



Konzeption und Umsetzung eines Augmented Reality basierten Assistenzsystems zur Unterstützung industrieller Prozesse

BACHELORARBEIT

für die Prüfung zum

Bachelor of Science

des Studienganges Informatik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

Mikka Jenne

Abgabedatum 31. August 2020

Bearbeitungszeitraum

12 Wochen

Matrikelnummer

2062885

Kurs

TINF17B4

Ausbildungsfirma

cjt Systemsoftware AG

Karlsruhe

Betreuer der Ausbildungsfirma

M. Sc. Florian Dunz

Gutachter der Studienakademie

Prof. Dr. Marcus Strand

Erklärung

Ich versichere hiermit, dass ich meine Bachelorarbeit mit dem Thema: *“Konzeption und Umsetzung eines Augmented Reality basierten Assistenzsystems zur Unterstützung industrieller Prozesse”* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum

Unterschrift

Sperrvermerk

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anders lautende Genehmigung der Ausbildungsstätte vorliegt.

Zusammenfassung

Computer sind heutzutage allgegenwärtig.

Kaum ein Unternehmen arbeitet ohne computergesteuerte Unterstützung, sei es das Schreiben einer Rechnung in einem mittelständischen Unternehmen, das Verwalten von einzelnen Arbeitsprozessen oder die Unterstützung bei einzelnen Arbeitsschritten. Speziell im Elektrotechnischen Bereich gibt es heute noch Prozesse die hinsichtlich der physischen und digitalen Welt stark voneinander getrennt sind, z.B. das Zeichnen von Schaltplänen und die digitale Erfassung von Messwerten, die über Programme verwaltet werden können.

Das Ziel dieser Arbeit ist es, die physische Welt durch die Möglichkeit der digitalen Zeichnung von Gebäudeschaltplänen zu erweitern. Für den Gebrauch designed, wird auf eine Verbesserung sowie auf die Transparenz von solchen Schaltplänen gezielt, um so weitere essentielle Informationen zu Gebäuden zu erhalten. Da die händische Zeichnung meist aufwändig, kostenintensiv ist und keiner Norm entspricht, soll durch die Digitalisierung dieses Prozesses Abhilfe geschaffen werden. Durch Nutzung neuester Technologien im Bereich Desktopanwendungen und des Einsatzes einer einheitlichen Zeichenstruktur sollte diese Applikation für den Otto Normalverbraucher leicht nutzbar sein.

Abstract

Inhaltsverzeichnis

1	Einleitung	7
1.1	Motivation	7
1.2	cjt Systemsoftware AG	10
1.3	Aufgabenstellung	11
1.4	Aufbau der Arbeit	12
1.5	Stand der Technik	13
2	Grundlagen	14
2.1	Augmented Reality	14
2.1.1	Virtual Reality, Augmented Reality und Mixed Reality	16
2.1.2	Varianten der Augmented Reality	18
2.1.3	Positionsbestimmung	19
2.1.4	Augmented Reality in der Industrie	21
2.2	SLAM - Simultaneous Localization And Mapping	22
2.2.1	Localization	22
2.2.2	Mapping	22
2.3	Quaternionen	22
2.3.1	Transformation	22
2.3.2	Rotation	22
2.4	Technologien	22
2.4.1	Google ARCore	22
2.4.2	Android Jetpack	25
2.4.3	Sceneform SDK	28
2.4.4	SQLite	29
2.5	Softwarearchitektur	30
2.5.1	Modulare Software Architektur	31
2.5.2	MVVM	32
2.5.3	Android Architecture Components	33
2.6	Datenmodellierung	35
3	Konzeption	37
3.1	Arbeitsumgebung / Umfeld	37
3.2	Objekterkennung / Scan-Phase	37
3.3	Visualisierungs-Phase	37
3.4	Architekturkonzept	37
3.5	Softwarekonzept	37

3.6	Auswahl des AR Frameworks	37
3.6.1	Google ARCore	37
3.6.2	ARToolKit	37
3.7	Datenmodell	37
4	Umsetzung	38
4.1	Implementierung	38
4.1.1	Umgebungserkennung / Scan - Lernphase	38
4.1.2	Visualisierungs-Phase	38
4.2	Testdurchlauf - Test-Szenario	38
5	Evaluation	39
6	Fazit	40
7	Ausblick	41
	Anhang	42
	Index	42
	Literaturverzeichnis	42

Abbildungsverzeichnis

2.1	Mobile-AR in Snapchat	19
2.2	Marker-basierte Augmented Reality Positionsbestimmung	20
2.3	Skizze zur Bewegungsverfolgung des Smartphones [ARCORE 2020]	24
2.4	Grundprinzip einer IMU [BÖTTCHER 2020]	24
2.5	Skizze zum Umgebungsverständnis des Smartphones [ARCORE 2020]	25
2.6	Skizze zu den Lichtverhältnissen des Smartphones [ARCORE 2020]	25
2.7	Komponenten von Android Jetpack [SELLS, POIESZ und NG 2018]	26
2.8	Room Architektur Diagramm [DEVELOPERS 2017b]	27
2.9	ViewModel Struktur Diagramm [CODELABS 2020]	28
2.10	MVC Architektur Diagramm [<i>ModelView Controller</i> 2020]	31
2.11	MVVM Architektur Diagramm [JAECKLE, GOLL und DAUSMANN 2015]	32
2.12	MVVM Informationsfluss [RICARDO 2019]	33
2.13	Android Grundkomponente [<i>Architecture Components</i> 2020]	34
2.14	ViewModel Komponente [<i>Architecture Components</i> 2020]	34
2.15	Repository Komponente [<i>Architecture Components</i> 2020]	34
2.16	Model / Room Komponente [<i>Architecture Components</i> 2020]	35
2.17	Android Architecture Components Diagram [<i>Architecture Components</i> 2020] . . .	36

Tabellenverzeichnis

2.1	Merkmale der Computergraphik gegenüber der VR [DÖRNER 2019]	17
-----	---	----

Liste der Code-Beispiele

Abkürzungsverzeichnis

API	Application Programming Interface	
AR	Augmented Reality	17
IOSB	Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung IOSB	
GPS	Global Positioning System	
HMD	Head-Mounted Display	15
IMU	Inertial Measurement Unit	23
IoT	Internet of Things, dt. Internet der Dinge	9
INS	Inertial Navigation System	21
KIT	Forschungszentrum Karlsruhe	10
MEMS	Micro-Electro-Mechanical Systems	23
MR	Mixed Reality	18
MVC	Model View Controller	30
SDK	Software Development Kit	22
SLAM	Simultaneous Localization And Mapping	11
SQL	Structured Query Language	
GUI	Graphical User Interface, dt. Benutzeroberfläche	28
VR	Virtual Reality	17
WMR	Windows Mixed Reality	18

Kapitel 1

Einleitung

In diesem Teil der Arbeit wird auf die Motivation des Themas eingegangen. Darüber hinaus wird sowohl die Aufgabenstellung als auch der Aufbau der Arbeit genauestens dargelegt. Eine nähere Betrachtung des Standes der Technik untermauert die Beweggründe dieser Ausarbeitung.

1.1 Motivation

Jede neu entwickelte Technologie durchlebt im Laufe der Entstehung ein enormes Aufsehen. Es wird viel darüber debattiert, fantasiert und geplant ohne jedoch genau die Resultate abwägen zu können. Durch fehlende Erfahrung und nicht ausgereifte Konzepte werden Highlights erwartet, die zu diesem Zeitpunkt technisch nicht umsetzbar sind. Jede neue technologische Idee durchläuft bestimmte Phasen der Entwicklung. [B12 2020]

Ein sogenannter Hype Cycle, dt. Hype-Zyklus, ist ein visualisiertes Modell, das die Entwicklung einer neuen Technologie von der Innovation über die Umsetzung bis hin zur ausgereiften Marktfähigkeit repräsentiert und so diese Phasen der Entwicklung verdeutlicht.

Der Hype Cycle, entwickelt von der Forschungsgruppe Gartner Inc. und die durch deren Mitarbeiterin Jackie Fenn geprägten Definitionen, ist in fünf Entwicklungsphasen untergliedert:

1. *Technologische Auslösung*: Bekanntwerden der Technologie und erste Projekte, die auf beachtliches Interesse in der Öffentlichkeit stoßen.
2. *Gipfel der überzogenen Erwartungen*: Die Technologie ist auf dem Gipfel der Aufmerksamkeit. Es werden unrealistische und enthusiastische Erwartungen veröffentlicht und diskutiert.
3. *Tal der Enttäuschung*: Da die Technologie nicht die zuvor aufgebauten Erwartungen erfüllen kann, sinkt die Aufmerksamkeit der enttäuschten Enthusiasten und der Medienvertreter auf den Tiefpunkt.
4. *Pfad der Erleuchtung*: Die neue Technologie wird in dieser Phase realistisch mit ihren Stärken und Schwächen betrachtet.
5. *Plateau der Produktivität*: Die neuen Möglichkeiten durch die Technologie werden als Vorteile akzeptiert, und Geschäftsprozesse werden entwickelt. [PINKER 2016]

Nachdem eine Innovation den Gipfel der überzogenen Erwartungen, z.B. die vollständige Revolutionierung der Industrie oder Szenarien, wie z.B. Hologramme, die man nur aus Science-Fiction Filmen kennt, passiert hat, folgt das Tal der Enttäuschung, wobei festgestellt wird, dass die Erwartungen nicht übertragbar sind, bzw. nur zum Teil in die Realität umgesetzt werden können und die Technologie an Interesse verliert. Nach der erneuten Sammlung, der *"Kurs-Korrektur"* [BITFORGE 2019], wird die präsenste Innovation realistischer beurteilt. Durch die objektive und realitätsnähere Betrachtungsweise entsteht ein neues und realistisches Bild der Möglichkeiten, als auch der Grenzen der Technologie. Zum Ende hin geht die ehemals neue Innovation in eine routinierte Technologie über, wobei diese an Anerkennung gewinnt und sich bewährt. Bei den Nutzern findet sie immer mehr Zuspruch und zu Beginn utopisch gedachte Konzepte sind längst einer realen Praxis gewichen. Durch die steigende Zuwendung zu der Technologie erfährt diese eine stetige Weiterentwicklung und die ursprünglich kleine Nutzergruppe kann zu einer großen Community wachsen. Bezogen auf die Phasen des Hype Cycles befindet sich diese Technologie dann an der Stelle des Plateaus der Produktivität und bestätigt die Marktreife. Ab diesem Zeitpunkt handelt es sich nicht mehr um eine Zukunftsvision, Hype oder Highlight, sondern um eine am Markt etablierte Technologie.

Momentan befindet sich Augmented Reality auf dem Pfad der Erleuchtung und ist auf sehr gutem Wege zu einer ausgereiften Technologie, da das Wachstum der Verwendung von Augmented Reality stetig steigt und mittlerweile ein weites Portfolio an möglichen Einsatzgebieten vorweist. Die jetzige Erfahrung und der technologische Fortschritt bringt Augmented Reality den ursprünglich angedachten Visionen und Ideen einen Schritt näher, sodass in naher Zukunft die letzte Phase, das Erreichen des Plateaus der Produktivität erzielt werden kann. Den finalen Schritt der endgültigen Marktreife zu erlangen ist ein faszinierender und wichtiger Grund für meine Motivation mich dieser Technologie und der dahinter stehenden Theorie zu widmen. Die Technologie der Augmented Reality weist, wie bereits erwähnt, ein immer größer werdendes Portfolio an Einsatzgebieten vor, unter anderem:

- Industrie
- Produktion und Lagerlogistik
- Wartung und Reparatur
- Spiele, bzw. Gaming
- Medizin
- Marketing und Werbung
- Navigation
- Unterhaltung und Fernsehen
- Schulung und Training
- Militär

Neben der Affinität der Technologie bringt AR Vorteile mit sich, z.B. Arbeitsprozesse und Herangehensweisen an Projekte zu modernisieren. Ebenso wird die Fehleranfälligkeit durch menschliche

Unachtsamkeit mit einer Kombination aus vielen Informationen und deren visueller Darstellung in Echtzeit reduziert. Hinzu kommt die Steigerung der Produktivität und Verbesserung des Wissenstransfers durch die dargebotenen Angaben die visualisiert werden.

Schon im Jahr 2016 sprach Apple CEO Tim Cook die Innovation *Augmented Reality* bei zahlreichen Events, in Keynotes und Interviews an und war zu diesem Zeitpunkt schon enorm begeistert davon. Er beteuerte: „...using the tech would become as normal as eating three meals a day.“ [LESWING 2016] Bei jeder sich ihm bietenden Möglichkeit ging der Apple CEO auf seine Überzeugung gegenüber Augmented Reality ein. Erst vor Kurzem bestätigte er, dass AR zahlreiche und innovative Einsatzgebiete erlangen und immer mehr an Wichtigkeit zunehmen würde, als er sagte: „pervade your life, it will play a big role...“ [EADICICCO 2020]

Ein sehr großer Anwendungsbereich mit viel Potential, das bei weitem noch nicht ausgeschöpft sei, ist der Industriezweig. Die Marktstudie zu Augmented Reality der IDG Research Services und PTC untermauern das immer weiter steigende Wachstum der Anwendungen in Unternehmen. „Fast 75% der deutschen Unternehmen setzen bereits *Virtual oder Augmented Reality* ein oder planen dies.“ [MUNDT 2020] Eine Prognose berechnet einen Umsatz in zweistelliger Milliarden-Höhe unter Einsatz und Anwendung von AR der bis 2021 erreicht werden soll. So wird um so deutlicher, dass es sich bei Augmented Reality um ein sehr zukunftsorientiertes Marktsegment mit viel Potential handelt. Die Technologie rückt immer weiter in den Vordergrund und immer mehr Unternehmen setzen sich damit auseinander.

