



# CARL VON OSSIEZKY UNIVERSITÄT OLDENBURG

INFORMATIK  
BACHELORARBEIT

---

## Entwicklung einer Augmented Reality Anwendung zum Tracken und Erstellen von Markern für den Bildungsbereich

---

*Autor:*  
Johannes Scheibe

*Erstgutachter:*  
Prof. Dr.-Ing. Jürgen Sauer

*Zweitgutachter:*  
M. Sc. B. Eng. Nils Hartmann

Abteilung Systemanalyse und -optimierung  
Department für Informatik

Oldenburg, 7. August 2020

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>iv</b>
<b>Tabellenverzeichnis</b>	<b>v</b>
<b>Abkürzungsverzeichnis</b>	<b>vi</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Ziele der Arbeit . . . . .	1
1.3. Aufbau der Arbeit . . . . .	1
<b>2. Grundlagen</b>	<b>2</b>
2.1. Augmented Reality . . . . .	2
2.1.1. Einsatzbereiche . . . . .	3
2.1.2. Technische Grundlagen . . . . .	4
2.2. OpenGL . . . . .	5
<b>3. Verwandte Arbeiten</b>	<b>6</b>
<b>4. Anforderungsanalyse</b>	<b>7</b>
4.1. Vision . . . . .	7
4.1.1. Mögliche Anwendungsfälle . . . . .	7
4.2. Der Prototyp . . . . .	8
4.3. Anforderungsanalyse . . . . .	8
4.3.1. Beschreibung der Systemumgebung . . . . .	9
4.3.2. Anwendungsfälle des Prototyps . . . . .	9
4.3.3. Anforderungsliste . . . . .	10
<b>5. Entwurf</b>	<b>13</b>
<b>6. Entwicklung</b>	<b>14</b>
6.1. Anforderungen . . . . .	14
6.2. Konzeptentwurf . . . . .	14
6.3. Implementierung . . . . .	14

6.4. Tests . . . . .	14
6.4.1. Testdurchführung . . . . .	14
6.4.2. Testfälle . . . . .	15
6.4.3. Zusammenfassung . . . . .	17
<b>7. Evaluation</b>	<b>18</b>
<b>8. Fazit</b>	<b>19</b>
8.1. Fazit . . . . .	19
8.2. Ausblick . . . . .	19
<b>Literaturverzeichnis</b>	<b>20</b>
<b>A. Beispielanhang</b>	<b>22</b>

# **Abbildungsverzeichnis**

2.1. RV Kontinuum . . . . .	2
2.2. Snapchat AR . . . . .	3
2.3. Barcode Marker . . . . .	5
4.1. Use Cases des Prototyps . . . . .	10
6.1. Testaufbau . . . . .	15
6.2. Perspektiven eines Markers . . . . .	16
6.3. Skalierungen eines Markers . . . . .	16
6.4. Rotationen eines Markers . . . . .	16
6.5. Belichtungen eines Markers . . . . .	17

# **Tabellenverzeichnis**

# **Abkürzungsverzeichnis**

**AR** Augmented Reality

**VR** Virtual Reality

# **Kapitel 1**

## **Einleitung**

**1.1. Motivation**

**1.2. Ziele der Arbeit**

**1.3. Aufbau der Arbeit**

# Kapitel 2

## Grundlagen

Dieses Kapitel behandelt die Grundlagen, die für die Realisierung des Prototyps notwendig sind. Dabei werden Methoden und Eigenschaften der Augmented Reality erläutert, sowie ein Überblick über OpenGL gegeben.

### 2.1. Augmented Reality

In der Literatur lassen sich für den Begriff der Augmented Reality (AR, deutsch: „angereicherte Realität“) viele unterschiedliche Definitionen finden, die meisten stützen sich dabei auf das von Milgram u. a. (1994) definierte Reality-Virtuality (RV) Kontinuum, welches in Abbildung 2.1 dargestellt ist.

Um dieses zu verstehen muss zunächst der Begriff der Virtual Reality (VR, deutsch: „virtuelle Realität“) definiert werden. Nach Klein (2006, S.1) beschreibt die VR eine völlig künstliche, computergenerierte Welt, in die der Nutzer eintauchen kann.

