

Thesis

zur Erlangung des Grades

Master

im Studiengang Medieninformatik Master

an der Fakultät Digitale Medien

Entwicklung einer Augmented Reality App zur Visualisierung historischer Gebäude der Stadt Villingen

Erstbetreuer: Prof. Dr. Uwe Hahne

Zweitbetreuer: Prof. Dr. Thomas Schneider

Abgegeben am: 31.08.2022

Abgegeben von: Oliver Kusch

Matrikelnummer: 268513

Erbsenlachen 6, 78050 Villingen-Schwenningen

oliver.kusch@hs-furtwangen.de

Abstract

Durch das Projekt NISABA¹ und der Veranstaltung Bildverarbeitung und Computergrafik im Studiengang Medieninformatik Master im Sommersemester 2021 sind 3D-Modelle der Kasernengebäude des Lyautey- und Mangin-Geländes entstanden. Um die fertigen 3D Modelle für jeden zugänglich und auf moderner Weise präsentieren zu können, wird in dieser Master-Arbeit eine Augmented Reality Anwendung für Smartphone und Tablet Geräte entwickelt.

Diese Forschungsarbeit beschäftigt sich mit der Entwicklung einer Augmented Reality AR Anwendung. Durch genaues Tracking und eine realistische Darstellung der Gebäude durch wetterspezifische Belichtung, Schattierung und Spiegelung soll eine hohe Immersion erschaffen werden. In der Anwendung werden die 3D-Objekte in der Größendarstellung 1:1 über das Videobild der Kamera gelegt und der Nutzer kann vor Ort das Gebäude in Echtzeit ein- und ausblenden lassen. Dafür wird der Begriff *world-scale Augmented Reality* herangezogen. Bekannte Beispiele sind das AR Spiel Pokemon GO² oder auch Anwendungen für Googles Holo Lens³.

Eine wesentliche Rolle bei der Immersion in AR spielt das Tracking, bei dem kontinuierlich die Positions- und Rotationsdaten des Endgeräts erfasst werden. Es gibt mehrere Verfahren, um die Position und Orientierung der Kamera relativ zur Umgebung zu bestimmen. Im begrenzten Räumen ist das Tracking durch einheitlichere Belichtung und vorhandenen Kanten und Flächen mit kamerabasierten Tracking Methoden gut umsetzbar. Im Freien kann das Tracking je nach Umgebung Probleme bereiten. Schwierigkeiten entstehen durch unterschiedliche Lichtverhältnisse und dem größeren Suchraum. Hinzu kommt, dass durch die große Distanz zwischen Kamera und virtuellem Objekt die Platzierung des 3D-Modells bereits durch kleine Bewegung der Kamera stark von der korrekten Position abweicht. Das Objekt erscheint nicht homogen in der realen Welt, wodurch die Immersion beeinträchtigt wird.

Klassischerweise wird bei AR im Freien GPS und die Neigungs- und Beschleunigungssensoren der Smartphones genutzt, um die Kameraposition und -orientierung zu ermitteln. Wie Platinsky und seine Koautoren[9] bereits erörtert ist die GPS Lokalisierung insbesondere in Städten mit Störfaktoren wie Gebäuden oder Vegetation ungenau. Deshalb wird in dieser Arbeit auf Methoden eingegangen, um die Genauigkeit der Position und der Orientierung der Kamera im Freien zu verbessern.

Um die Immersion bei der Nutzung der App zu steigern, werden die Wetterbedingungen bei der Darstellung der 3D-Modelle berücksichtigt. So sollen die Fassaden bei regnerischem Wetter dunkler und gegebenenfalls spiegelnd dargestellt werden. Die Schattierungen sollen sich anpassen, indem bei hartem Licht (z.B. durch starke Sonneneinstrahlung) auch harte Schatten und bei weichem Licht

¹<https://nisaba.dm.hs-furtwangen.de/>

²<https://pokemongolive.com/de/>

³<https://www.microsoft.com/de-de/hololens>

(z.B. bei einer dichten Wolkendecke) weiche Schatten dargestellt werden.