Durch den Trend zur Vollautomatisierung von Produktions- und Industrieprozessen werden Ummengen an Daten erzeugt. Die moderne Industrie 4.0 steigert das Wachstum der Daten, indem Maschinen immer mehr digitalisiert und Protokolle parallel zur Laufzeit der Anlage erzeugt werden. Die entstehenden Daten werden über eine Cloud zur Verfügung gestellt. Dieser Vorgang des Datentransfers ist Teil einer globalen Infrastruktur, die es ermöglicht physische und virtuelle Gegenstände miteinander zu vernetzen, auch bekannt als Internet of Things, dt. Internet der Dinge (IoT).

Entstehende Daten sind z.B. Maschinendaten aus Maschinen-Protokollen, Prozessinformationen, Produktionsdaten oder Informationen der Endverbraucher. [KRAUSS 2019] Augmented Reality ermöglicht es, einen Großteil der Daten sinnvoll zu nutzen, d.h. dem Nutzer Auskünfte über Maschinen besonders hervorzuheben. Darüber hinaus können unter anderem durch AR veraltete Papierprozesse abgelöst oder digitale Lösungen in bestehende IT-Infrastrukturen eingebunden werden. [BOUVERET und HUMAN 2019] Dabei sind Rückschlüsse auf die allgemeine Arbeitsoptimierung zu ziehen. Notwendige Informationen sind ohne großen Aufwand, schnell und zentral an einem Ort durch AR zur Verfügung gestellt und abrufbar. Damit können Wartungen schneller durchgeführt, Defekte besser behoben, Leerläufe oder Fehler am Endprodukt vermieden werden. Mit solch einer Unterstützung kann z.B. auch die Orientierung in riesigen Produktionshallen aufrecht erhalten werden, um einen besseren Überblick über alle Maschinen zu erhalten. Auch wird die Effizienz von Prozessen erhöht. Ein Mechaniker kann alle notwendigen Informationen, die er benötigt, der Augmented Reality Applikation entnehmen, um z.B. eine Wartung einer Anlage schneller durchzuführen oder über Vorfälle an der Maschinerie in Echtzeit informiert zu sein.

1.2 cjt Systemsoftware AG

Die Arbeit wurde bei der Firma cjt Systemsoftware AG durchgeführt. Diese wurde 1999 von Christian J. Tauber und Ulrich Beck gegründet. Damals mit einem Team von 20 Personen, beschäftigt die cjt Systemsoftware AG heute mehr als 60 Mitarbeiter. Mit ihrem Sitz in Karlsruhe ist sie in einer der größten Technologiestädten Deutschlands angesiedelt.

Durch das stetige Wachstum der cjt Systemsoftware AG vergrößert sich auch deren Portfolio kontinuierlich. Dabei setzt das Consulting-Unternehmen hauptsächlich auf maßgeschneiderte Software- und Netzwerklösungen. Großkunden wie Siemens AG, Lufthansa Cargo, Forschungszentrum Karlsruhe (KIT) und Fraunhofer IOSB zeugen von der hohen Qualität der geleisteten Arbeit. Dabei agiert das Unternehmen nicht nur in Deutschland sondern auch international, darunter in Ländern wie China und den USA.

Einer der größten Auftraggeber der Firma ist ebenso das in Karlsruhe angesiedelte Unternehmen Siemens AG, dass eine beispielhafte Anwendung für das Assistenzsystem bietet und in Zukunft auch der Firma zur Nutzung unterbreitet werden könnte.

1.3 Aufgabenstellung

Im Rahmen dieser Ausarbeitung soll ein System konzipiert, entwickelt und umgesetzt werden, welches basierend auf Augmented Reality ein Informations- und Unterstützungssystem im industriellen Bereich grundlegend realisiert.

Die entstehende prototypische Applikation soll es ermöglichen, einen Überblick über eine Produktions- oder Industriehalle zu verschaffen, indem die Umgebung und alle in der Halle stehenden Maschinen über die Kamera eines Smart-Devices erkannt und als Overlay über das Live-Bild gelegt und angezeigt werden. Die erkannten Objekte können vom Nutzer eingetragen werden, um die Position des Gegenstandes festhalten zu können. Nach dem Positionieren des Objektes kann der Nutzer die benötigten Informationen eintragen, um so grundlegende Informationen über diesen Gegenstand griffbereit zu haben. Damit soll der Grundbaustein für ein Unterstützungssystem gelegt werden, das beliebig erweiterbar ist, um ein nützliches *Gadget* mit vielen *Features* in der Industrie zur Verfügung zu stellen.

Grundlegend ist das Projekt in drei Aufgabenbereiche unterteilt:

- Planung der Architektur
- Implementierung der Grundfunktionen der Applikation
- Grundlage der Objektinformation als Datenmodell

Um eine Applikation übersichtlich zu gestalten und für die Zukunft Erweiterungen möglichst einfach integrieren zu können, ist es zu Anfang die Aufgabe, eine solide und übersichtliche Grundstruktur, bzw. Software- Architektur zu erstellen. Dadurch werden Funktionen und Klassen einer klaren Struktur zugeordnet und eine einheitliche Linie vorgegeben.

Ein weiterer Aspekt, der bei der Erstellung der Architektur berücksichtigt werden soll, ist der Ansatz der modularen Softwarearchitektur. Damit können einzelne Funktionen unabhängig voneinander getestet und Abhängigkeiten oder Frameworks leichter ersetzt, hinzugefügt oder entfernt werden. Darüber hinaus begünstigt eine modulare Konzeption bessere Kontrollierbarkeit und Übersichtlichkeit in großen Softwareprojekten.

Die Hauptaufgabe ist die Realisierung der Augmented Reality-Funktion, sozusagen der Kern der Anwendung. Dabei wird die Applikation in zwei Phasen unterteilt, welche es gilt separat zu planen und implementieren.

Die erste Phase, genannt Scan-Phase, beschäftigt sich mit dem scannen der Umgebung, bzw. des Raumes. Die Aufgabe dabei besteht darin, mittels dem Simultaneous Localization And Mapping (SLAM) - Verfahren eine Karte der Umgebung zu erstellen und die räumliche Lage innerhalb dieser Karte zu schätzen, um auf Basis dieser erstellten Karte virtuell Objekte auf der Karte platzieren zu können. Mit den gewonnen Informationen der räumlichen Darstellung durch das SLAM - Verfahren kann der Nutzer virtuelle Objekte im virtuellen Raum an Ort und Stelle platzieren, als Referenz zu dem existierenden Objekt in der Realität, welches erkannt wurde. Bei der Erstellung eines Objekts soll der Nutzer die Möglichkeit haben, Informationen über das Objekt in das System einzupflegen, damit er diese immer abrufen kann.

In der zweiten Phase, genannt Visualisierungs-Phase, sollte der Nutzer die Möglichkeit haben, sich im Raum frei bewegen zu können. Mit der Lokalisierung des Nutzer-Geräts und den

bekannten Informationen der in Phase eins gesetzten Objekte, sollten dem Nutzer die virtuellen Objekte in seiner unmittelbaren Umgebung angezeigt werden. Mit dem Wissen, dass sich im Blickfeld der AR-Applikation ein Objekt befindet, können für dieses in der Datenbank weitere Informationen abgefragt und dem Nutzer zur Verfügung gestellt werden.

Ein weiterer wichtiger Punkt dieses Konzeptes wird die Modellierung eines geeigneten, grundlegenden und prototypischen Datenmodells sein. Dieses Datenmodell gibt vor, welche Informationen in Phase eins beim Erstellen eines Objektes vom Nutzer eingegeben und damit erfasst werden sollten, um die wichtigsten Daten zur Verfügung zu haben.

1.4 Aufbau der Arbeit

Nach den soeben genannten einleitenden Informationen widmet sich das Kapitel (2) den essentiellen und wichtigsten Grundlagen dieser Arbeit. Zu Anfang wird dem Leser der Terminus der Augmented Reality (2.1) offenbart, um allgemein Kontexte im Bezug zu dieser Arbeit zu begreifen, gefolgt von einer Einführung in die Thematik des Verfahrens SLAM-Simultaneous Localization And Mapping (2.2) der überbegrifflichen Materie der Robotik. Eine weitere notwendige Grundlage ist das Verständnis von Quaternionen (2.3) und die damit zusammenhängende Rotation und Translation von Objekten in einem dreidimensionalen Raum. Nach den erworbenen Grundkenntnissen der Basis-Thematiken, wird das Wissen über die Voraussetzungen und verwendeten Technologien (2.4) geschaffen. Darauf folgend wird im Allgemeinen auf Softwarearchitektur (2.5) und Modulare Software Architektur (2.5.1) eingegangen. Abschließend zu Kapitel (2) wird zu guter Letzt der Bereich der Datenmodellierung (2.6) thematisiert.

An Kapitel (2) anschließend wird in Kapitel (3) die Konzeption dargelegt. Anfänglich werden in diesem Abschnitt der Arbeit Gedanken, Überlegungen und vorläufige Konzeptionen der Arbeit aufgefasst und erläutert. Unter anderem welche Bedingungen die Arbeitsumgebung (3.1), in der die Applikation ihren Nutzen erweist, mit sich bringt. Erweiternd dazu wird darauf eingegangen, wie die beiden Phasen Scan-Phase (3.2) und Visualisierungs-Phase (3.3) konzipiert wurden.

Ebenso wird das Architekturkonzept (3.4), welches für das System vorgesehen war, genauesten dargelegt. Im Anschluss wird auf das ebenso tragende Softwarekonzept (3.5) eingegangen. Darauf folgend eine kurze Evaluierung wieso sich für das angewendete AR-Framework (3.6) entschieden wurde und abschließend zu Kapitel (3) die Intension des konzipierten Datenmodell's (3.7).

Kapitel (4) befasst sich mit der Umsetzung des Konzepts, dem Ablauf, den besonders erwähnenswerten Lösungen und den dabei aufgetretenen Problemen.

Nach der Umsetzung (4) gibt es eine Evaluierung (5), um das Projekt selbst zu beurteilen und einen eigenen Entschluss zu ziehen.

Die letzten zwei Kapitel, Fazit (6) und Ausblick (7), runden die Dokumentation ab und schließen die Arbeit. Die vorzuweisenden Ergebnisse werden analysiert und Verbesserungsvorschläge angemerkt.

Der Ausblick gibt Aufschluss darüber welche Erweiterungsmöglichkeiten es für diese Arbeit gibt und wie innovativ sich dieser Grundbaustein in Zukunft erweist.

1.5 Stand der Technik

Kapitel 2

Grundlagen

In diesem Kapitel werden die für diese Bachelorarbeit notwendigen Grundlagen geschaffen, um ein fundiertes Wissen und Verständnis über verwendete Technologien zu schaffen. Auf alle diese Informationen und Voraussetzungen wird im Folgenden eingegangen, um nachfolgende Konzeption und Umsetzung besser zu verstehen.

2.1 Augmented Reality

Eine der wichtigsten Grundlagen dieser Arbeit ist das Verständnis des Begriffs der Augmented Reality.

Augmented Reality, deutsch erweiterte Realität, ist eine durch den Computer gestützte Erweiterung der Realität, bzw. der menschlichen Wahrnehmung. Es ermöglicht dem Nutzer die reale Welt mit Überlagerung oder Zusammensetzung virtueller Objekte und visueller Informationen zu sehen. Mittels einer Art Overlay werden diese Objekte und Informationen über die reale Welt gelegt und dem Nutzer zur Verfügung gestellt. Allgemein soll damit dem Nutzer ein weit gefächerter Überblick verschafft werden und Hilfestellung leisten, aber den Nutzer in keinerlei Interaktion mit der Umgebung einschränken. Die Definition, welche sich in der Wissenschaft weitestgehend durchgesetzt und etabliert hat ist die Definition nach Azuma aus dem Jahre 1997.

„Augmented Reality (AR) is a variation of Virtual Environments (VE), or Virtual Reality as it is more commonly called. VE technologies completely immerse a user inside a synthetic environment. While immersed, the user cannot see the real world around him. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world. Therefore, AR supplements reality, rather than completely replacing it.“ [AZUMA 1997]

Ein Augmented Reality System verfügt nach [AZUMA 1997] über folgende drei charakteristische Merkmale:

1. Es kombiniert Realität und Virtualität.
2. Es ist interaktiv in Echtzeit.
3. Die virtuellen Inhalte sind im 3D registriert.