Milgrams Definition fasst nun die Virtual Reality und die reale Welt als zwei, sich gegenüberliegende Enden eines Kontinuums auf. Dabei ist die reale Welt an die physikalischen Gesetze gebunden, während die virtuelle Welt diese überschreiten und sich von ihnen lösen kann (Milgram u. a., 1994, S. 283). Nach dem RV Kontinuum bewegt sich die Augmented Reality zwischen beiden Welten und stellt ein Kombination beider dar. Eine weitere Definition, die sich auch mit der von Milgram

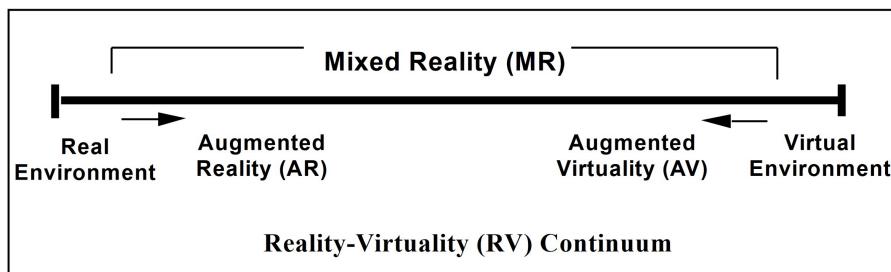


Abbildung 2.1.: Das Reality-Virtuality (RV) Kontinuum. (Quelle: Milgram u. a. (1994, S. 283))

vereinbaren lässt, beschreibt die Augmented Reality als eine computergestützte Erweiterung der wahrnehmbaren Realität um virtuelle Objekte (Kind u. a., 2019, S. 9). Auf dieser Grundlage kann man Augmented Reality als das Einbinden und Visualisieren digitaler, computergenerierte Objekte in der realen Welt auffassen. Oftmals wird dabei das Ziel verfolgt eine möglichst realistische Illusion für den Nutzer zu schaffen.

### 2.1.1. Einsatzbereiche

Die erste Assoziation, die die meisten mit dem Begriff Augmented Reality verbinden ist vermutlich der Unterhaltungsbereich. Große Firmen wie zum Beispiel Snapchat nutzen die Technologie um kleine Gimmicks für ihre Nutzer bereitzustellen (siehe Abbildung 2.2) Die meisten Personen, die den Begriff Augmented Reality hören, werden vermutlich an ein lustiges Gimmick zur Unterhaltung denken. Doch auch neben dem Bereich der Unterhaltung wird AR an vielen weiteren Stellen eingesetzt. Beispielsweise zu nennen wären hier der Bereich der Produktion, in welchem AR unter Anderem als Hilfsmittel zum Prototyping (Kind u. a., 2019, S. 44) genutzt werden kann, oder der Bereich der Medizin. Im letzteren können mittels AR Therapiemaßnahmen für psychische Erkrankungen oder Assistenzsysteme zur Diagnose und Operation entwickelt werden (Kind u. a., 2019, S. 52, 54).

Der Einsatzbereich in dessen Rahmen sich diese Arbeit bewegt ist jedoch der Bereich der Bildung.

Hier bietet Augmented Reality die Möglichkeit eines neuen Informationsmediums, welches vor allem zur Betrachtung dreidimensionaler Objekte genutzt werden kann. Dieses ermöglicht ein verbessertes, räumliches Verständnis des Lerninhaltes. Laut einer systematischen Analyse der Universität Stockholm, in welchem basierend auf der Anzahl der wissenschaftlichen Veröffentlichungen Schlüsse für das Lernen mit AR gezogen wurden, sind vor allem die Naturwissenschaften relevant für den Einsatz von AR (Hedberg u. a., 2018, S. 81).



Abbildung 2.2.: AR Gimmick aus der Anwendung Snapchat. (Quelle: Screenshot aus der Anwendung Snapchat, 01.08.2020)

## 2.1.2. Technische Grundlagen

Zur Umsetzung der Augmented Reality ist eine Umgebungserfassung und -analyse des Systems mit Hilfe einer Tracking Software (auch Tracker genannt) notwendig. Auf Grundlage dieser können dann im Anschluss computergenerierte virtuelle Objekte in die Umgebung eingefügt werden.

Zur Analyse der Systemumgebung können verschiedene Eingabesysteme genutzt werden, mit deren Hilfe die Eigenschaften der Umgebung und der in ihr vorhandenen Objekte wahrgenommen werden können (Kind u. a., 2019, S. 22). Zu diesen Eigenschaften zählen neben statischen, wie der Größen und Position, auch dynamische Eigenschaften, welche die Veränderung der statischen Attribute einzelner Objekte umfassen. Beispielhafte Technologien, die zur Erfassung genutzt werden können, wären Kamerasysteme, Laser, Infrarot oder sonstige Sensoren (Kind u. a., 2019, S. 22). Neben der Umgebungserfassung ist auch die Verfolgung einzelner, in der Umgebung enthaltender Objekte notwendiger Bestandteil der Tracking Software, um die dynamischen Eigenschaften der realen Umgebung auf die virtuellen Objekte zu übertragen.