Inhaltsverzeichnis

Zusammenfassung	3
1 Hintergrund der Arbeit	8
2 Aufgabenstellung	8
2.1 Fragestellungen / Problemstellungen	8
3 Forschungsstand	8
3.1 Ähnliche Arbeiten	8
4 Tracking	8
4.1 Koordinatensysteme	9
4.2 Kamera-basiertes Tracking	9
4.3 Marker Tracking	9
4.4 Marker-less Tracking	9
4.5 Algorithmen zur Erkennung von Merkmalen	9
4.5.1 SIFT	9
4.5.2 SURF	9
4.5.3 ORB	9
4.5.4 Merkmalerkennung in AR Foundation / AR Core	9
4.6 SLAM	9
4.7 SLAM in ARFoundation / AR Core	9
4.8 GPS Tracking	9
5 Computergraphik Shader	9
5.1 Rendering Pipeline	9
5.2 Darstellung von Wetterphänomenen in der Computergraphik	9
5.3 Erkennung der Lichtsituation in AR Foundation / AR Core	9
5.4 Spiegelungen, Raytracing(?)	9
6 Entwicklung	10
6.1 Methodiken in der Planung	10
6.2 Frameworks	10
6.3 (Vergleich vorhandener Frameworks)	10
6.4 Entwicklungsumgebung	10
6.5 Aufbau der Anwendung	10

7 Nutzertest bzw. Evaluierung	10
7.1 Aufbau des Tests	10
7.2 Durchführung	10
7.3 Ergebnisse/Evaluierung	10
8 Fazit der Arbeit	10
9 Ausblick	10
10 Forschungsstand	11
10.1 Tracking	11
10.1.1 Koordinatensysteme	11
10.2 Kamera-basiertes Tracking	12
10.2.1 GPS Tracking	12
10.3 Software Entwicklung	13
11 Fragestellungen und Methodik	14
Abbildungsverzeichnis	16
Eidesstattliche Erklärung	17

1 Hintergrund der Arbeit

Ein kurzes Kapitel über die vorangegangenen Projekte, Verweise auf Projekt NISABA etc.

Welche Gebäude sind vorhanden? Bestandsaufnahme vorhandener Gebäude.

2 Aufgabenstellung

Die Aufgabe besteht darin die im vorhergegangenen Arbeiten entstandenen 3D Modelle des SABA Gebäudes, des Lyautley- und des Mangin-Geländes mithilfe einer AR Applikation zu visualisieren. Ziel ist es eine App zu entwickeln, die eine hohe Immersion bietet. Daher wird zunächst der Begriff der Immersion in AR-Anwendungen definiert. Die Zielgruppe sind Bewohner der Stadt Villingen und interessierte Personen wie Touristen, die mehr über die Geschichte der Stadt Villingen erfahren möchten. Die Anwendung wird vor Ort genutzt, da die Gebäude im Größenverhältnis 1:1 dargestellt werden. So hat der Nutzer ein direktes Verständnis über die Größenverhältnisse der Gebäude. Die 3D Modelle sollen an das Umgebungslicht und der herrschenden Wetterbedingungen angepasst sein. Das bedeutet, dass bei hartem Licht die Modelle auch harte Schattenkanten werfen. Bei regnerischen Wetter soll die Fassade dunkler wirken, diffuse Schatten werfen und eventuell Spiegelungen darstellen.

2.1 Fragestellungen / Problemstellungen

Was sind die Probleme in der Entwicklung von AR Anwendungen im Freien?

- GPS ungenauigkeit
 - größere Entfernung -> genaueres Tracking benötigt
- Realistisches Umgebungslicht
Wetterphänome und Darstellung in Computergraphik

3 Forschungsstand

3.1 Ähnliche Arbeiten

Outdoor AR Projekte

Wie wurden die Probleme gelöst?

4 Tracking

Definition, was ist mit dem Begriff gemeint?

4.1 Koordinatensysteme

4.2 Kamera-basiertes Tracking

4.3 Marker Tracking

4.4 Marker-less Tracking

4.5 Algorithmen zur Erkennung von Merkmalen

4.5.1 SIFT

4.5.2 SURF

4.5.3 ORB

4.5.4 Merkmalerkennung in AR Foundation / AR Core

4.6 SLAM

Mathematische Definition des SLAM Problems und bekannte Arbeiten zur Lösung des Problems.

