

Fantastische Reise: Ein erweiterter Realitätsansatz zur anatomischen Bildung der breiten Öffentlichkeit

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Manuel T. Schrempf, B. Eng.

Matrikelnummer 00920136

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao. Univ. Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Mitwirkung: Univ. Ass. Dr. Renata G. Raidou

Wien, 5. März 2020

Manuel T. Schrempf

Eduard Gröller

Fantastic Voyage: An Augmented Reality Approach to Anatomical Education for the General Public

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Media Informatics and Visual Computing

by

Manuel T. Schrempf, B. Eng.

Registration Number 00920136

to the Faculty of Informatics

at the TU Wien

Advisor: Ao. Univ. Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Assistance: Univ. Ass. Dr. Renata G. Raidou

Vienna, 5th March, 2020

Manuel T. Schrempf

Eduard Gröller

Erklärung zur Verfassung der Arbeit

Manuel T. Schrempf, B. Eng.
Kleinberg 20/2, 5151 Nußdorf am Haunsberg, Österreich

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 5. März 2020

Manuel T. Schrempf

Danksagung

Für mich zählt die Familie am meisten, und all den Familienmitgliedern die mich in irgendeiner Weise unterstützt haben nicht zu Danken, wäre eine Schande für mich selbst. Ich kann nicht dankbarer sein für das Zuhören meiner Familie bei den Erzählungen des derzeitigen Status oder Problemen und für das Nachfragen von hilfreichen und unhilfreichen Fragen was mir neue Perspektiven gegeben hat. Ich wurde von meiner Familie von der Außenwelt geschützt, um Forschung, eine Mobiltelefonanwendungsentwicklung, und diese Diplomarbeit über letzteres schreiben zu können.

Aber die meiste Unterstützung und die meiste Dankbarkeit, die ich fühle, ist für meine Freundin Malli. Malli bezeugte aus erster Hand unzählige Male das Auf und das Ab meiner mobilen Telefonanwendungsentwicklung und Diplomarbeitstellung.

Die Leitung von Renata Raidou war perfekt für mich. Renata gab benötigte Information und Unterstützung genug kreativen Freiraum durch viele Besprechungen und beaufsichtigte mit einem guten Sinn für Humor. Ohne der freien Unterstützung für die *ARnatomy* Anwendung hätte diese sich in eine andere, unfruchtbare Richtung entwickelt und diese Diplomarbeit wäre anders.

Die TU Wien hat mir die Möglichkeit gegeben, meine Fähigkeiten in menschlichen und technologischen Feldern zu entwickeln, zu erweitern, und zu schärfen. Ich bin dankbar für meine Zeit an der TU und in Wien selbst, speziell für all die Freunde und Erfahrungen, die ich gemacht habe. Darum beende ich die Danksagung mit einem Zitat von einem Wiener Wissenschaftler, der die Welt beeinflusst hat:

Die beiden Eckpfeiler des Glücks: Liebe und Arbeit.
— Sigmund Freud, 1856 - 1939

Acknowledgements

To me family matters the most and not mentioning and thanking all family members who have supported me in all ways would be a disgrace for myself. I could not be more thankful to my family for listening to my narratives of the current status or problems and asking more and also less helpful questions giving me new perspectives. I was sheltered by them from the outside world to do research, to develop a mobile device application, and to be able to write this master thesis about the latter.

But the most support I received and the most thankfulness I feel is for my girlfriend Malli. Malli witnessed at first hand and endured times without number the ups and downs in mobile device application development and the master thesis creation.

The guidance of Renata Raidou was perfect for me. Renata gave me the needed information and support as well enough creative freedom through meetings with a good sense of humour. Without the support for the *ARnatomy* application, it would have been developed into another, unfruitful direction and the master thesis would be different.

The TU Wien gave me the opportunity to evolve, extend, and sharpen my abilities in human and technological fields. I am grateful for my time at the TU and in Vienna itself, especially for all friends and experiences I have made. Hence, I end the acknowledgements with a quote from a Viennese scientist, who influenced the world:

The two cornerstones of joy: love and labour.
— Sigmund Freud, 1856 - 1939

Kurzfassung

Der Zweck dieser Diplomarbeit ist die Entwicklung einer mobilen, anatomischen Bildungsanwendung für die breite Öffentlichkeit, welche die passive, unüberlegte Mobilgerätnutzung zu einer aktiveren, lehrenden Nutzung mittels makroskopischer und regionaler Anatomie verlagert. Die neue, immersive Lernerfahrung mit interaktiven, 3D, Augmented Reality (AR) Anatomiemodellen synchronisiert diese in Echtzeit mit dem Gesicht des Benutzers. Individuen der breiten Öffentlichkeit können durch das digitale Sezieren ihrer eigenen Gesichtsanatomie geometrische, räumliche und textliche Anatomieeigenschaften lernen.