Das erste genannte Merkmal kombiniert die reale Welt mit dem oben genannten Overlay, der Überlagerung der Realität um künstliche virtuelle Objekte und visuelle Informationen. Dies

bedeutet, der Nutzer nimmt die reale Umgebung gleichzeitig mit den darin liegenden virtuellen Objekten als ein Ganzes wahr. Daraus resultiert die Interaktion von virtuellen Objekten und Informationen mit der realen Welt in Echtzeit, damit sie als Teil der Realität registriert werden können. Das dritte Merkmal umfasst die Darstellung von Objekten als scheinbar reales Objekt. Mit dem letzt genannten Merkmal wird das Ziel verfolgt die projizierten, bzw. nicht realen Teile täuschend echt in die Umgebung zu integrieren.

Eine etwas allgemein formuliertere Definition ist die nach [DÖRNER 2019], welche die drei charakteristischen Merkmale besonders aufgreift:

„Augmentierte Realität (AR) ist eine (unmittelbare und interaktive) um virtuelle Inhalte (für beliebige Sinne) angereicherte Wahrnehmung der realen Umgebung in Echtzeit, welche sich in ihrer Ausprägung und Anmutung soweit wie möglich an der Realität orientiert, sodass im Extremfall (so dies gewünscht ist) eine Unterscheidung zwischen realen und virtuellen (Sinnes-) Eindrücken nicht mehr möglich ist., [DÖRNER 2019]

Diese Definition nimmt sich als Grundlage die oben aufgeführte Definition nach [AZUMA 1997].

Der Author L. Frank Baum [GAEVERT 1856] verkündete die ersten Ideen und Gedanken einer Augmented Reality Anwendung in „*The Master Key*“ [BAUM 1996]. Eine erste tatsächliche Realisierung eines Augmented Reality Systems erfolgte erst über 60 Jahr später. Ivan Edward Sutherland [SUTHERLAND 1938] stellte sein Projekt 1968 an der University of Utah vor. Dabei handelte es sich um ein sogenanntes *Head-Mounted Display (HMD)*. Ziel dieser Entwicklung war weniger das Erweitern der Realität, sondern dreidimensionale Illusionen zu erzeugen die reale Objekte mit einer einfachen Grafik in Echtzeit überlagert. Trotz dessen gilt er als erste Person mit der Vision, einen Nutzer in realer Umgebung mit virtuellen Objekten interagieren zu lassen. Anfang der 90er Jahre prägten zwei Forscher, Thomas P. Caudell und David W. Mizell, den Begriff der Augmented Reality durch ein Pilotprojekt bei Boeing. Das Projekt diente dazu Informationen in das Gesichtsfeld über eine Brille einzusetzen, um Arbeitern das Verlegen von Kabeln im und um das Flugzeug zu erleichtern. Nach dieser bahnbrechenden Erfindung begann eine stetige Weiterentwicklung der Technologie. Im Jahre 1999 wurde von Hirokazu Kato und Mark Billinghurst *ARToolKit*, ein Computer-Vision-basiertes Tracking für AR, veröffentlicht und „löste eine große Welle an Forschungsarbeiten auf der ganzen Welt aus.“ [DÖRNER 2019]

We describe an augmented reality conferencing system which uses the overlay of virtual images on the real world. Remote collaborators are represented on Virtual Monitors which can be freely positioned about a user in space. Users can collaboratively view and interact with virtual objects using a shared virtual whiteboard. This is possible through precise virtual image registration using fast and accurate computer vision techniques and HMD calibration. We propose a method for tracking fiducial markers and a calibration method for optical see-through HMD based on the marker tracking. [KATO und BILLINGHURST 1999]

Dieser Ausschnitt war der grundlegende Baustein des Durchbruchs dieser Technologie und den vorangestellten Forschungen und eine fundierte Grundlage für alle Forschungen und Entwicklungen die darauf folgten.

Heutzutage dreht sich die Entwicklung um mobile AR, welche durch die anfängliche Revolution von *ARToolKit* und die darauf entstehenden Entwicklungen und Produktionen von großen

Firmen, wie z.B. Google, Microsoft, Apple und Facebook entstand. Die zuletzt große Bewegung in dem Bereich der AR waren die Vorstellungen großer Software-Plattformen für mobile AR-Applikationen. Durch *Apple's ARKit* und *Google's ARCore* kamen im Jahr 2017 zwei moderne und innovative Frameworks auf den Markt, die die Entwicklung von Augmented Reality-Applikationen stark beeinflussen. Die Frameworks wurden bei den ersten Produktionen für Entertainment-Anwendungen genutzt, um z.B. mobile Spiele zur Unterhaltung oder Funktionen bei Sport-Fernsehübertragungen zur Anzeige der Entfernung des Freistoßes zu realisieren.

Diese Arbeit jedoch widmet sich ausschließlich dem industriellen Aspekt und stellt andere Bereiche in den Hintergrund. Die in Kapitel 1.1 aufgeführte Markstudie bestätigt das enorme Potential hinter Augmented Reality und deren Einsetzbarkeit in der Industrie. Hauptsächlich in der Produktion, der Wartung oder der Reparatur von Maschinen kann Augmentierte Realität eingesetzt werden und zeigt einen positiv erzeugten Mehrwert. Dabei können bei Maschinen das Anzeigen von protokollierten Fehlern oder eine visuelle Hilfestellung bei Defekts, sowohl bei der Reparatur, als auch bei der Ersetzung einzelner Komponenten eine deutliche Reduzierung des zeitlichen Aufwands oder eine effektivere Arbeitsweise vorweisen.

Harvard Business Review legte einen Vergleich offen, indem ein Techniker ein Steuergerät einer Windkraftanlage mithilfe eines AR-Headsets verkabelt und in Betrieb nimmt. Alle benötigten Informationen wurden Schritt für Schritt über das Headset zur Verfügung gestellt. Dadurch gab es keinen Mehraufwand, z.B. das Nachschlagen in einer Dokumentation. Danach führte der Techniker den gleichen Prozess ohne die Hilfe der AR-Anwendung durch, lediglich mit Verwendung des vorliegenden Handbuchs, welches physisch beiseite lag. Dieser Test bestätigte eine Leistungsverbesserung des Arbeiters beim ersten Gebrauch um 34%. [ABRAHAM und ANNUNZIATA 2017] Diese Erkenntnis des Tests stützte den Gedanken der Leistungsverbesserung und der Reduzierung des zeitlichen Aufwands, somit lies dieses Resultat den Anlass zu dieses Ergebnis auf die Gesamtheit zu projizieren und für alle Anwendungsfälle allgemeingültig zu machen.

2.1.1 Virtual Reality, Augmented Reality und Mixed Reality

Während immer mehr Leute mit dem Begriff Virtual Reality etwas anfangen können, gibt es doch noch viele Unsicherheiten bei den dazukommenden Begriffen der Augmented und der Mixed Reality. Diese drei Begriffe lassen sich meist nicht immer voneinander unterscheiden, da es viele Überschneidungen aber auch gravierende Unterschiede gibt.

Folgender Abschnitt beleuchtet die Unterschiede und lässt die drei Formen der erweiterten Realität voneinander unterscheidbar machen.

Virtual Reality

Virtual Reality, dt. Virtuelle Realität (VR), ist eine in Echtzeit computergenerierte, interaktive und virtuelle Umgebung. Eine Darstellung und gleichzeitige Wahrnehmung der Wirklichkeit in all ihren Facetten und Eigenschaften. Das Ziel dieser Technologie ist, den Nutzer von der Außenwelt abzuschirmen und diese durch eine computergenerierte und detaillierte Welt zu ersetzen. [MÜHLROTH 2018] Auch bekannt als Immersion.

Die konventionelle Computergraphik ist für den Menschen spürbar nicht von belangen und weckt keine physischen Emotionen, wogegen VR diese etwas beeinflussen kann. Wie diese Unterschiede

spürbar sind, wird in folgendem erläutert.

Die Tabelle 2.1 fasst die Unterscheidungsmerkmale von Virtueller Realität zur konventionellen Computergraphik zusammen. [DÖRNER 2019]

3D- Computergraphik	Virtuelle Realität
Rein visuelle Präsentation	Multimodale Präsentation
Präsentation nicht notwendigerweise zeitkritisch	Echtzeitdarstellung
Exozentrische Perspektive	Egozentrische Perspektive
Statische Szene oder vorberechnete Animation	Echtzeitinteraktion und -simulation
2D-Interaktion (Maus, Tastatur)	3D-Interaktion (Körperbewegung, -gestik)
Nicht-immersive Präsentation	Immersive Präsentation

Tabelle 2.1: Merkmale der Computergraphik gegenüber der VR [DÖRNER 2019]

Virtuelle Realität ist immer in Verbindung mit Head-Mounted Displays zu betrachten, da ein Gerät benötigt wird, welches den Nutzer von der realen Welt abschottet und in die virtuelle Welt begleitet. Diese werden auf Hochtouren von großen Firmen, wie Microsoft, Sony, Facebook etc. entwickelt. Die erste entwickelte und auf dem Markt veröffentlichte Brille war die HoloLens von Microsoft, gefolgt von der Brille Namens Oculus Rift von Facebook usw.

Mit der stetigen Weiterentwicklung dieser Brillen und der Technologie wird versucht nach und nach mehr Sinne des Menschen manipulieren zu können, bzw. das Spiel- und Gefühlserlebnis bei Konsolen-Spielen immer realistischer zu gestalten. Allerdings sind die Möglichkeiten im Massenmarkt stark beschränkt auf die folgenden aufgelisteten Sinne:

- Sehen: Durch Head-Mounted Displays (Oculus Rift), die die reale Welt abschirmen und vom Nutzer nicht mehr wahrgenommen werden kann
- Hören: Durch Kopfhörer, somit werden Geräusche der Realität übertönt
- Fühlen: Durch Controller mit haptischem Feedback, um Ereignisse der virtuellen Welt physisch spürbar zu gestalten.

Die Entwicklung dieser Technologie wird uns in Zukunft weiter verfolgen. Vielleicht gibt es irgendwann die Möglichkeit weitere Sinne virtuell zu steuern. An den Universitäten von Singapur und Tokio zum Beispiel, gibt es Forscher-Teams die ein großes Budget zur Verfügung haben, um dieser Sinnestörung auf den Grund zu gehen. Sie fanden heraus, dass thermische und elektrische Stimulationen bestimmte Geschmackseindrücke vermitteln können. Auch Wissenschaftler in China und an der Oxford University haben herausgefunden, dass bestimmte Audiosignale mit einem süßen Geschmack assoziiert werden. [BREHM 2017] Forscher aus aller Welt gehen mit einer bestimmten Ernsthaftigkeit die Möglichkeiten der Sinneserweiterung in der virtuellen Realität an.

Augmented Reality

Augmented Reality (AR) setzt im Gegenzug zu Virtual Reality (VR) auf das tatsächliche Erweitern der Realität durch das Einblenden von Informationen, Vorgängen, Hilfestellungen oder die Wegbeschreibung bei Head-Up Displays in Personen-, Kraft- und Nutzfahrzeugen, während

Virtual Reality den Nutzer in eine völlig eigene Welt entlockt und von der Realität abkapselt. Bei AR soll dem Nutzer die zu bewältigenden Aufgaben und dessen Bestreben vereinfacht und gestützt werden ohne die Realität außer Acht zu lassen. Darüberhinaus bietet Augmented Reality deutlich mehr Ansatzmöglichkeiten diese Technologie umzusetzen und zu nutzen, z.B. wie erwähnt durch Head-Up Displays bei Autos und Flugzeugen, durch Smartphones und Tablets die durch die Kamera die Realität erweitern, Brillen mit eingeblendeten Projektionen, wie die Oculus Rift und anderweitige große Projektionen.

In Kapitel 2.1.2 wird auf die Varianten genauer eingegangen.

Mixed Reality

Dem Namen entsprechend ist Mixed Reality (MR) eine Mischung aus Augmented Reality und Virtual Reality, jedoch die Art und Weise ist dabei entscheidend. Es gibt MR -Brillen, die die reale Welt als Basis der Interaktion nehmen und Brillen die alleinig auf digitalen Bildern aufbauen. [SCHANZE 2018]

Mixed Reality bezüglich der realen Welt

Diese Art der erweiterten Realität basiert auf den gegebenen Grundlagen der AR. Ähnlich zu Augmented Reality werden Objekte der realen Welt hinzugefügt, indem sie an bestimmte Stellen projiziert werden. Mixed Reality baut darauf auf und verankert die digitalen Objekte mit dem realen dreidimensionalen Raum, sodass eine Realitätsnahe Interaktion stattfinden kann. Virtuelle und echte Welt verschmelzen dadurch endgültig zu einer einzigen Welt.

Mixed Reality bezüglich der virtuellen Welt

Hinsichtlich der VR gibt es zu Anfang keine grundlegenden Änderungen gegenüber der MR. Hierbei kann der Nutzer durch MR ebenso die Außenwelt ausblenden und sich lediglich auf die virtuelle Wahrnehmung fixieren. Erst in der Benutzung werden die Unterschiede zu VR sichtbar und dieser ist gravierend. Der Nutzer kann sich frei bewegen, d.h. die Bewegungsfreiheit wird nur durch die reale Umgebung eingeschränkt, während Virtual Reality sich auf einen sensorgestützten Raum begrenzt. [MÜHLROTH 2018]

Bei MR wird jeder Schritt und jede Bewegung in die computergenerierte Welt übertragen und schafft so deutlich mehr Bewegungsfreiheiten.