4.7 SLAM in ARFoundation / AR Core

4.8 GPS Tracking

5 Computergraphik Shader

Darstellung von Wetterphänomenen (hartes/diffuses Licht, Spiegelungen, Raytracing?) - Raytracing in mobiltelefon in Unity machbar?

5.1 Rendering Pipeline

5.2 Darstellung von Wetterphanomenen in der Computergraphik

5.3 Erkennung der Lichtsituation in AR Foundation / AR Core

5.4 Spiegelungen, Raytracing(?)

Wie werden Spiegelungen erzeugt?

Raytracing, auch in Smartphones schon möglich?

6 Entwicklung

6.1 Methodiken in der Planung

UML, Agile Softwareentwicklung

6.2 Frameworks

6.3 (Vergleich vorhandener Frameworks)

Was gibt es auf dem Markt (AR Foundation, AR Core, AR Kit, Vuforia, Kudan, Wikitude), Vorteile/Nachteile und geeignetes Framework für die Entwicklung dieser App.

6.4 Entwicklungsumgebung

Ein paar beschreibende Worte über AR Foundation und Unity.

Welche Sprache wird genutzt?

Wie erfolgt die Entwicklung? Code schreiben, Debugging, Smartphone anschließen

Was muss für die Entwicklung installiert werden?

6.5 Aufbau der Anwendung

Beschreibung des Aufbaus der App, also die UML Grafik

7 Nutzertest bzw. Evaluierung

7.1 Aufbau des Tests

7.2 Durchführung

7.3 Ergebnisse/Evaluierung

8 Fazit der Arbeit

9 Ausblick

10 Forschungsstand

Augmented Reality ist ein nachgefragtes Thema und wird in unterschiedlichen Bereichen wie Industrie, Medizin oder in der Computerspiel-Branche untersucht[11][10][6]. Es gilt Probleme zu lösen, die sowohl Hardware als auch Software betreffen. Die Hardwareentwicklung betrifft dabei hauptsächlich die Mobilität bei der Nutzung. Beispielsweise funktioniert die HTC Vive nur mit einem Computer, der mit einem Kabel am Headset verbunden ist. Die Microsoft Holo Lens hingegen benötigt eine gute Internetverbindung über Wi-Fi. In der Software ist es eine Herausforderung aus den vielfältigen Software-Paketen eigene AR Anwendungen zu entwickeln. Nicht jede Software Bibliothek ist miteinander kompatibel, sodass die Entwicklung eine lange Einarbeitungszeit benötigt[Vgl. 10].

Ein weiteres Problem in der Nutzung von AR ist die Beleuchtung der Umgebung. In Fabriken oder Lagern ist es dunkel und Lichter kommen aus verschiedenen Richtungen, während es im Freien die Wetterbedingungen wie starke Sonne oder Wolken, die störende Schatten verursachen, zu beachten gilt. Die unterschiedlichen Lichtverhältnisse rufen Ungenauigkeiten z.B. beim Tracking hervor.

10.1 Tracking

Der Begriff Tracking beschreibt die kontinuierliche Verfolgung von Positions- und Rotationsdaten, die von Eingabegeräten (z.B. VR Controller) oder Sensoren (z.B. *Inertial Measurement Units* (Gyroskop und Beschleunigungssensoren)) erfasst werden. Die Bewegung eines starren Körpers kann „durch die Angabe von sechs Werten (drei Koordinaten als Position und drei Winkel zur Beschreibung der Orientierung) für jeden Zeitschritt spezifiziert werden“[4, Dörner (2019) S.119f.].

Die Werte für die Position und Orientierung für ein Objekt werden als *Freiheitsgrade* (engl. *Degrees of Freedom – DOF*) bezeichnet. Beim Tracking ist es das Ziel, die sechs Freiheitsgrade für die Translation und Rotation der Kamera zu bestimmen bzw. zu schätzen[4]. Es gibt zwei Tracking Systeme. Beim *Inside-Out-Tracking* befinden sich Sensoren im Objekt, das getrackt werden soll, während beim *Outside-In-Tracking* sich die Sensorik in der Umgebung befinden und das Objekt von außen getrackt wird. In dieser Arbeit wird ein Inside-Out-Tracking verwendet, da sich die Sensorik im Smartphone bzw. Tablet befinden.