Immanente Eigenschaften des erstellten Lernprozesses sind selbst-gelenktes Anatomie-lernen, weniger geistige Belastung, ein Motivations-, Aufmerksamkeits-, Konzentrationsanstieg, länger erhalten bleibende Zufriedenheit, neues anatomisches Wissen, bessere räumliche Fähigkeiten und ein kurzfristiger Leistungsgewinn verglichen mit traditionellem Lernen. Die hohe Komplexität der menschlichen Anatomie beschränkt die synchronisierten Anatomiemodelle auf den Kopf. Nichtsdestotrotz können alle menschlichen Anatomiemodelle als Two Dimensional (2D) oder AR Three Dimensional (3D) Darstellungen betrachtet werden, jedoch ist nur die Kopfanatomie mit dem Kopf des Benutzers synchronisiert.

Die beantworteten Forschungsfragen sind “*Wie kann interaktive AR zur anatomischen Bildung der breiten Öffentlichkeit genutzt werden?*” und “*Wie viel und welche Anatomie kann in welcher Zeit mit der entwickelten Anwendung erlernt werden im Vergleich zu Arbeiten die am neuesten, technischen Stand sind?*”. Head Pose Estimation (HPE) verbindet AR verwaltet durch das Framework ARCore mit 3D Anatomiemodellen von Body Parts 3D (BP3D) und anatomischen Informationen von Foundational Model of Anatomy (FMA), um die breite Öffentlichkeit in der Anatomie zu bilden. Empfohlene Anforderungen aus der Fachliteratur werden erfüllt in der entwickelten Anwendung namens *ARnatomy*, welche eine zusammengefügte, anatomische, interaktive Lernerfahrung ist. Das Entwicklungsergebnis wurde in einer informellen Studie mit acht Teilnehmern evaluiert, welche zeigte, dass mobile AR zur anatomischen Bildung der breiten Öffentlichkeit genutzt werden kann. Sieben der acht Teilnehmer gewannen anatomisches Wissen in geometrischer, räumlicher und textlicher Form.

Abstract

The purpose of this master thesis is the development of a mobile, anatomical education application for the general public, which shifts the passive, unthoughtful mobile device usage to a more active, teaching usage by utilizing macroscopic and regional anatomy. The new, immersive learning experience with interactive, Three Dimensional (3D), Augmented Reality (AR) anatomy models synchronizes the models in realtime with the user's face. Individuals of the general public can digitally dissect their own facial anatomy to learn geometrical, spatial, and textual anatomy features.

Immanent features of the created learning process are self-directed anatomy learning, less cognitive load, a motivation-, attention-, concentration increase, longer preserved satisfaction, new anatomical knowledge, and better spatial abilities compared to traditional learning. The high complexity of the human anatomy restricts the synchronized anatomy models to the head. Notwithstanding, all human anatomy models can be viewed as Two Dimensional (2D) or AR 3D renderings, whereby only the head anatomy is synchronized with the user's head.

The answered research questions are “*How can interactive AR be used in anatomical education for the general public?*” and “*How much and what anatomy can be learned in which time with the developed application compared to state of the art works?*”. Head Pose Estimation (HPE) links AR managed by the framework ARCore with 3D anatomy models from Body Parts 3D (BP3D) and anatomy information from Foundational Model of Anatomy (FMA) to educate the general public in anatomy. Recommended requirements from professional literature are fulfilled by the developed mobile application named *ARnatomy*, which is a jointed, anatomical, interactive learning experience. The development result has been evaluated in an informal study with eight participants, which showed that mobile AR can be used for the anatomical education of the general public. Seven of eight participants gained anatomical knowledge in a geometrical, spatial, and textual form.

Contents

Kurzfassung	xi
Abstract	xiii
Contents	xv
1 Introduction	1
1.1 Motivation	1
1.2 Problem Definition and Aim of the Work	3
1.3 Methodological Approach and Contribution	4
1.4 Structure	7
2 Related Work	9
2.1 Augmented Reality	9
2.2 Anatomy Applications without Augmented Reality	29
2.3 Anatomy Applications with Augmented Reality	31
2.4 Model Datasets	40
2.5 Face Tracking	40
2.6 Mobile Databases	44
2.7 Software Development Kits	46
3 Methodology	49
3.1 Selection of Anatomy Perspectives (APs)	50
3.2 Selection of Databases	53
3.3 Selection of Head Pose Estimation Technology	57
3.4 Development and Design Choices	61
3.5 ARanatomy Rendering	65
4 Implementation	81
4.1 Applied Technologies	81
4.2 Selection of Databases	82
4.3 Selection of Head Pose Estimation	85
4.4 Development and Design Choices	91
4.5 ARanatomy Rendering	94
	xv