Ein großer Investor dieser Technologie ist Microsoft mit Windows Mixed Reality (WMR), wobei bereits schon Erfolge erzielt werden konnten. Eine Standhafte Plattform die nur darauf wartet ausgebaut und intensiver genutzt zu werden. Für die Plattform vorgesehene Brillen gibt es schon viele Produzenten, wie z.B. die *HMD Odyssey* von Samsung oder der *Explorer* von Lenovo.

2.1.2 Varianten der Augmented Reality

Augmented Reality Anwendungen funktionieren alle nach dem gleichen Prinzip und verfolgen das gleiche Ziel, die Realität durch digitale Informationen oder Objekte zu erweitern. Die einzige deutliche Abweichung ist das jeweilige Endgerät und die technische Umsetzung in Zusammenhang mit der Hardware. Auf zwei dieser Varianten, mobiles und Smart-Brillen- oder auch Headset-AR genannt, wird in Folgendem eingegangen.

Mobiles AR

Smartphones sind in unserer Gesellschaft nicht mehr wegzudenken und haben sich fest etabliert. Durch die vielseitige und alltägliche Nutzung von Tablets und Smartphones eröffnete diese Sparte eine gute Möglichkeit Augmented Reality in den Alltag zu integrieren. Somit beleibt der ständige Gebrauch dieser Technologie nicht aus und bereichert die Art und Weise Spiele und Anwendungen zu entwickeln und zeigt den Fortschritt der technologischen Aspekte sowohl software-, als auch hardware-technisch. Durch die im Smartphone integrierte Kamera werden Live-Aufnahmen analysiert und dienen als Ausgangssituation für die AR-Anwendung. Durch die gegebene Möglichkeit der vorhandenen Kamera, kann ein Echtzeithintergrund erzeugt und zusätzlich mit Informationen per Overlay dargestellt werden. Je nach Anwendung kann der Nutzer auf verschiedene Weise mit den eingeblendeten und virtuellen Objekten interagieren, z.B. durch das Bewegen des Geräts, um das Objekt aus verschiedenen Blickwinkeln anschauen zu können, oder der direkten Interaktion mit dem Objekt durch Drehen, Skalieren oder Verschieben am Bildschirm. Ein Vorreiter der mobilen AR ist das 2016 auf dem Markt erschienenen *Pokémon Go* von Niantic, das den Ansatz der AR prägt. [FISCHBACH 2016]

Die weit verbreitete social Media Applikation *Snapchat* baut seit geraumer Zeit ebenso auf Augmented Reality, um Bilder lebhafter zu gestalten, wie der folgenden Abbildung 2.1 zu entnehmen ist.

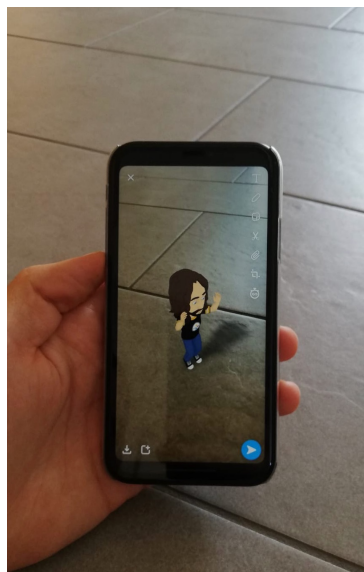


Abbildung 2.1: Mobile-AR in Snapchat

Smart-Brillen AR

2.1.3 Positionsbestimmung

Um ein digitales Objekt als Overlay dem Kamera-Live-Bild hinzuzufügen, werden genauestens definierte Positionen benötigt. Diese Positionen können durch unterschiedliche Ansätze ermittelt

werden. Je nach Anwendungsfall, z.B. als Navigation, Routenplaner oder Google Maps „Live View“ [BERGER 2019], reicht eine etwas ungenauere Positionsbestimmung per GPS, da in Relation zur realen Welt eine Abweichung um Zentimeter oder wenige Meter nicht von belangen ist. Bei Positionsbestimmungen auf kleinstem Raum ist eine genaue Lokation wichtig und basiert auf einem deutlich präziseren Ansatz.

Welche oben genannten Ansätze es gibt und welche Unterschiede zu beachten sind, wird in Folgendem näher darauf eingegangen.

Marker-basierte Positionsbestimmung

Speziell bei der Marker-basierten Positionsbestimmung gibt es verschiedene Möglichkeiten den Marker zu gestalten. Es können Binär- oder QR-Codes als Markierung verwendet werden, ein Beispiel eines solchen Codes ist der Abbildung 2.2 zu entnehmen. Diese Codes sind meistens quadratisch und haben ein eindeutiges Zeichen in der Mitte. Um die Rechenzeit gering zu halten gibt es einfache Muster, wie die Abbildung 2.2 zeigt. Neben der einfachen Markierungserkennung

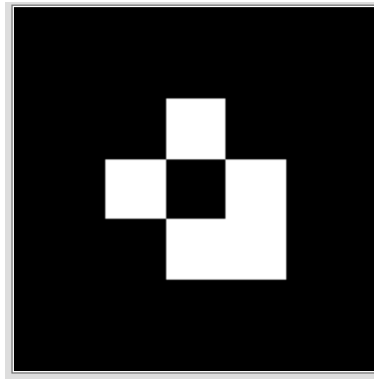


Abbildung 2.2: Marker-basierte Augmented Reality Positionsbestimmung

gibt es eine weiterentwickelte Möglichkeit der Bild- sowie der Objekterkennung. Diese sind Ansätze die grundlegend auf dem Ausgangspunkt des Binär-Code-Verfahrens aufbauen. Eine detailliertere Erläuterung der soeben genannten Erkennungsmöglichkeiten findet im Rahmen dieser Arbeit nicht statt. Allerdings wird allgemein kurz auf die Funktionsweise einer Marker-basierten Positionsbestimmung eingegangen.

Durch ein Kamerabild wird nach einem vordefinierten Marker, bzw. nach einer festgelegten Markierung gesucht. Ist diese mit der Kamera erfasst, wird die Markierung durch Bildverarbeitungsalgorithmen und bestimmter Filterung eindeutig identifiziert. Mit den gewonnen Informationen und der Übereinstimmung des angegebenen Codes wird die Position, sowie die Orientierung des Markers berechnet. Mit den Angaben der Lage und Orientierung wird auf der Markierung das anzuzeigende digitale Objekt generiert und als Overlay über dem Kamerabild angezeigt. Die Markierung dient sozusagen als Grundlage um den digitalen Gegenstand überhaupt anzeigen zu können.

Da der Marker immer im Blickfeld der Kamera sein muss, um das virtuelle Objekt anzuzeigen, bringt diese Art der Positionsbestimmung eine enorme Einschränkung mit sich, die in diesem Bezug unumgänglich ist.

Ein weiterer Ansatz der Positionsbestimmung ist als Überbegriff das Gegenstück zur oben

aufgeführten Lokalisierung von Markern, die sogenannte Marker-unabhängige Positionsbestimmung, welche ebenso verschiedene Ausführungen vorweist.

GPS-basierte Positionsbestimmung

Die Methode des GPS-basierten Positionsbestimmungsverfahrens verwendet hauptsächlich die Koordinaten der realen Welt. In Zusammenarbeit mit zusätzlichen im Anwendungsgerät verbauten internen Sensoren, bspw. Positions-, Geschwindigkeits- und Beschleunigungssensoren und Teile des Inertial Navigation System (INS), z.B. dem Gyroskop, kann diese Art der Positionsbestimmung optimal Anwendung finden. Allerdings werden dabei deutlich mehr Komponenten benötigt und ist deutlich komplexer umzusetzen, als ein markerbasiertes System.

Bei einem Anwendungsfall von GPS-basierter Positionsbestimmung geht es meist um Routenplaner, Navigation oder Szenarien die sich auf offenen Flächen abspielen, da im Gegensatz zu Marker-basierten Anwendungen das Größenverhältnis deutlich geräumiger ist. Der Benutzer ist nicht auf einen bestimmten Bezirk beschränkt und ist nicht auf die millimetergenaue Darstellung angewiesen.

Eine Alternative zu GPS ist die Anwendung des SLAM-Verfahrens. Dabei wird eine virtuelle Karte, bzw. ein geometrisches Modell der Umgebung erstellt. Wichtige Grundlagen zum Erzeugen eines solchen Modells sind eigenständig gefundene Landmarken die gleichzeitig lokalisiert werden. Darauf folgt ein Vergleich der Pose, der Position des Geräts und der geschätzten Karteninformationen aus dem Scan der Umgebung. Damit sind im erweiterten Sinne Marker geschaffen, die Anhaltspunkte für Augmented Reality-Interaktionen schaffen.

In Kapitel 2.2 wird die Thematik des SLAM-Verfahrens genauestens erläutert.

2.1.4 Augmented Reality in der Industrie

Das weit gefächerte Portfolio der Augmented Reality umfasst viele Anwendungsbereiche und Fachgebiete. Selbst dem übergeordneten Bereich der Industrie gibt es viele verschiedene Einsatzgebiete. Aus den vielen Möglichkeiten der Anwendung haben sich über die Jahre der Entwicklung der Technologie einige Gebiete in der Industrie herauskristallisiert, die besonders hohen Nutzen stiften. [SCHAT 2017] In den Bereichen Instandhaltung und Wartung, Betrieb und Training ist Augmented Reality auf bestem Wege fester Bestandteil des Alltags zu werden. Bei dieser Betrachtung ist es sinnvoll sich auf die damit einhergehenden Lösungen zu fokussieren, die Reduktion der Kosten und dem Zeitaufwand, sowie die Verbesserung der Sicherheit. [MUNDT 2020]

Bei einer Wartung oder Reparatur einer Maschine sind notwendige Informationen direkt greifbar und werden Schritt für Schritt angezeigt, sodass in bestimmten Situationen selbst ein Leihnehmer die Anweisungen befolgen könnte. So werden zusätzliche Recherchearbeiten oder Unklarheiten über Vorgänge aus dem Weg geräumt. Ein Mitarbeiter kann so mit einem Tablet oder einer Smart-Glas Anweisungen visuell auf die realen Maschinen projizieren und Arbeitsschritte im Sichtfeld anzeigen lassen. Ebenso geht dieser Vorgang auch bei der Produktion von Bauteilen o.ä., indem eine AR-Anwendung Anweisungen und Prozessschritte, z.B. auf das Werkstück oder Produkt, projiziert.

Die Inbetriebnahme oder Bedienung einer komplexen Anlage oder Maschine ist nach herkömmlichen Standard enorm zeitintensiv und dadurch können zusätzlich viele Fragen aufkommen, die

meist den Prozess noch länger gestalten als vorgesehen. AR-Anwendungen können Bedienungsanleitung oder -hilfen digitalisiert, indem Informationen, Inhalte oder Bedienelemente durch Augmented Reality auf der Anlage platziert werden.

In Fortbildungen, Schulungen oder Einarbeitungen in neue Geräte kann AR durchaus von großem Vorteil sein. Mit wenig Aufwand, einem effektiveren Training können Schulungen interaktiver und vor allem sicherer abgehalten werden. Durch die Visualisierung der Trainings- und Schulungsinhalten verbessert sich der Lernprozess und damit wird auch die Nutzung der Maschinen verständlicher. [SCHART 2017]

2.2 SLAM - Simultaneous Localization And Mapping

Das Simultaneous Localization and Mapping oder kurz SLAM Problem behandelt das gleichzeitige Schätzen der Position und Ausrichtung einer mobilen Plattform im Raum anhand der sich an Bord befindlichen Sensoren sowie den Aufbau eines Modells der Umgebung. Dieses Problem ist von großer praktischer Relevanz und ist Kernbestandteil der meisten mobilen Sensor- systeme. [FREEDEN und RUMMEL 2016]

Als Simultaneous Localization and Mapping bezeichnet man die Aufgabe, die Trajektorie samt Orientierungsinformation einer sich bewegenden Plattform, z.B. ein Smartphone, Tablet oder jegliche Art von Roboter, aus Beobachtungen zu schätzen und gleichzeitig aus den gewonnenen Informationen eine Karte der Umgebung zu erstellen. Diese Aufgabe ist für den weiteren Prozess von hoher Bedeutung. Zum einen sollen die generierten Karten sehr präzise sein, um einen hohen Wert für den Benutzer oder für spezielle Anwendungen, die auf der Karte aufbauen, darzustellen. Zum anderen benötigen autonome Roboter, beispielsweise Saug- oder Mähroboter, ein solch erzeugtes geometrisches Modell der Umgebung, um zielgerichtet selbstständig navigieren zu können. [FREEDEN und RUMMEL 2016]

2.2.1 Localization

2.2.2 Mapping

2.3 Quaternionen

2.3.1 Transformation

2.3.2 Rotation

2.4 Technologien

Dieser Abschnitt befasst sich mit den wichtigsten Programmier-Technologien, die in diesem Projekt eingesetzt werden.

