, I. Sutskever, G. E. Hinton. „Imagenet classification with deep convolutional neural networks“. In: *Advances in neural information processing systems*. 2012, S. 1097–1105 (zitiert auf S. 34).
- [LCWN15] B. LOESCH, M. CHRISTEN, R. WÜEST, S. NEBIKER. „Geospatial Augmented Reality–Lösungsansätze mit natürlichen Markern für die Kartographie und die Geoinformationsvisualisierung im Außenraum“. In: *T. Kersten (Hrsg.) 35 (2015)*, S. 89–97 (zitiert auf S. 49).
- [LF04] V. Lepetit, P. Fua. *Towards recognizing feature points using classification trees*. Techn. Ber. 2004 (zitiert auf S. 27).
- [LF06] V. Lepetit, P. Fua. „Keypoint recognition using randomized trees“. In: *IEEE transactions on pattern analysis and machine intelligence* 28.9 (2006), S. 1465–1479 (zitiert auf S. 27).
- [Lin98] T. Lindeberg. „Feature detection with automatic scale selection“. In: *International journal of computer vision* 30.2 (1998), S. 79–116 (zitiert auf S. 30).
- [LM02] R. Lienhart, J. Maydt. „An extended set of haar-like features for rapid object detection“. In: *Image Processing, 2002. Proceedings. 2002 International Conference on*. Bd. 1. IEEE. 2002, S. I–I (zitiert auf S. 34).
- [Low04] D. G. Lowe. „Distinctive image features from scale-invariant keypoints“. In: *International journal of computer vision* 60.2 (2004), S. 91–110 (zitiert auf S. 28, 29).
- [LSP+07] A. Leardini, Z. Sawacha, G. Paolini, S. Ingrosso, R. Nativio, M. G. Benedetti. „A new anatomically based protocol for gait analysis in children“. In: *Gait & posture* 26.4 (2007), S. 560–571 (zitiert auf S. 44).
- [Mic18] Microsoft. *Microsoft HoloLens*. 30. Dez. 2018. URL: <https://www.microsoft.com/hololens> (zitiert auf S. 38).
- [Mic19] Microsoft. *What is the Mixed Reality Toolkit*. 5. Apr. 2019. URL: <https://microsoft.github.io/MixedRealityToolkit-Unity/README.html> (zitiert auf S. 38).
- [MK94] P. Milgram, F. Kishino. „A taxonomy of mixed reality visual displays“. In: *IEICE TRANSACTIONS on Information and Systems* 77.12 (1994), S. 1321–1329 (zitiert auf S. 18, 19).
- [NKY05] Y. Nakazato, M. Kanbara, N. Yokoya. „Localization of wearable users using invisible retro-reflective markers and an IR camera“. In: *Stereoscopic Displays and Virtual Reality Systems XII*. Bd. 5664. International Society for Optics und Photonics. 2005, S. 563–571 (zitiert auf S. 25).
- [NY99] U. Neumann, S. You. „Natural feature tracking for augmented reality“. In: *IEEE Transactions on Multimedia* 1.1 (1999), S. 53–64 (zitiert auf S. 26, 27).
- [Ope15] OpenCV team. *Detection of ChArUco Corners*. 18. Dez. 2015. URL: https://docs.opencv.org/3.1.0/df/d4a/tutorial_charuco_detection.html (zitiert auf S. 34, 35).