Umformulieren,  
wegen Zitat

Eine wichtige Rolle beim Tracking spielt die Genauigkeit, mit der die Eigenschaften der Umgebung wahrgenommen werden, sie bestimmt wie akkurat virtuelle Objekte in die reale Umgebung eingefügt werden können und wie realistisch die Illusion erscheint (Klein, 2006, S. 2).

### Trackingverfahren

Grundsätzlich kann bei der Tracking Software zwischen zwei Verfahren unterschieden werden, dem nicht visuellen und dem visuellen Tracking (Mehler-Bicher und Steiger, 2014, S. 26). Während bei letzterem auf Daten von zum Beispiel einer Kamera zurückgegriffen wird, beruht das nichtvisuelle Tracking auf Sensoren, die direkten Zugriff auf die Eigenschaften der Umgebung, wie der Position, liefern.

Bei dem für diese Arbeit relevantem visuellen Tracking, müssen die Eigenschaften der Umgebung aus dem Kamerabild abgeleitet und verarbeitet werden. Dazu werden in der Regel nach Mehler-Bicher und Steiger (2014, S. 26) zwei Schritte benötigt:

1. Die Initialisierung, bei welcher nach einem bestimmten Muster im Kamerabild gesucht wird. Dieses kann dabei im Vorfeld definiert sein oder aus dem Kamerabild abgeleitet werden.
2. Die Verfolgung bzw. Antizipation der möglichen Bewegung, bei welcher das gefundene Muster in den einzelnen, aufeinanderfolgenden Videoframes verfolgt wird und eine Prognose der zukünftigen Position des Muster berechnet wird,

um den Rechenaufwand zu verkleinern.

Des Weiteren kann beim Visuellen Tracking zwischen Marker Based Feature Tracking und Natural Feature Tracking unterschieden werden.

**Marker Based Feature Tracking** verwendet feste, im Vorfeld definierte Muster, die in der Umgebung platziert werden. Diese Muster werden als Marker bezeichnet. Meistens handelt es sich dabei um schwarze Quadrate, in deren Mitte eine ID als ein Muster aus schwarzen und weißen Vierecken codiert ist. Ein beispielhafter Marker ist in Abbildung 2.3 zusehen. ~~Die Marker sind dabei optisch so angepasst, dass sie sehr~~

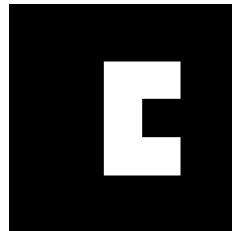


Abbildung 2.3.: Ein Barcode Marker mit der ID 10. (Quelle: Generiert mit dem Marker Generator <https://au.gmented.com/app/marker/marker.php>)

link ins Literaturverzeichnis?

einfach von einem Tracker erfasst werden können, um die Erkennungsgeschwindigkeit und somit die Performanz des gesamten Systems zu optimieren (Mehler-Bicher und Steiger, 2014, S. 28).

Owen u. a. stellen dazu folgende Anforderungen an einen optimalen Marker:

- Ein idealer Marker sollte die eindeutige Bestimmung seiner Position und Orientierung im Verhältnis zur Kamera unterstützen.
- Der Marker sollte alle Ausrichtungen unterstützen.
- Der Marker sollte Teil einer Reihe an Markern sein, die sich eindeutig von einander unterscheiden lassen.
- Ein Marker muss einfach lokalisierbar und identifizierbar sein.
- Die Marker müssen über einen weiten Aufnahmebereich funktionieren

(Nach Owen u. a. (2002, S. 2))

Durch diese Eigenschaften ist die Erkennung von einem Marker relativ simpel und funktioniert im Allgemeinen in drei Schritten:

### Natural Feature Tracking

## 2.2. OpenGL

# **Kapitel 3**

## **Verwandte Arbeiten**

# Kapitel 4

## Anforderungsanalyse

In diesem Kapitel werden die Anforderungen an die Anwendung und den zu entwickelnden Prototypen erhoben. Dazu werden verschiedene Anwendungsfälle herbeigezogen und daraus Anforderungen abgeleitet.

### 4.1. Vision

Die Hauptzielgruppe der Anwendung ist der Bildungsbereich. Sie soll Schülern, Studieren und auch Professoren die Möglichkeit bieten mit Hilfe von Augmented Reality das Lernen sowie Lehren zu bereichern.

Das Grundprinzip ist dabei folgendes: Die App soll dem Anwender die Möglichkeit geben 3D-Modelle hochzuladen, für die dann ein einzigartiger Marker generiert wird. Dieser kann dann ausgedruckt oder anderweitig angezeigt werden. Mit Hilfe der Kamera lässt sich dann das entsprechende 3D-Modell im Augmented Reality Bereich darstellen.

#### 4.1.1. Mögliche Anwendungsfälle

Im folgenden werden mögliche Anwendungsfälle für zwei verschiedene Personas definiert.