Der Augenmerk liegt auf dem Framework *Google ARCore*, da dieses Hauptbestandteil des Projekts ist.

2.4.1 Google ARCore

ARCore ist die von Google veröffentlichte Plattform zum Erstellen von Augmented Reality Erlebnissen, primär entwickelt für Android Phones. Das Software Development Kit (SDK) ist

in den Programmiersprachen Java und Kotlin geschrieben. Mit bestimmten APIs ermöglicht *ARCore* die Erfassung der Umgebung durch ein Tablet oder Smartphone, um so die Welt zu verstehen und mit Informationen zu interagieren. Mitten des Jahres 2017 wurde die *Google ARCore API* zum ersten Mal den Entwicklern und der AR-Community präsentiert. Im ersten Quartal 2018 wurde diese dann veröffentlicht und löste das Vorreiterprojekt „*Project Tango*“ ab.

Project Tango war ebenso eine Google-Plattform, mit dem Smartphones und Tablets ein „*Raumgefühl*“ erhalten. [ELGAN 2016] Die Plattform kombinierte die Eingabe verschiedener Sensoren, z.B. radarähnliche Infrarotstrahler, einer Infrarotkamera und hochpräzisen Beschleunigungsmessern, Barometern und Gyroskopen, um schnell nutzbare Informationen der Umgebung zu generieren. Mit der Überarbeitung des Konzepts wurde das neue unabhängige Projekt *ARCore* geschaffen und somit die Anzahl der Hardwarekomponenten reduziert, bspw. wird keine spezielle Infrarotkamera mehr benötigt. Die *ARCore API* verwendet bei Anwendungen ausschließlich das Kamerabild des Smartphones und dessen Sensoren.

Zum Integrieren von virtuellen Objekten in die reale Welt verwendet *ARCore* drei Schlüsselfunktionen unter Benutzung von Java und OpenGL, Unity und Unreal:

- Motion tracking, dt. Bewegungsverfolgung, ermöglicht das Verstehen und Verfolgen der Position des Geräts relativ zur Realität.
- Environmental understanding, dt. Verständnis der Umgebung, erkennt die Position und Größe aller Arten von Oberflächen: horizontale, vertikale und abgewinkelte Oberflächen, wie Wände, Böden, Tische oder Stühle.
- Light estimation, dt. Lichtschätzung, schätzt die durch das Smartphone gegebenen Lichtverhältnisse der Umgebung. [LANHAM 2018]

Motion tracking

Für die Funktion der Bewegungsverfolgung verwendet *ARCore* das Verfahren von Simultaneous Localization And Mapping (SLAM) 2.2, um so zu verstehen wo sich das Smartphone in Relation zur realen Welt befindet. Zusätzlich erkennt *ARCore* visuell unterschiedliche Merkmale im aktuell aufgenommenen Kamerabild, sogenannte Merkmalspunkte „*feature points*“. Anhand dieser Punkte wird die Änderung des Standortes berechnet. Durch Trägheitsmessungen der Inertial Measurement Unit (IMU), dt. Inertiale Messeinheit des Geräts, kombiniert mit den visuellen Informationen des Kamerabildes wird die Position und Ausrichtung der Kamera relativ zur Realität abgeschätzt.

Die Inertial Measurement Unit in Smart-Devices besitzt drei Beschleunigungssensoren, z.B. piezoelektrische Beschleunigungssensoren oder Micro-Electro-Mechanical Systems (MEMS), für die drei Raumachsen:

- Die Abszissenachse (X-Achse), die horizontale (waagerechte) Koordinatenachse,
- Die Ordinatenachse (Y-Achse), die darauf vertikale (senkrechte) Koordinatenachse und
- Die Applikatenachse (Z-Achse), die auf beiden anderen Achsen senkrechte Achse [DIN-461 1973]

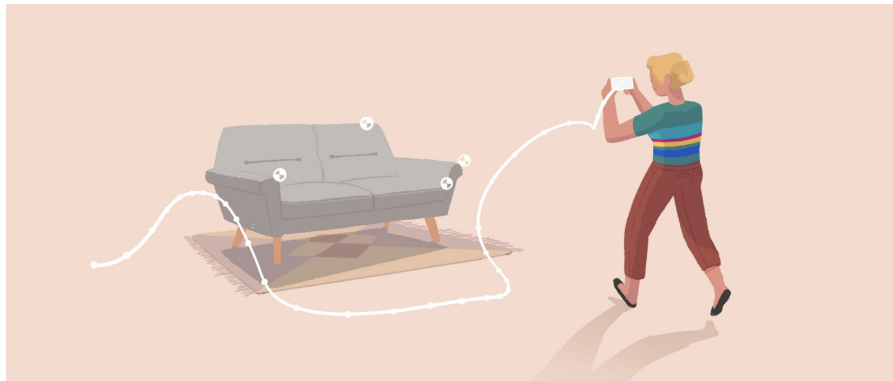


Abbildung 2.3: Skizze zur Bewegungsverfolgung des Smartphones [ARCore 2020]

Ebenso besitzt die IMU Drehratensensoren, z.B. ein Gyroskop zur Messung der Rotationsgeschwindigkeit, welche die Drehraten um diese Raumachsen messen. Durch die kontinuierliche Auslesung der Sensordaten, wird die Position des Geräts berechnet und in Form von drei Koordinaten ausgegeben. Ausgehend von dem Referenzkoordinatensystem wird die Position neu bestimmt. Die Abbildung 2.4 veranschaulicht solch ein Beispiel.

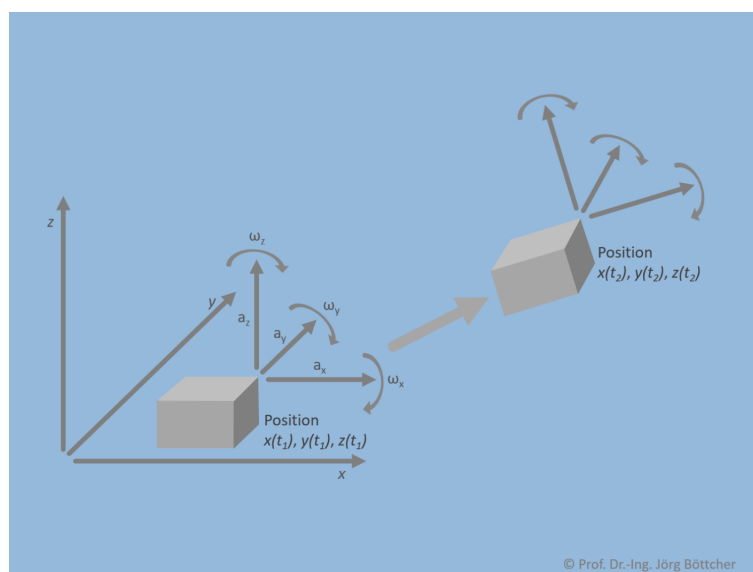


Abbildung 2.4: Grundprinzip einer IMU [BÖTTCHER 2020]

Environmental understanding

Die zweite Schlüsselfunktion des Frameworks ist das Verständnis der Umgebung. Dieses wird erlangt, indem „*feature points*“, Ebenen und Flächen beim scannen des Umfelds erkannt werden. Beim Entstehen mehrerer solcher *feature points* erstellt ARCore ein Cluster aller naheliegenden Punkte. Dieses erzeugte Cluster repräsentiert eine horizontale oder vertikale Fläche. Mit diesen gewonnenen Informationen ist das Platzieren von virtuelle Objekte auf den erfassten Flächen möglich.

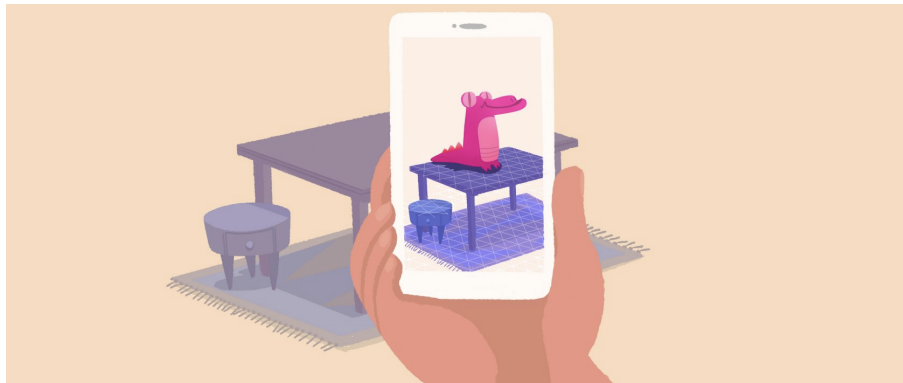


Abbildung 2.5: Skizze zum Umgebungsverständnis des Smartphones [ARCORE 2020]

Light estimation

Mit der Funktion der Lichtschätzung kann ARCore Lichtverhältnisse und Informationen über die Beleuchtung der Umgebung erkennen. Dadurch kann das SDK durchschnittliche Intensität und eine angepasste Farbkorrektur eines bestimmten Kamerabildes liefern, um so die Aufarbeitung des Bildes der Umgebung anzupassen. Diese Funktion verstärkt die Anpassung des virtuellen Objektes an die Realität und lässt dieses noch glaubwürdiger erscheinen.

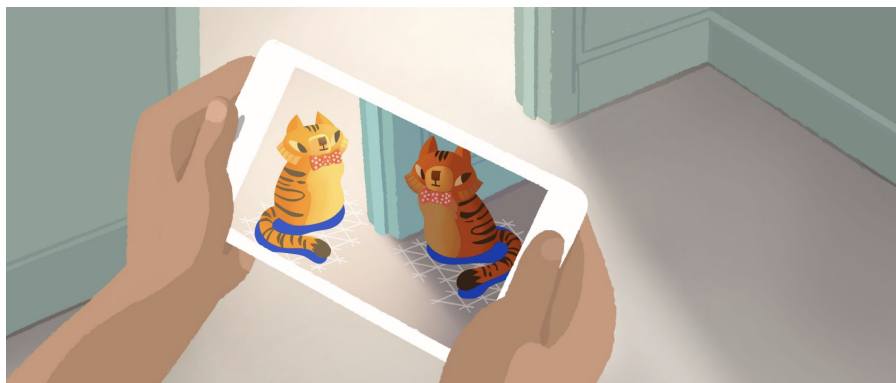


Abbildung 2.6: Skizze zu den Lichtverhältnissen des Smartphones [ARCORE 2020]

2.4.2 Android Jetpack

Android Jetpack ist eine Ansammlung an Bibliotheken, die es Entwicklern ermöglicht *Best Practices* zu befolgen, Boilerplate¹-Code zu reduzieren und Code zu schreiben. Die durch Android Jetpack zur Verfügung gestellten Bibliotheken fördern die konsistente Funktionsweise der Android-Versionen und -Geräten. Darüber hinaus wird mit Android Jetpack ein Überblick gewährleistet, um Best Practices und empfohlene Architekturen zu berücksichtigen.

Außerdem wurde Android Jetpack für moderne Design-Praktiken entwickelt, darunter fällt

¹Sind Codeabschnitte, die an mehreren Stellen mit wenigen oder gar keinen Änderungen aufgeführt sein müssen. [DEVELOPER 2019]

„separation of concerns“², *testability*, *loose coupling*, *Observer Pattern*³, *Inversion of Control*⁴ und *productivity features*, z.B. Plugin-Integrationen. Dies sind Techniken, um eine robuste und qualitativ hochwertige App zu erstellen.

Die Bibliothekensammlung ist die Grundlage für die verwendete Architektur, *Android Architecture Components* in Abschnitt 2.5.3.

Die Abbildung 2.7 verschafft einen Überblick über die in Jetpack enthaltenen Bibliotheken.



Abbildung 2.7: Komponenten von Android Jetpack [SELLS, POIESZ und NG 2018]

Eine nähere Erläuterung aller Bibliotheken findet im Rahmen dieser Ausarbeitung nicht statt, wobei die zur Verwendung geplanten „Architecture Components“ beschrieben werden.

LiveData, Room und ViewModel sind Bestandteil der *Android Architecture Components*-Konstellation und bilden die Basis für die Architektur, welche an das *MVVM-Pattern* angelehnt ist.

LiveData

LiveData gehört zu den *Observer-Pattern* und ist eine Datenhalterklasse die eine Beobachtungsfunktion, bzw. -charakter besitzt. Der Unterschied zu einem regulären Observable ist, dass die LiveData-Komponente lebenszyklusunabhängig ist. Dies bedeutet, der Lebenszyklus anderer Komponenten z.B. *Activities*, *Fragments* oder *Services*, wird berücksichtigt. Mit dieser Kenntnis wird sichergestellt, dass LiveData nur App-Komponenten beobachtet, welche sich in einem aktiven Lebenszyklus befinden.