- [Ope18] OpenCV team. *About*. 23. Dez. 2018. URL: <https://opencv.org/about.html> (zitiert auf S. 34).
- [PBDM07] K. Pentenrieder, C. Bade, F. Doil, P. Meier. „Augmented Reality-based factory planning-an application tailored to industrial needs“. In: *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society. 2007, S. 1–9 (zitiert auf S. 40).
- [PPV16] A. Patrizi, E. Pennestrí, P.P. Valentini. „Comparison between low-cost markerless and high-end marker-based motion capture systems for the computer-aided assessment of working ergonomics“. In: *Ergonomics* 59.1 (2016), S. 155–162 (zitiert auf S. 44, 45).
- [PSW00] R. Parasuraman, T. B. Sheridan, C. D. Wickens. „A model for types and levels of human interaction with automation“. In: *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans* 30.3 (2000), S. 286–297 (zitiert auf S. 18, 51).
- [PTC18] PTC Inc. *Vuforia Engine Features*. 22. Dez. 2018. URL: <https://library.vuforia.com/content/vuforia-library/en/features/overview.html> (zitiert auf S. 35, 37).
- [PTC19] PTC Inc. *Model Target Generator User Guide*. 2. Apr. 2019. URL: <https://library.vuforia.com/articles/Solution/model-target-generator-user-guide.html> (zitiert auf S. 50).
- [PZ17] F. Peng, J. Zhai. „A mobile augmented reality system for exhibition hall based on Vuforia“. In: *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*. IEEE. 2017, S. 1049–1052 (zitiert auf S. 49).
- [Qui86] J. R. Quinlan. „Induction of decision trees“. In: *Machine learning* 1.1 (1986), S. 81–106 (zitiert auf S. 28).
- [RD06] E. Rosten, T. Drummond. „Machine learning for high-speed corner detection“. In: *European conference on computer vision*. Springer. 2006, S. 430–443 (zitiert auf S. 28).
- [Roa06] D. Roam. *Harvey balls? Harvey who?* 20. März 2006. URL: https://digitalroam.typepad.com/digital_roam/2006/03/harvey_balls_ha.html (zitiert auf S. 52).
- [RSW+18] Q. Raddich, M. Satran, T. Whitney, M. Jacobs, S. Weston. *What's a Universal Windows Platform (UWP) app?* 6. Dez. 2018. URL: <https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide> (zitiert auf S. 51).
- [ST93] J. Shi, C. Tomasi. *Good features to track*. Techn. Ber. Cornell University, 1993 (zitiert auf S. 26).
- [Suz+85] S. Suzuki et al. „Topological structural analysis of digitized binary images by border following“. In: *Computer vision, graphics, and image processing* 30.1 (1985), S. 32–46 (zitiert auf S. 25).
- [SV78] T. B. Sheridan, W. L. Verplank. *Human and computer control of undersea teleoperators*. Techn. Ber. MASSACHUSETTS INST OF TECH CAMBRIDGE MAN-MACHINE SYSTEMS LAB, 1978 (zitiert auf S. 17, 52).
- [Val18] Valve Corporation. *Virtuelle Realität auf Steam*. 30. Dez. 2018. URL: <https://store.steampowered.com/vr/> (zitiert auf S. 43).

- [Wec06] M. Weck. *Werkzeugmaschinen 4: Automatisierung von Maschinen und Anlagen*. Springer-Verlag, 2006 (zitiert auf S. 13).
- [WWL+16] H. Wang, F. Wang, A. P. Y. Leong, L. Xu, X. Chen, Q. Wang. „Precision insertion of percutaneous sacroiliac screws using a novel augmented reality-based navigation system: a pilot study“. In: *International orthopaedics* 40.9 (2016), S. 1941–1947 (zitiert auf S. 41, 42).
- [XL14] C. Xiao, Z. Lifeng. „Implementation of mobile augmented reality based on Vuforia and Rawajali“. In: *2014 IEEE 5th International Conference on Software Engineering and Service Science*. IEEE. 2014, S. 912–915 (zitiert auf S. 49).
- [ZB18] M. Zeller, B. Bray. *HoloLens hardware details*. 21. März 2018. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details> (zitiert auf S. 38).
- [ZDB08] F. Zhou, H. B.-L. Duh, M. Billinghurst. „Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR“. In: *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society. 2008, S. 193–202 (zitiert auf S. 22).

Alle URLs wurden zuletzt am 14. 05. 2019 geprüft.

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