##### Der Professor

Das Anwendungsszenario, welches für einen Professor oder Lehrer denkbar wäre, orientiert sich an der bereits existierenden Webseite Socrative, welche bereits im Universitätsalltag häufig eingesetzt wird und Lehrenden die Möglichkeit gibt Umfragen zu erstellen, die von Studierenden über einen Raumcode beantwortet werden können.

Ein solches Raumsystem wäre auch für die AR Anwendung denkbar. Mit Hilfe der Anwendung könnte dann ein Professor einen Raum erstellen, in welchem er verschiedene 3D-Modelle speichern könnte, die Marker könnten im Anschluss dann Online

ref socrative

zur Verfügung gestellt, über Ausdrucke vervielfältigt oder in die Präsentation eingebunden werden.

Die Studierenden können dann über einen Zugangscode dem Raum beitreten und dann mit Hilfe der Kamera die entsprechenden Modelle des Raumes, in dem sie sich aktuell befinden, anzeigen lassen.

### **Die Studierenden**

Den Studierenden soll neben der Möglichkeit öffentlichen Räumen beizutreten auch die Möglichkeit gegeben werden selbst Modelle hochzuladen und die entsprechenden Marker zum Beispiel auf Lernzettel zu drucken. Dazu könnte jeder einen privaten Raum besitzen. Die Modelle die der Anwender hier hochlädt, wären dann nur lokal gespeichert und nicht öffentlich über einen Code oder Ähnlichem zugänglich.

## **4.2. Der Prototyp**

Der zu entwickelnde Prototyp fokussiert sich dabei auf die Umsetzung der Augmented Reality und das Erstellen der zu trackenden Marker. Die Modelle, die hochladen werden, werden dabei lediglich lokal gespeichert.

Dadurch handelt es sich bei dem Prototyp um die Umsetzung des „privaten Raumes“. Er bietet dem Anwender nicht die Möglichkeit die Modelle zuteilen.

Eine ansonsten notwendige Datenspeicherung auf einem Server, das Implementieren eines Raumsystems mit Zugangscode und weitere Funktionen fallen dadurch bei diesem ersten Prototyp weg.

## **4.3. Anforderungsanalyse**

Im folgenden wird eine Anforderungsanalyse für den zu entwickelnden Prototyp durchgeführt. Dazu wurde das in der Softwaretechnik I Vorlesung vorgestellte Verfahren zur Anforderungsdefinition genutzt (Winter, 2018, Folie 209-214). Dieses Verfahren sieht den folgenden Aufbau vor:

1. Vision
2. Machbarkeitsstudie
3. Beschreibung der Systemumgebung
4. Anwendungsfälle
5. Anforderungsliste

## 6. Prototypen

## 7. Glossar

Im folgenden wird jedoch nur eine vereinfachte, an diese Arbeit angepasste Version dieses Verfahrens genutzt.

### **4.3.1. Beschreibung der Systemumgebung**

Bei dem Prototyp handelt es sich um eine mobile Anwendung. Das Endgerät für den Nutzer ist also ein Smartphone.

Der relevante Sensor, den das Smartphone für das Trackingverfahren bereitstellt, ist die Kamera. Die Anwendung muss die Videoframes analysieren und eine Ausgabe in Form von einem gerenderten 3D-Modell auf dem Display darstellen.

Insgesamt besitzt der Prototyp drei Unterfunktionen:

1. Das Generieren von Markern. Hierbei erstellt der Prototyp Marker für den Anwender, die alle eine unterschiedliche ID besitzen und aus jeder Rotation eindeutig zu erkennen sind. Dafür müssen sich zwei Marker auch unterscheiden, wenn einer von ihnen beliebig rotiert wurde.
2. Das Tracken von Markern. Bei dem die Anwendung einen Marker in einem Videoframe erkennt und seine Position und Transformation im Videoframe berechnet. Dabei liegt der Marker in gedruckter Form auf einem Blatt Papier vor oder wird auf einem Bildschirm angezeigt.
3. Das Rendern von Modellen. Das anzuzeigende Modell wurde dabei vom Nutzer als Datei hochgeladen und muss zunächst vom System verarbeitet werden. Um im Anschluss das Modell realistisch in Relation zum Marker zu platzieren, muss die beim Tracking berechnete Position und Transformation des Markers verwendet werden.

Da der Prototyp auf einem mobilen Endgerät genutzt wird, variiert die Umgebung in welcher dieser zum Einsatz kommt stark. Deshalb ist es notwendig, dass das System robust gegenüber verschiedenen Umwelteinflüssen ist.