²(SoC), Entwurfsprinzip, um verschiedene Verantwortlichkeiten auf verschiedene Abschnitte zu trennen. [NATESAN 2019]

³Beobachter-Muster, Verhaltensmuster der GoF-Muster

⁴(IoC), Umsetzungsparadigma, das ein spezifisches Objekt über ein mehrfach verwendetes Objekt aufruft. [NIEMANN 2020]

Vorteile die die *LiveData*-Komponente mit sich bringt sind unter anderem die Sicherstellung der Übereinstimmung der Benutzeroberfläche und deren Datenstatus, vorbeugend gegenüber Speicherverlust, immer auf dem aktuellsten Stand der gespeicherten Informationen und der Wegfall der manuellen Lebenszykluskoordination. [DEVELOPERS 2020]

Room

Room ist eine Persistenzbibliothek und bietet ein Abstraktionsschicht über der eigentlichen *SQLite*-Datenbank, um einen robusten Datenbankzugriff zu managen und das volle Potential von *SQLite* gleichzeitig zu nutzen. [DEVELOPERS 2017a]

Die robuste SQL-Objektzuordnungsbibliothek *Room* kann einen Cache mit den Daten der Applikation auf einem Gerät erstellen. Dieser Cache dient als Wahrheitsquelle der Applikation und ermöglicht den Benutzern eine konsistente Kopie der Informationen der App anzuzeigen und immer zugänglich zu machen. *Room* kann in drei Kategorien, bzw. in drei Hauptkomponenten unterteilt werden:

- Room database: Dient als Zugriffspunkt für die zugrundeliegende *SQLite*-Datenbank
- DAO (Data Access Object): Beinhaltet Methoden um Datenbankzugriffe zu gewährleisten.
- Entity: Repräsentiert ein Objekttable der Datenbank.

Das Diagramm 2.8 veranschaulicht die Interaktion, bzw. Funktionsweisen der Hauptkomponenten.

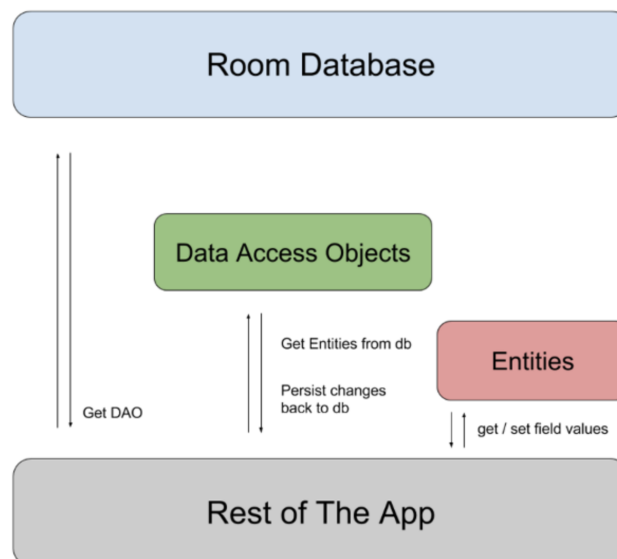


Abbildung 2.8: Room Architektur Diagramm [DEVELOPERS 2017b]

Allgemein ermöglicht *Room* einen vereinfachten Zugriff auf die Datenbank, hat einen hohen Grad an „*testability*“, zur Kompilierungszeit wird eine SQL-Abfrageüberprüfung durchgeführt und interagiert einwandfrei mit anderen Android Architecture Components, z.B. *LiveData* und *ViewModel*.

ViewModel

Grundsätzlich dient das *ViewModel* als lebenszyklusbewusste Speicherung und Verwaltung von Benutzeroberflächenbezogenen Daten. Durch die *ViewModel*-Klasse können Informationen Konfigurationsänderungen, z.B. Bildschirmdrehungen, die als Änderung zählen, überstehen. Eine Konfiguration kann sog. Activities verwerfen und neu laden, so können nicht persistierte Daten verloren gehen.

Ein *ViewModel* fungiert ebenso als Kommunikationsschnittstelle zwischen den darüber- und darunterliegenden Komponenten der Architektur (siehe Abschnitt 2.5.3). Des Weiteren kann die *ViewModel*-Komponente Informationen zwischen Benutzeroberflächen und verschiedenen Fragmenten, engl. „*fragments*“⁵, teilen.

Die Abbildung 2.9 zeigt eine sich ändernde Activity, die zu Beginn und nach Neuerstellung oder Aktualisierung Zugriff auf die unveränderten Daten hat.

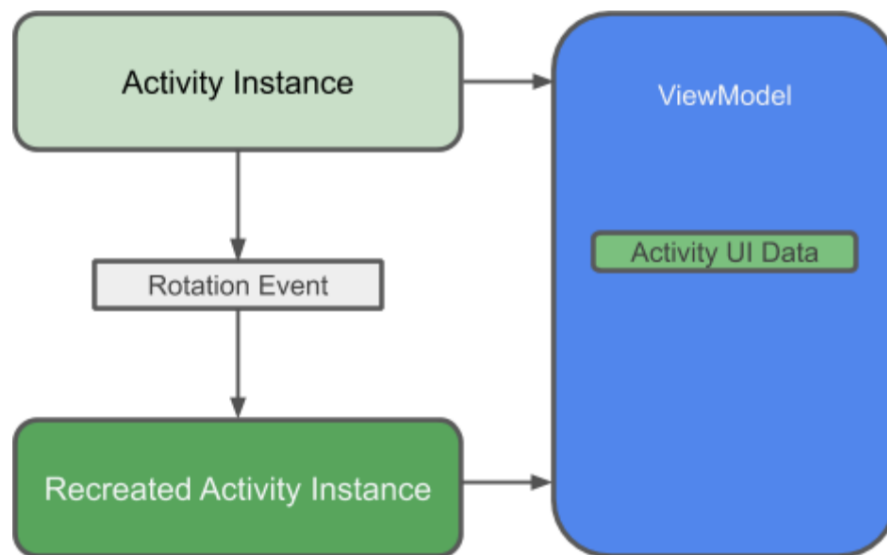


Abbildung 2.9: ViewModel Struktur Diagramm [CODELABS 2020]

2.4.3 Sceneform SDK

Das Sceneform SDK ist ein Open-Source Projekt der Google LLC., welches für das Rendern von 3D-Szenen und -Animationen zuständig ist. Basierend auf der physikalischen Echtzeit-Rendering-Engine, *physically based renderer* für Android, iOS, macOS, Linux und Windows von Google Filament, wurde Sceneform entwickelt.

Bei dem Ansatz des physikalisch basierten Renderns in der Computergrafik wird versucht, alle Modalitäten der realen Welt genauer zu modellieren. Mit dieser Methode werden virtuelle 3D-Objekte in der Augmented Reality noch realer dargestellt, um einen Echtheitseffekt zu

⁵Teil einer Graphical User Interface, dt. Benutzeroberfläche (GUI) in einer Android Activity

erzielen und simulieren. Ein Beispiel ist das simulieren des Lichtflusses, um einen Schatten des Objekts zu erzeugen, damit die Wirksamkeit immer mehr der Realität entspricht.

Dreidimensionale Modelle werden als *.obj*-Dateien in das Projekt importiert. Diese Dateien beinhalten aufgelistete Punkte, Vektoren und Linien, welche das Modell repräsentieren. Das *Sceneform SDK* verwendet die *.obj*-Datei und konvertiert, bzw. rendert diese in eine, *Sceneform binary assets*, *.sfb*-Datei, um daraus das dreidimensionale Objekt zu generieren, welches anschließend in AR-Anwendungen basierend auf Google ARCore verwendet werden kann. Für die finale Rendering und Anzeige ist die *ModelRenderable*-Klasse des SDKs zuständig.

2.4.4 SQLite

SQLite ist eine Open-Source In-Process-Bibliothek, die ein in sich geschlossenes, serverloses, transaktionsfreies SQL-Datenbankmodul ohne Konfiguration implementiert. SQLite ist eine eingebettete SQL-Datenbank-Engine die im Gegensatz zu anderen SQL-Datenbanken über keinen separaten Serverprozess verfügt, d.h. Transaktionen, Lese- und Schreibzugriffe werden direkt auf eine normale Festplattendatei getätigt. [HIPP 2018]

Darüber hinaus ist SQLite Plattformübergreifend und kann beliebig Datenbank- und Objekttabellen, Indizes und Ansichten erstellen und verwalten.

2.5 Softwarearchitektur

Komponenten in Form von Klassen, Objekten oder Bibliotheken und deren Verbindungen zwischen einzelnen Komponenten beschreibt die Architektur eines Softwaresystems. Vielmehr geht es bei der Software-Architektur darum, Anforderungen und deren Zusammenhänge untereinander von dem zu konstruierenden System zu beschreiben und nicht einen detaillierten Entwurf vorzulegen. Jedoch hat die Architektur einen enormen Einfluss auf die qualitativen und nicht-funktionalen Eigenschaften des daraus resultierenden Systems.

Die Terminologie nach dem *IEEE-Standard 1471-2000* zur Software Architekturbeschreibung, deren Aufgaben und Zweck [JECKLE u. a. 2005] sind wie folgt definiert:

Die grundlegende Organisation eines Systems, dargestellt durch dessen Komponenten, deren Beziehungen zueinander und zur Umgebung sowie den Prinzipien, die den Entwurf und die Evolution des Systems bestimmen. [HASSELBRING 2006]

Softwarearchitektur bietet viele Möglichkeiten ein System zu entwerfen und Anforderungen und Eigenschaften umzusetzen, daher gibt es auch hier viele Ansätze, Lösungen und Abwandlungen, um den Standards und Richtlinien gerecht zu werden. Beispiele dafür sind unter anderem die Modulare Software Architektur (2.5.1) und das Architektur-Entwurfsmuster MVVM (2.5.2).

Das Buch *Design Patterns - Elements of Reusable Object-Oriented Software* von E. Gamma, R. Helm, R. Johnson und J. Vlissides befasst sich mit den verschiedensten Möglichkeiten und Ausprägungen der Softwarearchitektur.

Im Rahmen dieser Ausarbeitung findet keine Aufzählung und Beschreibung verschiedener Arten der Architekturmuster und -stile statt, lediglich die für das Projekt verwendeten Muster werden in folgenden Kapiteln aufgegriffen.

Die Abbildung 2.10 veranschaulicht eine vereinfachte Struktur eines Architekturmusters und deren Komponenten und Zusammenhänge zueinander. Die Zeichnung dient zur Veranschaulichung, um darzulegen wie ein Architekturdiagramm aussehen kann, bzw. wie die allgemeine Struktur repräsentiert wird.

Im Falle dieser Abbildung wurde das Strukturmuster Model View Controller (MVC) ausgewählt, welches die zu sehenden Komponenten in drei in sich geschlossene und unabhängige Fragmente unterteilt die miteinander interagieren.

Neben der Strukturierung von Systemen und Applikationen kann die Modellierung einer Softwarearchitektur auch dabei helfen eine Architektur genauer zu beschreiben und zu dokumentieren. Ebenso können Diagramme auch das Management sowie die Planung beeinflussen und verstärken. Die Modellierung der Softwarearchitektur stellt somit keinen Selbstzweck dar, sondern bietet einen Mehrwert indem es zur Verständigung, Dokumentation und Kommunikation zwischen Entwicklern und Kunden zusätzlich beiträgt.

Durch die Modellierung der Architektur kann frühzeitig eine sinnvolle Evaluierung und Bewertung des Entwurfs durchgeführt werden. Mit diesen entstehenden Bewertungen können folgende Schritte besser geplant und umgesetzt werden.

Eine weitere Variante, bzw. Ausprägung der Software-Architektur ist die Modulare Software Architektur (siehe Abschnitt 2.5.1). Viele Softwarearchitekturen lehnen sich an das Prinzip der Modularen Architektur an und nehmen diese als Grundlage.

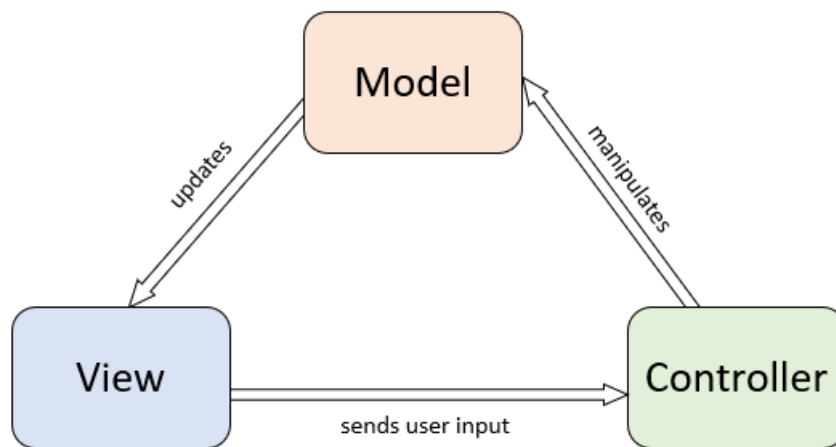


Abbildung 2.10: MVC Architektur Diagramm [ModelView Controller 2020]

2.5.1 Modulare Software Architektur

Der Begriff der Modularität beschreibt den Hauptteil der Architektur. Modularität bezeichnet die Aufteilung eines großen Ganzen in mehrere kleinere Teile, die als Module, Komponenten oder Bausteine beschrieben werden. Verschiedene Teile können bei geeigneter Funktion und Kompatibilität zusammengeführt werden oder über entsprechend definierte Schnittstellen interagieren. Unter der allgemeinen modularen Software-Architektur, bzw. Programmierung, versteht man ein Programmierparadigma. Dabei sollte die Software aus systematisch und logisch aufgeteilten Teilblöcke bestehen, diese werden Module oder Bausteine genannt. Der Aufbau der modularen Software kann praktisch in allen imperativen Programmiersprachen ⁶ Anwendung finden. Durch Modularität soll die Software bessere Kontrollierbarkeit, Übersichtlichkeit und Testbarkeit innerhalb großer Softwareprojekte gewährleisten. So sind die einzelnen Bausteine an sich unabhängig und für sich selbst zuständig, trotz dessen können diese mit weiteren Bausteinen kombiniert und verschachtelt werden.