10.1.1 Koordinatensysteme

Um eine Bestimmung bzw. Schätzung der Translationen und Rotationen durchzuführen, können zwei Koordinatensysteme herangezogen werden. Ein Kamerakoordinatensystem und ein Objektkoordinatensystem, womit die relativen Transformationen zwischen den Koordinatensystemen bestimmt werden können. Weiterhin gibt es die Möglichkeit, dass für alle Objekte im Raum ein

Koordinatensystem (Weltkoordinatensystem) verwendet wird. Voraussetzung für das Tracking ist, dass die Transformationen zwischen den Objekten bekannt sind. Dann kann die Transformation zwischen der Kamera und das Weltkoordinatensystem geschätzt werden. Sind einzelne Transformationen von Objekten im Weltkoordinatensystem nicht bekannt, kann es auch Mischformen geben Zitat [4, Dörner (2019) S.124f.].

10.2 Kamera-basiertes Tracking

Kamera-basiertes Tracking nutzt Informationen zu Objekten aus dem Video Datenstrom, um die relative Position und Orientierung der Objekte zur Kamera zu bestimmen. Hartley und Zisserman bezeichnen diese als (extrinsische Kameraparameter) [5, Hartley, Zisserman (2003) S.156]. Für das Kamera-basierte Tracking wird zwischen Marker-basierten und Marker-less Tracking unterschieden.

Beim Marker-basierten Tracking werden schwarz-weiß Marker (sogenannte Kanji und Hiro Marker) (Verweis) mit einfachen geometrischen Formen platziert und erkannt, sodass diese als Orientierungshilfen fungieren. Eine Anwendung von NGIN-Mobility arbeitet mit Markern, die auf dem Boden platziert werden können ⁴. Da die Marker in dauerhaft in der Umgebung platziert werden müssen, ist die Trackingmöglichkeit für die Nutzung im Freien ungeeignet.

Die zweite Möglichkeit erfasst über Algorithmen der Computer Vision Merkmale in der realen Umgebung. Diese Art wird als *marker-less AR* bezeichnet. Merkmale können geometribasiert sein (z.B. Kanten, Formen wie Vierecke) oder es werden Detektoren wie z.B. *SIFT* (engl. *Scale Invariant Feature Transform*)[8] oder *SURF* (engl. *Scale Invariant Feature Transform*)[2] verwendet. Auch die aus der Robotertechnik bekannte *SLAM* (engl. *Simultaneous Localization and Mapping*)[3] [1] Methode wird genutzt. Dabei werden entweder kamerabasiert (Visual SLAM) nach Merkmalen gesucht oder es kommen Sensoren zur Generierung von Tiefeninformationen zum Einsatz, z.B. Kinect ⁵ oder *LIDAR-Sensoren* (engl. *light detection and ranging*)[7]. Vorteil dieser Methode ist, dass gleichzeitig eine 3D-Karte des Raumes generiert wird, die immer wieder zur Positions- und Rotationsbestimmung verwendet werden kann.

10.2.1 GPS Tracking

Im Außenbereich wird bei AR auch GPS für das Tracking herangezogen. Dabei sind Positionsabweichungen von bis zu 10 Metern möglich, sodass eine genaue Bestimmung der extrinsischen Kameraparameter beeinträchtigt wird. Um die Tracking-Genauigkeit zu erhöhen, gibt es mehrere Methoden. *DGPS* (engl. *Differential GPS*) verbessert GPS-Signale, indem es ein Korrektursignal durch eine ortsfeste Referenzstation mit bekannter Lokalisierung berechnet. Da es in Deutschland

⁴<https://www.logistik-watchblog.de/startups/1479-insider-navigation-ar-loesung-lagerhalle.html>

⁵<https://developer.microsoft.com/de-de/windows/kinect/>

lediglich acht solcher Stationen gibt und einige Anbieter nur kommerziell die Daten bereitstellen, ist diese Methode nicht für diese Arbeit geeignet^{6 7}. Eine weitere bekannte Möglichkeit bietet *SBAS* (engl. *Satellite Based Augmentation System*), bei dem mehrere geostationäre Satelliten das GPS Signal auf bis zu einem Meter Genauigkeit zu verbessern [4].