### **4.3.2. Anwendungsfälle des Prototyps**

Ein Beispielhaftes Anwendungsszenario des Prototypen, das sich aus dem Anwendungsfall des Studierenden ableitet (siehe Abschnitt 4.1.1), ist die mit dem Prototyp

generierten Marker auf einem Lernzettel einzufügen und ein vorlesungsrelevantes Modell zu verlinken. Dazu wird das Modell in dem Prototyp hochgeladen und der generierte Marker in ein Dokument eingefügt. Im Anschluss kann dann das Dokument mit der Kamera gefilmt werden, um sich das 3D-Modell anzeigen zu lassen. Allgemein lassen sich die Funktionen und Anwendungsfälle in dem in Abbildung 4.1 gezeigtem Anwendungsfalldiagramm visualisieren.

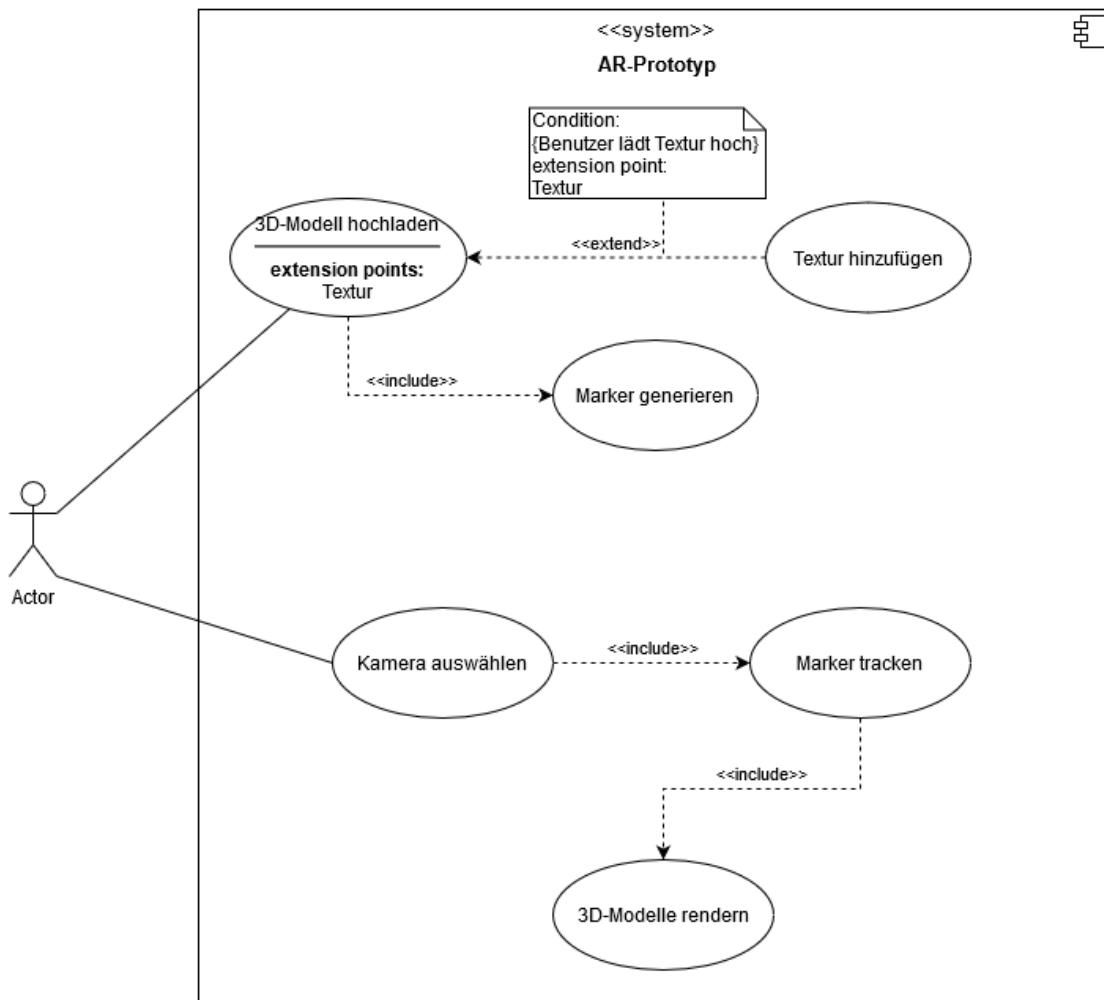


Abbildung 4.1.: Use Case Diagramm des Prototyps. (Quelle: Eigene Darstellung)

### 4.3.3. Anforderungsliste

Im folgenden Abschnitt werden die Anforderungen an den Prototyp definiert. Dabei wird zwischen funktionalen und nichtfunktionalen Anforderungen unterschieden.

**Funktionale Anforderungen** beschreiben die Funktionen eines Systemes, die notwendig sind, damit dieses seine Aufgaben erfüllen kann (Robertson und Robertson, 2012, S. 10).