In der Entwicklung sind die einzelnen Module eigenständig und separat zu planen, programmieren und testen. Dadurch ist der Rahmen des einzelnen überschaubar und leicht zu praktizieren. Nach erfolgreicher Testung der Module können die Einzelteile als eine Anwendung zusammengeführt werden, indem die einzelnen Module logisch über Schnittstellen verknüpft werden. Erst nach diesem Schritt ist die entstehende Applikation vollständig einsatzbereit.

Die modulare Programmierung gilt als Erweiterung des prozeduralen Ansatzes, dabei sind die Module in kleineren Ansätzen auf die Klassen der objektorientierten Programmierung zurückzuführen. [SCHREYER 2018]

Um eine Software-Architektur übersichtlich und strukturiert umzusetzen, gibt es demzufolge Entwurfsmuster, engl. Patterns, die zusätzlich für Ordnung innerhalb der Architektur sorgen. Ein für die Ausarbeitung relevantes Muster ist das MVVM-Pattern. (siehe 2.5.2)

⁶Programmierparadigma, das aus einer Folge von Anweisungen besteht.

2.5.2 MVVM

Das Model-View-ViewModel-Pattern ist, wie bereits erwähnt, ein Architekturmuster das den Entwicklern als Vorlage dient, um ordnungsgemäße, standardisierte und strukturierte grafische Oberflächen und das dahinterstehende logische System zu entwickeln. Es wird besonders für interaktive Systeme eingesetzt. Das MVVM-Muster wird allgemein als Weiterentwicklung des bekannten MVC-Patterns (der Abbildung 2.10 zu entnehmen) angesehen.

Grundlegend ist das Model-View-Controller-Muster, MVC, für die klare Abgrenzung zwischen Daten, deren Darstellung und Präsentation und den Nutzerinteraktionen für die graphischen Benutzeroberflächen gedacht.

Demnach ist auch das Model-View-ViewModel-Muster daran angelehnt, die Benutzeroberfläche von deren Logik zu trennen. Durch diese Trennung ist eine erhöhte Wartbarkeit und Testbarkeit gewährleistet. Die ermöglichte parallele Arbeitsweise von UI-Designern und Entwicklern unterstützt deren Arbeitsfluss und beschleunigt den Entwicklungsprozess. [MVVM-Pattern 2010]

Die Logik der Anwendung wird in den ViewModel-Klassen (2.4.2) dargestellt und dienen als Kommunikationsschnittstelle zwischen dem Model, dem Datenhalter, und der View, der Datenrepräsentation. Durch die Unabhängigkeit der einzelnen Module kann ein Unit Test vereinfacht durchgeführt werden, mitunter ein großer Vorteil des Patterns.

Die Ebenen, die das MVVM-Pattern enthält, werden in Abbildung (2.11) grafisch aufgeführt.

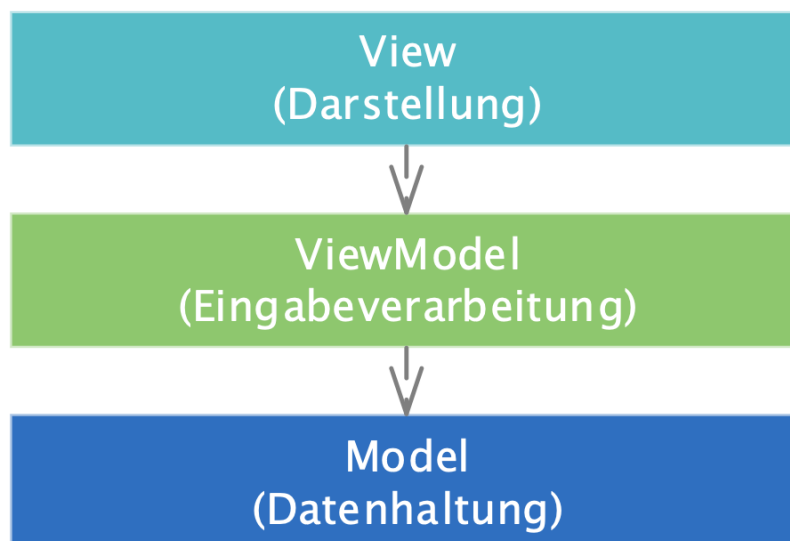


Abbildung 2.11: MVVM Architektur Diagramm [JAECKLE, GOLL und DAUSMANN 2015]

Die View-Ebene stellt die Informationen der Benutzeroberfläche dar, fängt die Benutzereingaben ab und gibt diese über Datenbindungen dann an das darunterliegende ViewModel weiter. Die View-Klasse enthält lediglich die Komponenten die die Oberfläche optisch aus machen und die dazugehörigen Datenbindungsschnittstellen, um die Informationen übergeben zu können. Ebenso werden über diese Schnittstelle Informationen geladen, um diese auf der Oberfläche repräsentieren zu können. Diese Vorgehensweise erleichtert das Austauschen der View ohne den

Code generell zu ändern.

Die ViewModel, bereits in Kapitel (2.4.2) beschrieben, ist dafür zuständig die Daten zwischen der persistierten Speicherung und der Repräsentation zu transferieren. Es stellt das Model für die View dar und gibt das eigentliche Model nach außen.

Die unterste Ebene, das Model, bzw. die Datenhaltung, stellt die Abbildung der Daten dar und giltet nicht als reine Abbildung der Datenquelle. Diese Daten werden für die visuelle Anwendung auf der View-Ebene benötigt und durch die ViewModel-Ebene bereitgestellt. Darüberhinaus können die Daten von dem Nutzer über die View manipuliert und an das Model zurückgegeben werden. Durch diese Möglichkeiten ist vorausgesetzt, dass das Model folgende Funktionalitäten bereitstellen sollte [EDER 2016]:

- Validierung der Daten
- Benachrichtigung bei Eigenschafts-Änderungen
- Verarbeitung von Business-Rules

Sozusagen dient das Model als Schnittstelle zwischen der anhängenden Datenbank und dem ViewModel, welches die Daten hält. Dazu kommt, dass das Model festlegt wie die Objekte aufgebaut sind, d.h. dort wird festgelegt welchen Informationen das Objekt beinhaltet.

Die nachfolgende Abbildung (2.12) veranschaulicht den Informationsfluss des MVVM-Patterns in einzelnen Schritten.

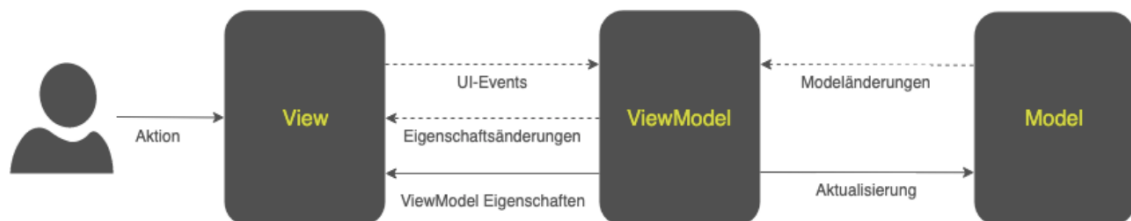


Abbildung 2.12: MVVM Informationsfluss [RICARDO 2019]

Basierend auf der Grundlage des Model-View-ViewModel-Patterns wurde die für das Projekt anvisierte Architektur, Android Architecture Components (2.5.3), entwickelt.

2.5.3 Android Architecture Components

Die Android Architecture Components bestehen aus einer Sammlung von Tools und Bibliotheken, engl. Libraries, die entwickelt wurden, um die Konstruktion von testbaren und stabilen Android-Apps zu vereinfachen. Mit der Einführung von Android Jetpack (2.4.2) wurden viele Komponenten, z.B. Bibliotheken und Frameworks, für die Entwicklung von Apps vereint. In den vielen Bibliotheken, die von Android Jetpack zur Verfügung gestellt werden, befinden sich auch die sogenannten *Architecture components*. Diese *Components* unterstützen und fördern die Umsetzung der MVVM Pattern basierten Architektur (2.5.2). Die zuvor genannten Bibliotheken ViewModel, LiveData und Room sind maßgeblich für die Gestaltung des angestrebten Musters

Model-View-ViewModel.

Google selbst empfiehlt im Rahmen der veröffentlichten „*Guide to app architecture*“ Dokumentation die Verwendung der *Architecture components*, um eine stabile und performante Applikation zu designen, da diese auf bewährten Prinzipien, unter anderem das *seperation of concers* Prinzip, basieren.

Folgender Abschnitt befasst sich im einzelnen mit den Komponenten, die für die *Android architecture* essentiell sind:



Abbildung 2.13: Android Grundkomponente [*Architecture Components* 2020]

Grundsätzlich sind Activities und Fragments nicht Teil der eigentlichen „*Architecture Components*“, sondern sind lediglich Grundkomponenten der Android App Entwicklung die die Benutzeroberfläche und die dazu korrespondierenden Layout-Dateien verwaltet.

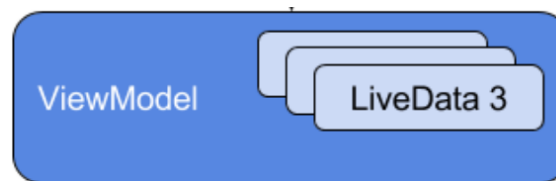


Abbildung 2.14: ViewModel Komponente [*Architecture Components* 2020]

Das ViewModel (siehe Abschnitt 2.4.2) verwaltet die relevanten Daten der UI. Diese Komponente hat einen unabhängigen Lebenszyklus dem Activity-Objekt gegenüber. Somit sind Daten bei unvorhergesehenen Unterbrechungen oder Fehlern der UI nicht betroffen und können bei erneutem Aufbau der Oberfläche ohne Probleme nachgeladen werden. Die im ViewModel enthaltenen LiveData Objekte (2.4.2) sind beobachtbare Daten-Container die eine Activity ohne expliziten Aufruf über Datenänderungen informiert. Durch diese Methode wird der Lebenszyklus der aktiven Komponente respektiert und berücksichtigt.

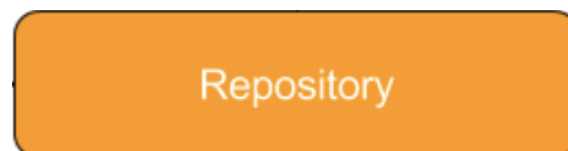


Abbildung 2.15: Repository Komponente [*Architecture Components* 2020]

Mit dem Repository wird ein „best practice“ Fall umgesetzt, der eigentlich kein Bestandteil der Android Jetpack Library ist. Diese Komponente ist dafür zuständig das ViewModel von der eigentlichen Datenquelle zu trennen, um Informationen aus verschiedenen Quellen beziehen und

synchronisieren zu können. Das Repository dient als Schnittstelle zu mehreren Datenbeziehungspunkten und verwaltet ein Verzeichnis zur Speicherung und Beschreibung von Objekten und Informationen. Zudem kapselt das Repositorium die Logik für das Persistieren und Erzeugen von Entitäten und Aggregaten von der Ansicht.

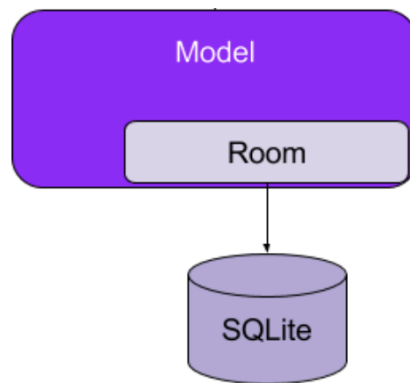


Abbildung 2.16: Model / Room Komponente [Architecture Components 2020]

Die *Persistence Library* Room (2.4.2) ist ein Datenbank *layer* auf der eigentlichen SQLite Datenbank. Zugriffe auf die Datenbank werden durch Room gekapselt und vereinfacht, indem viel mit Annotationen, z.B. *Entity*, *Dao*, gearbeitet wird. Der Abbildung 2.16 ist demnach zu entnehmen, dass die Bibliothek Room einen Großteil der Funktionen eines Models abdeckt und so die Model-Komponente repräsentiert. Anschließend auf das Datenbank layer ist die eigentliche Persistenzschicht, die SQLite Datenbank.