Platinsky und seine Koautoren[9] erstellen für ein besseres Tracking bei fehlender GPS Genauigkeit ein 3D-Modell der Umgebung. Bei der anschließenden AR Nutzung in diesem Gebiet wird auf dem Smartphone SLAM betrieben. Die Daten vom Smartphone werden mit der 3D-Karte verglichen, um ein genaueres Tracking durchzuführen. Ein ähnliches System wäre für die Anwendung in dieser Arbeit denkbar, da eine große Datenmenge von Bildern des Geländes vorhanden ist. Über Structure from Motion Methoden, kann mit den Bildern eine große 3D-Karte erstellt werden.

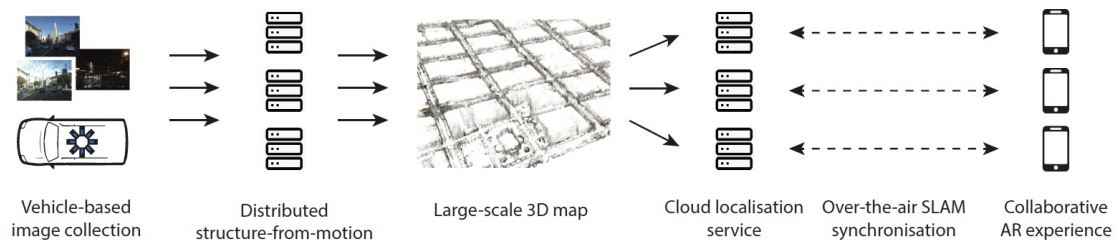


Abbildung 1: Das Grundprinzip der Methode von Platinsky.

10.3 Software Entwicklung

Für Augmented Reality Anwendungen gibt es im Smartphone und Tablet Bereich mehrere Softwarepakete (*SDK* (engl. *Software Development Kit*)). Die bekanntesten sind ARKit von Apple, das nur auf iOS Endgeräten läuft und ARCore von Google, das für Android Endgeräte entwickelt wird.⁸

⁹ Andere SDK's wie Vuforia¹⁰, Wikitude¹¹, ARToolKit¹² oder Lightship¹³ sind unabhängig vom Endgerät nutzbar.

⁶<https://www.heise.de/newsticker/meldung/Differential-GPS-und-WLAN-RTT-Präzise-Ortung-mit-Android-P-4046935.html>

⁷Liste von DGPS-Sendern: <https://www.ndblist.info/datamodes/worldDGPSdatabase.pdf>

⁸<https://developer.apple.com/augmented-reality/>

⁹<https://developers.google.com/ar>

¹⁰<https://www.ptc.com/en/products/vuforia>

¹¹<https://www.wikitude.com/>

¹²<http://www.hitl.washington.edu/artoolkit/>

¹³<https://lightship.dev/>

11 Fragestellungen und Methodik

Für die Entwicklung der AR Anwendung können die vorgestellten SDK's genutzt werden. Diese haben Vor- und Nachteile, die in dieser Arbeit erörtert werden, um daraus die passende SDK für dieses Projekt auszuwählen. Die Methode von Platinsky und seinen Koautoren[9], in der das Tracking durch eine vorgefertigte 3D-Karte der Stadt verbessert wird, wird in dieser Arbeit weiter untersucht. Im Beispiel von Platinsky ist die GPS Genauigkeit durch die hohen Gebäude in der Großstadt stark beeinträchtigt. Das Gelände in Villingen befindet sich am Stadtrand und die Gebäude könnten keinen großen Einfluss auf die GPS Genauigkeit haben. Daher wird untersucht, ob der Mehraufwand, der durch die Erstellung der 3D Map und des SLAM Systems über eine Cloud entsteht, für eine mittelgroße Stadt sinnvoll ist. Hierfür wird die GPS Genauigkeit mit und ohne dieser Methode gemessen und miteinander verglichen.