**Nichtfunktionale Anforderungen** beschreiben Eigenschaften, die das System besitzen muss, um den Nutzer zufrieden zu stellen (Robertson und Robertson, 2012, S. 10). Diese Eigenschaften beziehen sich vor allem auf die Bereiche Service Level, Zugriffsbeschränkungen, Sicherheit, Monitoring, Kontrolle, Schnittstellen, Archivierung, Benutzerfreundlichkeit und Konversion (Böhm und Fuchs, 2002, S. 139)

Die folgende Anforderungsliste leitet sich aus den Anwendungsfällen des Prototyps ab.

### Funktionale Anforderungen

- FA1 Die Anwendung soll ein Set von mindestens 10 Markern generieren können.
- FA2 Die Anwendung soll die generierten Marker in einem Kamerabild erkennen können.
- FA3 Die Anwendung soll die generierten Marker unterscheiden können.
- FA4 Der Benutzer soll eigene 3D-Modelle als OBJ-Dateien hochladen können.
- FA5 Die Anwendung soll OBJ-Datei laden und verarbeiten können.
- FA6 Die Anwendung soll Textur-Dateien im Bildformat(jpeg, png) laden und verarbeiten können.
- FA5 Die Anwendung soll den hochgeladenen Modellen einen generierten Marker zuordnen.
- FA6 Die Anwendung soll die 3D Modelle im Kamerabild anzeigen können.
- FA7 Die Anwendung soll die 3D-Modelle basierend auf der Transformation und Position der Marker im Kamerabild rendern.

### Nichtfunktionale Anforderungen

- NF1 Die Anwendung soll auf einem Android Smartphone(Huawei P30 Pro) laufen.
  - NF1.1 Die Anwendung soll in Java mit Hilfe von Android Studio entwickelt werden.
- NF2 Das Tracking soll mittels eines markerbasierten Verfahrens realisiert werden.
- NF3 Das Tracking soll mit 30FPS laufen.
- NF4 Das Tracking soll bei vollständig erkanntem Marker robust gegenüber Rotation, Skalierung, Perspektive und Belichtung sein.

NF4 Die Anwendung soll ein simples User-Interface bereitstellen

NF4 Mit Hilfe der UI soll ein Wechsel zwischen Kamera und Einstellungen möglich sein.

NF4 Über die Einstellungsseite soll ein hochladen von 3D-Modellen möglich sein.

# **Kapitel 5**

## **Entwurf**

Dieses Kapitel beschäftigt sich mit dem Entwurf des Prototyps. Die Aufgabe des Entwurfs eines Softwaresystems ist es aus den gegebenen Anforderungen softwaretechnische Lösungen zu entwickeln (Balzert, 2011). Die Grundlage bilden dabei die in Kapitel 4 definierten Anforderungen.

# **Kapitel 6**

## **Entwicklung**

### **6.1. Anforderungen**

### **6.2. Konzeptentwurf**

### **6.3. Implementierung**

### **6.4. Tests**

Während der Entwicklung wurde mit Hilfe von ausführlichen Tests die Funktionalität neuer Features sichergestellt und eine ausführliche Testdokumentation angefertigt. Dadurch sollten Fehler, Probleme und mögliche Verbesserungen entdeckt und festgehalten werden.

Das Ziel war es für jedes neue Feature einen Test durchzuführen.

#### **6.4.1. Testdurchführung**

Bei der Testdurchführung wurde darauf Wert gelegt, dass die einzelnen Durchläufe in der selben Testumgebung und unter den gleichen Umständen stattfinden, um eine Vergleichbarkeit zwischen den verschiedenen Versionen herzustellen. So konnte nach jedem Test zusätzlich festgestellt werden, ob sich eventuell die bestehende Eigenchaften im Vergleich zur Vorgängerversion verschlechtert oder verbessert hatten.

Zu diesem Zweck wurde für jeden Durchlauf der in Abbildung 6.1 gezeigte Versuchsaufbau gewählt.

Da der Prototyp in Android Studio entwickelt wurde, war es zudem möglich die Anwendung direkt auf einem Androidendgerät zu testen. Dieses bot die Möglichkeit die Tests in einer realistischen, praxisnahen Umgebung durchzuführen und demzufolge noch bessere Erkenntnisse über die Alltagstauglichkeit der neuen Features zu erhalten.