In einem fortlaufenden Architektur-Diagramm sind die Komponenten wie folgt angeordnet, um die Abhängigkeiten zwischen den einzelnen Teilen zu verdeutlichen.

In dem Diagramm 2.17 wird deutlich, welches Modul welche Abhängigkeiten besitzt, bzw. zu welchen oder von welchen Modulen Informationen bezogen werden können. Einzelne Komponenten können ihren Vor- oder Nachgänger niemals umgehen und sind so einer festen Hierarchie eingeordnet. Die Abbildung zeigt auch eine beispielhafte Ausführung mehrerer Datenbezugsquellen. Bei genauer Betrachtung des Diagramms sind alle Komponenten wiederzufinden, die ebenso in dem MVVM-Muster gegeben sind und das Muster prägen. Durch die vielen Komponenten und Schnittstellen ist modularität gegeben und trägt dazu bei, dass weitere Module angehängt oder ausgetauscht werden können.

2.6 Datenmodellierung

tbd...

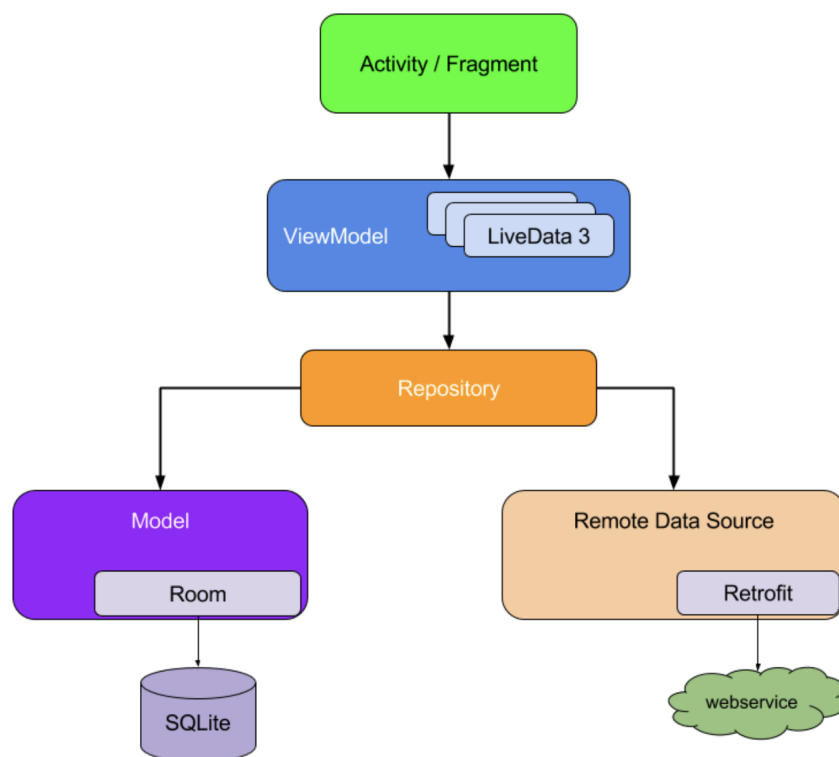


Abbildung 2.17: Android Architecture Components Diagram[*Architecture Components* 2020]

Kapitel 3

Konzeption

- 3.1 Arbeitsumgebung / Umfeld
- 3.2 Objekterkennung / Scan-Phase
- 3.3 Visualisierungs-Phase
- 3.4 Architekturkonzept
- 3.5 Softwarekonzept
- 3.6 Auswahl des AR Frameworks
 - 3.6.1 Google ARCore
 - 3.6.2 ARToolKit
- 3.7 Datenmodell

Kapitel 4

Umsetzung

4.1 Implementierung

4.1.1 Umgebungserkennung / Scan - Lernphase

4.1.2 Visualisierungs-Phase

4.2 Testdurchlauf - Test-Szenario

Kapitel 5

Evaluation

Kapitel 6

Fazit

Kapitel 7

Ausblick

Literatur

- ABRAHAM, Magid und Marco ANNUNZIATA [März 2017]. *Augmented Reality Is Already Improving Worker Performance*. <https://hbr.org/2017/03/augmented-reality-is-already-improving-worker-performance> [siehe S. 16].
- Architecture Components* [2020]. <https://developer.android.com/jetpack/guide> [siehe S. 34–36].
- ARCORE, Google [2020]. *Fundamental Concept*. <https://developers.google.com/ar/discover/concepts> [siehe S. 24, 25].
- AZUMA, Ronald T. [Aug. 1997]. *A Survey of Augmented Reality*. Massachusetts Institute of Technology [siehe S. 14, 15].
- B12, Studio [Juni 2020]. *VR und AR- Nur ein Hype oder echter Mehrwert?* https://studio-b12.de/content/projects/vrar/virtual-reality/pdf/2-Nur_ein_Hype_oder_echter_Mehrwehrt.pdf [siehe S. 7].
- BAUM, L. Frank [Feb. 1996]. *The Master Key - An Electrical Fairy Tale*. Hrsg. von Dennis AMUNDSON. Project Gutenberg-tm [siehe S. 15].
- BERGER, Daniel [2019]. *Live View: Google Maps schaltet AR-Navigation für alle frei*. <https://www.heise.de/newsticker/meldung/Live-View-Google-Maps-schaltet-AR-Navigation-fuer-alle-frei-4491407.html> [siehe S. 20].
- BITFORGE [Okt. 2019]. *Augmented Reality offiziell kein Hype mehr*. <https://bitforge.ch/augmented-reality/augmented-reality-offiziell-kein-hype-mehr/> [siehe S. 8].
- BOUVERET, Christopher und Sebastian HUMAN [Nov. 2019]. *Augmented Reality in der Industrie: Herausforderungen, Potenziale, Chancen*. <https://www.industry-of-things.de/augmented-reality-in-der-industrie-herausforderungen-potenziale-chancen-a-882695/> [siehe S. 9].
- BREHM, Frank [März 2017]. *VR-Angriff auf alle Sinne - Wie schmeckt die virtuelle Realität?* <https://www.mice-club.com/magazin/artikel/vr-angriff-auf-alle-sinne> [siehe S. 17].
- BÖTTCHER, Prof. Dr.-Ing Jörg [2020]. *Inertiale Messeinheit (IMU)*. <https://messtechnik-und-sensorik.org/8-sensoren-fuer-drehzahl-geschwindigkeit-beschleunigung-und-position-im-raum/> [siehe S. 24].
- CODELABS, Google [2020]. *Android Room with a View - Java*. <https://codelabs.developers.google.com/codelabs/android-room-with-a-view/> [siehe S. 28].

- DEVELOPER, Moody [2019]. *What is boilerplate code? Why and when to use the boilerplate code?* <https://fullstackworld.com/post/what-is-boilerplate-code-why-when-to-use-the-boilerplate-code> [siehe S. 25].
- DEVELOPERS, Google [2020]. *LiveData Overview*. <https://developer.android.com/topic/libraries/architecture/livedata> [siehe S. 27].
- [2017a]. *Room Persistence Library*. <https://developer.android.com/topic/libraries/architecture/room> [siehe S. 27].
- [2017b]. *Save data in a local database using Room*. <https://developer.android.com/training/data-storage/room> [siehe S. 27].
- DIN-461 [1973]. *Graphische Darstellung in Koordinatensystemen*. ISO - Internationale Organisation für Normen [siehe S. 23].
- DÖRNER, Ralf [Sep. 2019]. *Virtual und Augmented Reality (VR/AR) - Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Springer Vieweg, Berlin, Heidelberg [siehe S. 15, 17].
- EADICICCO, Lisa [Jan. 2020]. <https://www.businessinsider.de/international/apple-q1-earnings-call-tim-cook-augmented-reality-ar-2020-1/?r=US&IR=T> [siehe S. 9].
- EDER, Norbert [2016]. *MVVM – Model-View-ViewModel: Die Serie*. <https://www.norberteder.com/model-view-viewmodel-die-serie/> [siehe S. 33].
- ELGAN, Mike [2016]. *How Google’s Project Tango will change your life*. <https://www.computerworld.com/article/3018733/how-googles-project-tango-will-change-your-life.html> [siehe S. 23].
- FISCHBACH, Pascal [Aug. 2016]. *First Mover: Pokémon Go präsentiert der Welt “Augmented Reality”*. <https://conrenfonds.com/2016/08/24/first-mover-pokemon-go-praesentiet-der-welt-augmented-reality-die-abloesung-des-smartphones-oder-nur-spielzeug> [siehe S. 19].
- FREEDEN, Willi und Reiner RUMMEL [2016]. *Handbuch der Geodäsie*. Springer Spektrum, Berlin, Heidelberg [siehe S. 22].
- GAEVERT, Thomas [Mai 1856]. *Bibliographie Lyman Frank Baum*. <http://www.isfdb.org/cgi-bin/ea.cgi?162> [siehe S. 15].
- HASSELBRING, Wilhelm [2006]. *Software-Architektur*. Springer-Verlag 2006 [siehe S. 30].
- HIPP, D. Richard [2018]. *About SQLite*. <https://www.sqlite.org/about.html> [siehe S. 29].
- JAECKLE, Lukas, Joachim GOLL und Manfred DAUSMANN [2015]. *Das Architekturmuster Model-View-ViewModel*. https://www.it-designers-gruppe.de/fileadmin/Inhalte/Studentenportal/Das_Architekturmuster_MVVM_Text__1_.pdf [siehe S. 32].
- JECKLE, M. u. a. [2005]. *UML 2 glasklar*. Hanser Fachbuchverlag [siehe S. 30].
- KATO, Hirokazu und Mark BILLINGHURST [Okt. 1999]. *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System*. hitl.washington.edu [siehe S. 15].
- KRAUSS, Melanie [Feb. 2019]. *Europa baut Führung bei Industrie 4.0 weiter aus*. <https://www.maschinenmarkt.vogel.de/europa-baut-fuehrung-bei-industrie-40-weiter-aus-a-801681/> [siehe S. 9].

- LANHAM, Micheal [2018]. *Learn ARCore - Fundamentals of Google ARCore*. Packt Publishing Ltd. [siehe S. 23].
- LESWING, Kif [Okt. 2016]. *Apple CEO Tim Cook thinks augmented reality will be as important as 'eating three meals a day'*. <https://www.businessinsider.com/apple-ceo-tim-cook-explains-augmented-reality-2016-10?r=DE&IR=T> [siehe S. 9].
- ModelView Controller* [2020]. <https://bmu-verlag.de/model-view-controller-mvc-softwareentwicklungsmuster/> [siehe S. 31].
- MUNDT, Elisa [Jan. 2020]. *Marktstudie zu Augmented und Virtual Reality - Einblicke in die Marktreife immersiver Technologien*. <https://www.industry-of-things.de/vr-und-ar-werden-realtaet-a-898199/> [siehe S. 9, 21].
- MVVM-Pattern* [2010]. <https://entwickler.de/online/tipps-und-tricks-zum-mvvm-pattern-153374.html> [siehe S. 32].
- MÜHLROTH, Adrian [Nov. 2018]. *Was ist der Unterschied zwischen VR, AR und MR?* <https://www.techbook.de/entertainment/virtual-reality/unterschied-vr-ar-mr> [siehe S. 16, 18].
- NATESAN, N. [2019]. *How to implement Design Pattern – Separation of concerns*. <https://www.castsoftware.com/blog/how-to-implement-design-pattern-separation-of-concerns> [siehe S. 26].
- NIEMANN, Alexander [2020]. *IoC (inversion of control)*. <https://www.itwissen.info/IoC-inversion-of-control-Umkehrung-des-Kontrollflusses.html> [siehe S. 26].
- PINKER, Alexander [Sep. 2016]. *Kurve der Innovationen- der Gartner Hype Cycle erklärt*. <https://medialist.info/2016/09/16/innovationen-im-wandel-der-zeit-der-gartner-hype-cycle/> [siehe S. 7].
- RICARDO [2019]. *Warum Android MVVM eine gute Sache ist*. <https://blog.cosee.biz/2019/04/18/warum-android-mvvm-eine-gute-sache-ist> [siehe S. 33].
- SCHANZE, Robert [Okt. 2018]. *Was ist Mixed Reality? – Unterschied zu Virtual und Augmented Reality erklärt*. <https://www.giga.de/artikel/was-ist-mixed-reality-unterschied-zu-virtual-augmented-reality-erklaert> [siehe S. 18].
- SCHART, Dirk [2017]. *Augmented Reality für die digitale Zukunft der Industrie*. <https://www.vdi-wissensforum.de/news/augmented-reality-fuer-die-digitale-zukunft-der-industrie/> [siehe S. 21, 22].
- SCHREYER, Eric [2018]. *Modulare Programmierung*. https://de.wikipedia.org/wiki/Modulare_Programmierung [siehe S. 31].
- SELLS, Chris, Benjamin POIESZ und Karen NG [2018]. *Use Android Jetpack to Accelerate Your App Development*. <https://android-developers.googleblog.com/2018/05/use-android-jetpack-to-accelerate-your.html> [siehe S. 26].
- SUTHERLAND [Mai 1938]. *Biographie Ivan E. Sutherland*. <https://dl.acm.org/profile/81100265287> [siehe S. 15].