Hinzu kommt die Frage, wie detailreich die 3D-Karte und die SLAM Daten des Smartphones für ein genaues Tracking sein müssen. Ein Problem der Methode ist, dass eine konstante Verbindung zum Internet bestehen muss, um die SLAM Daten mit der 3D-Karte zu vergleichen. Die 4G Verbindung war nicht ausreichend schnell [9, sinngemäß aus]. Um an diesem Problem anzuknüpfen, wird untersucht, wie stark die Qualität der 3D-Karte reduziert werden kann, ohne die Vorteile beim Tracking zu verlieren. Eine Optimierung der Dateigröße der 3D-Karte wird im Paper zwar erwähnt, jedoch nicht umgesetzt. So wird in dieser Arbeit ein Experiment durchgeführt, bei der drei Qualitätsstufen (Hohe Details, mittlere Details, grobe Details) auf die GPS Genauigkeit untersucht werden.

Je nachdem wie die Ergebnisse der Experimente ausgehen, ist es für die Anwendung sinnvoll dem Nutzer zwei Möglichkeiten anzubieten: Eine hohe Tracking Genauigkeit bei hohem Datenverbrauch und eine möglicherweise geringe Tracking Genauigkeit, bei dem diese Methode nicht genutzt wird.

Abbildungsverzeichnis

1	Das Grundprinzip der Methode von Platinsky.	13
---	-----------------------------------------------------	----

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Thesis selbständig und ohne unzulässige fremde Hilfe angefertigt habe. Alle verwendeten Quellen und Hilfsmittel sind angegeben.

Villingen-Schwenningen, 31.08.2022

Oliver Kusch

Literatur

- [1] T. Bailey und H. Durrant-Whyte. „Simultaneous localization and mapping (SLAM): part II“. In: *IEEE Robotics and Automation Magazine* 13 (3 Sep. 2006), S. 108–117. ISSN: 1070-9932. DOI: 10.1109/MRA.2006.1678144.
- [2] Herbert Bay u. a. „Speeded-Up Robust Features (SURF)“. In: *Computer Vision and Image Understanding* 110 (3 Juni 2008), S. 346–359. ISSN: 10773142. DOI: <https://doi.org/10.1016/j.cviu.2007.09.014>.
- [3] H. Durrant-Whyte und T. Bailey. „Simultaneous localization and mapping: part I“. In: *IEEE Robotics and Automation Magazine* 13 (2 Juni 2006), S. 99–110. ISSN: 1070-9932. DOI: 10.1109/MRA.2006.1638022. URL: <https://ieeexplore.ieee.org/document/1638022/>.
- [4] Ralf Dörner u. a. *Virtual und Augmented Reality (VR/AR)*. Bd. 2. Springer Berlin Heidelberg, 2019. ISBN: 978-3-662-58860-4. DOI: 10.1007/978-3-662-58861-1.
- [5] Richard Hartley und Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Bd. 2. Cambridge University Press, März 2004. ISBN: 9780521540513. DOI: 10.1017/CB09780511811685.
- [6] Bo-Chen Huang u. a. „ARBIN: Augmented Reality Based Indoor Navigation System“. In: *Sensors* 20 (20 Okt. 2020), S. 5890. ISSN: 1424-8220. DOI: 10.3390/s20205890.
- [7] Weiquan Liu u. a. „Learning to Match 2D Images and 3D LiDAR Point Clouds for Outdoor Augmented Reality“. In: IEEE, März 2020, S. 654–655. ISBN: 978-1-7281-6532-5. DOI: 10.1109/VRW50115.2020.00178.
- [8] David G. Lowe. „Object recognition from local scale-invariant features“. In: IEEE, 1999, 1150–1157 vol.2. ISBN: 0-7695-0164-8. DOI: 10.1109/ICCV.1999.790410.
- [9] Lukas Platinsky u. a. „Collaborative Augmented Reality on Smartphones via Life-long City-scale Maps“. In: (Nov. 2020). URL: <http://arxiv.org/abs/2011.05370>.
- [10] Gian Maria Santi u. a. „Augmented Reality in Industry 4.0 and Future Innovation Programs“. In: *Technologies* 9 (2 Apr. 2021), S. 33. ISSN: 2227-7080. DOI: 10.3390/technologies9020033.
- [11] Sud Sudirman und Abdenmour El-Rhalibi. „Improving Camera Pose Estimation for Indoor Marker-less Augmented Reality“. In: IEEE, Okt. 2015, S. 994–999. ISBN: 978-1-5090-0154-5. DOI: 10.1109/CIT/IUCC/DASC/PICOM.2015.150.