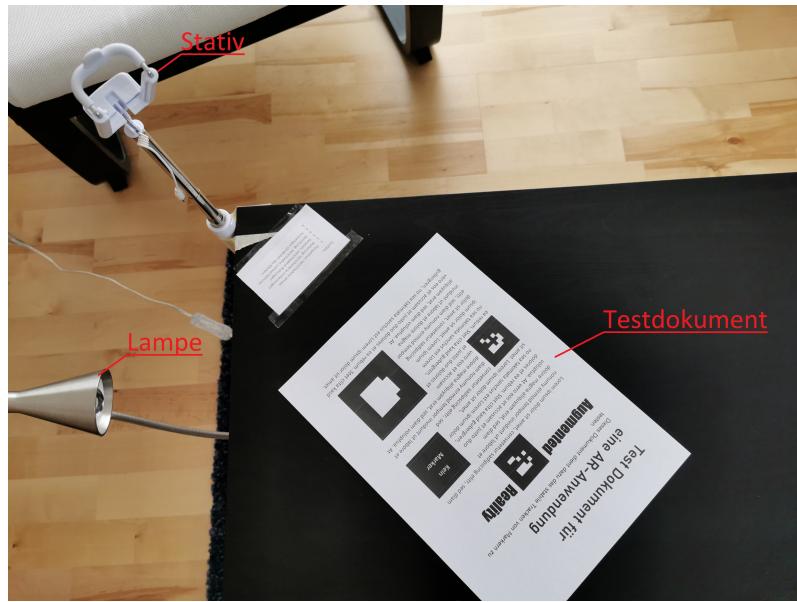


Abbildung 6.1.: Der Testaufbau für jeden Durchlauf. (Quelle: Eigene Darstellung)

Dafür wurde bei jedem Durchlauf ein Huawei P30 Pro genutzt und das Tracking wurde anhand eines dafür angefertigten Dokumentes getestet. Dieses Dokument wurde im Laufe der Entwicklung an die neuen Features angepasst und optimiert. Grundlegend bildet das Dokument einen oder mehrere Marker ab. Gegebenenfalls sind die Marker in Textpassagen eingebunden, um den Schwierigkeitsgrad für die Markererkennung zu erhöhen.

Verweis auf  
Testdoku-  
ment(Anhang)

Während des Testlaufs wurden jedes mal vier Testfälle durchlaufen, die in Abschnitt 6.4.2 beschrieben werden. Dabei wurden sowohl die Neuerungen getestet, als auch Veränderungen in den bereits bestehenden Features festgehalten.

Jeder Testfall wurde dabei mit Hilfe der in Android enthaltenen Funktion „Bildschirmrekorder“ aufgezeichnet und in dem entsprechenden Testbericht dokumentiert.

#### 6.4.2. Testfälle

Die Testfälle beruhen auf den Eigenschaften des SIFT-Algorithmus, welcher Bildmerkmale extrahiert, die invariant gegenüber Rotation, Translation, Skalierung und partiell invariant gegenüber Helligkeitsveränderungen sind (Nischwitz u. a., 2011, S. 345). Mithilfe des Algorithmus können dieselben Objekte in zwei verschiedenen Aufnahmen wiedererkannt werden.

Da auch beim Marker Tracking ein ähnliches Verfahren angewandt werden muss, sind auch hierbei die genannten Eigenschaften relevant. Deshalb wurden die folgenden Testfälle gewählt, um die Funktionen der Anwendung zu testen:

- Perspektivische Invarianz: Dieser Testfall diente dazu das Tracking aus verschiedenen Perspektiven zu testen. Dazu wurde die Kamera auf das Testdoku-

ment gerichtet und anschließend der Winkel zum Dokument so verändert das verschiedenen, perspektivische Verzerrungen der Marker erzeugt wurden.

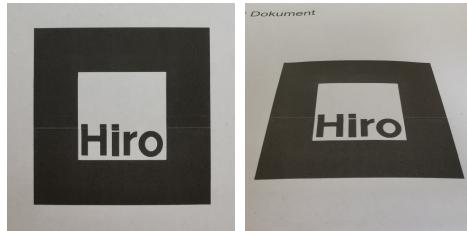


Abbildung 6.2.: Perspektivische Verzerrung eines Markers. (Quelle: Eigene Darstellung)

- Skalierungsinvarianz: Dieser Testfall diente dazu das Tracking aus verschiedenen Entfernung zu testen. Dazu wurde die Kamera langsam auf das Testdokument zu- und weg bewegt, um verschiedenen Markergrößen bzw. -auflösungen zu erhalten.



Abbildung 6.3.: Verschiedene Skalierungen eines Markers. (Quelle: Eigene Darstellung)

- Rotationsinvarianz: Dieser Testfall diente dazu das Tracking von Markern mit unterschiedlichen Rotationen zu testen, dazu wurde die Kamera auf das Dokument gerichtet und anschließend wurde letzteres langsam rotiert, um zu evaluieren, ob die Marker auch mit unterschiedlichen Rotationen korrekt erkannt werden.

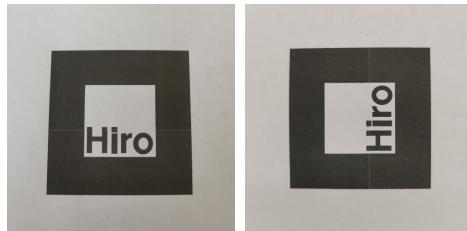


Abbildung 6.4.: Verschiedene Rotationen eines Markers. (Quelle: Eigene Darstellung)

- Belichtungsinvarianz: Dieser Testfall diente dazu das Tracking von Markern gegenüber unterschiedlichen Belichtungen zu testen. Dazu wurde mit Hilfe einer Lampe die Belichtung des Testdokumentes verändert.



Abbildung 6.5.: Verschiedene Belichtungen eines Markers. (Quelle: Eigene Darstellung)

- Trackinggeschwindigkeit: Dieser Testfall sollte die Geschwindigkeit des Trackings, sowie die Robustheit testen. Dazu wurden ein oder mehrere Marker kurzzeitig mit der Hand verdeckt, um zu testen, ob anschließend alle Marker wieder erfolgreich getrackt wurden und wie schnell dieses erfolgte.

Besserer Name

All diese Testfälle beziehen sich auf das Marker Tracking, während dessen konnten jedoch auch das Rendern des Modells und weitere Features die nicht direkt mit dem Tracking zusammenhängen getestet werden.

#### 6.4.3. Zusammenfassung

- Belichtung nicht einfach zu testen - starke Belichtungswechsel teilweise kurze Aussetzer

Am Ende schreiben

# **Kapitel 7**

## **Evaluation**

# **Kapitel 8**

## **Fazit**

**8.1. Fazit**

**8.2. Ausblick**

# Literaturverzeichnis

[Balzert 2011] BALZERT, H.: *Lehrbücher der Informatik*. Bd. 1: *Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb*. 3. Spektrum Akademischer Verlag, 2011. – ISBN 978-3-8274-1706-0

[Böhm und Fuchs 2002] BÖHM, R. ; FUCHS, E.: *Wirtschaftsinformatik*. Bd. 1: *System-Entwicklung in der Wirtschaftsinformatik*. 5. vdf Hochschulverlag, 2002. – ISBN 978-3-7281-2762-4

[Hedberg u. a. 2018] HEDBERG, H. ; NOURI, J. ; HANSEN, R.: A Systematic Review of Learning Through Mobile Augmented Reality. In: *International Journal of Interactive Mobile Technologies* 12 (2018), Nr. 3, S. 75–85. – URL <https://www.online-journals.org/index.php/i-jim/article/view/8404>

[Kind u. a. 2019] KIND, H. ; FERDINAND, J. ; JETZKE, T. ; RICHTER, S. ; WEIDE, S.: Virtual und Augmented Reality: Status quo, Herausforderungen und zukünftige Entwicklungen / TAB - Büro für Technikfolgen-Abschätzung beim Deutschen Bundestag. 2019. – TAB-Arbeitsbericht

[Klein 2006] KLEIN, G.: *Visual Tracking for Augmented Reality*, University of Cambridge, Dissertation, 2006

[Mehler-Bicher und Steiger 2014] MEHLER-BICHER, A. ; STEIGER, L.: *Augmented Reality - Theorie und Praxis*. Bd. 1. 2. De Gruyter Oldenbourg, 2014. – ISBN 978-3-11-035385-3

[Milgram u. a. 1994] MILGRAM, P. ; TAKEMURA, H. ; UTSUMI, A. ; KISHINO, F.: Augmented Reality: A class of displays on the reality-virtuality continuum. In: *Telemanipulator and Telepresence Technologies* Bd. 2351, SPIE, 1994, S. 282–292

[Nischwitz u. a. 2011] NISCHWITZ, A. ; FISCHER, M. ; HABERÄCKER, P. ; SOCHER, G.: *Computergrafik und Bildverarbeitung - Band II: Bildverarbeitung*. Bd. 2. 3. Vieweg+Teubner Verlag, 2011. – ISBN 978-3-8348-1712-9

[Owen u. a. 2002] OWEN, C. ; XIAO, F. ; MIDDLIN, P.: What is the best fiducial?, 2002, S. 8 pp.. – ISBN 0-7803-7680-3

[Robertson und Robertson 2012] ROBERTSON, S. ; ROBERTSON, R.: *Mastering the Requirements Process: Getting Requirements Right.* Bd. 1. 3. Pearson Education, 2012. – ISBN 978-0-321-81574-3

[Winter 2018] WINTER, Andreas: *SRS Anforderungen.* Vorlesung Softwaretechnik I. 2018

# **Anhang A**

## **Beispielanhang**

Beispiel für einen Anhang!

# **Erklärung**

Hiermit versichere ich, Johannes Scheibe, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe.

---

Johannes Scheibe