5	Results	103
5.1	Development	103
5.2	Informal Evaluation	104
5.3	Case Studies	106
5.4	Interview	119
5.5	Limitations	123
6	Discussion and Conclusion	129
6.1	Discussion	129
6.2	Conclusion	146
7	Future Work	149
A	Appendix	155
A.1	Body Parts 3D and Foundational Model of Anatomy Column Names	155
A.2	Projects and Issues on Github	156
A.3	Instructions and Questions	158
	List of Figures	161
	List of Tables	167
	Glossary	169
	Acronyms	177
	Bibliography	183

Introduction

Three main information technologies influencing our perception of reality exist. Mixed reality fuses the reality with the virtual one, Virtual Reality (VR) replaces the whole reality, and Augmented Reality (AR) partially converges VR into reality for better immersion and interactivity [1, 2, 3]. Within this thesis, AR is utilized in a mobile device application for the exploration of anatomy on the user's own body. A mobile application with face-synchronized AR anatomy learning capabilities to educate the general public was inexistent before this master thesis. The AR superimposition is composed with texts as well as models from a flat file collection, namely the Body Parts 3D (BP3D) [4], and a Database (DB), namely the Foundational Model of Anatomy (FMA) [5]. Section 1.1 treats the motivation to write this master thesis, Section 1.2 treats the problem definition and aim of the work, Section 1.3 treats the methodological approach and contribution, and Section 1.4 treats the structure of this master thesis.

1.1 Motivation

The Flynn effect is the steady Intelligence Quotient (IQ) increase of the population in the 20th century [6]. However, an inversion started in the final decade of the last millennium and is caused environmentally. The environmental dependence of the Flynn effect coheres with other IQ decline hypothesis, a quality decline in education, high mass media consumption, unhealthy nutrition, and poor health, as well as an increase of immigration [7], media exposure, and mobile device usage [8, 9, 10]. The increasing time of mobile device usage can be utilized fruitfully for digital education of the general public and to shift passive, unthoughtful mobile device usage to a more active, autodidact one. The key of mobile media consumption is what media with which mindset is consumed to lower or even resolve the harmful effects of IQ decline [8, 9, 10]. In this master thesis, models and texts focus on anatomical education of the general public with AR on mobile

devices.

A fundamental understanding about anatomy is essential for doctors, nursing personnel, and other medical employees to conduct their work [1]. During the lifespan of individuals from the general public, topics, issues, or interest in the own health and anatomy may arise [1]. AR on a mobile device can bring enlightenment about the functionality of the human body to the general public and particular to patients. Patients with more knowledge about anatomy have more benefits. Studies [1, 11, 12] show an improvement in the communication between doctor and patient before and even more after the diagnosis. Successful information exchange during consultation is positively linked with patient compliance during therapies and affects the overall treatment success [11, 12].

The medical staff can also use the developed application named *ARnatomy* during a consultation to show anatomy to patients on the patient's own body. Additionally, professional treatment with patient involvement becomes more popular and ought to be coupled with fundamental anatomical knowledge of the patient for a satisfying outcome [13]. The information exchange between patient and the medical staff can be facilitated by using digital, anatomical depictions causing less anatomy abstraction in the minds of patients. A digital depiction of anatomical structures in AR can be separated into different layers, has less discomfort, and less cadaveric complexity as real anatomy [14]. Digital anatomy can have more interactivity than Two Dimensional (2D) or physical models, multiple Anatomy Perspectives (APs), wider access, and lower costs [14] than real anatomy.

Constructivism and embodied cognition is a major learning theory and states that *active learning constructs spontaneous knowledge* as well as lower cognitive load than traditional learning [13, 15]. Interaction with virtual anatomical objects is analogous to the kind of exploration users would have with real anatomical objects. The resemblance and association to the embodied cognition and learning benefits from own body movements as perceptual and motoric insights, which enhance not only the learning process in general, but also with interactive AR.

Correspondence between the depiction of the real and virtual world are important for the realism of AR. Interaction with the virtual objects enriches User Experience (UE), allows user navigation, and enables manipulation. The interaction with virtual objects designed for learning influences knowledge gain positively and is more satisfactory compared to textbook images [16]. Interactive Three Dimensional (3D) anatomy systems have shown substantial learning progress for medical students, especially in understanding anatomical relationships and spatial visualization [13, 16]. The progress originates from the direct manipulation, which is better than passive viewing for learning anatomy in a 3D VR environment [17]. In this master thesis, memorization is categorized in three different terms. Short-Term is defined in the Oxford dictionary as *occurring over or relating to*

a short period of time [18], and is in this master thesis considered to be shorter than a day. Medium-Term is defined in the Oxford dictionary as *occurring over or relating to a period of time of moderate length* [19], and is in this master thesis considered to be less than a month but longer as short-term. Long-Term is defined in the Oxford dictionary as *occurring over or relating to a long period of time* [20], and is in this master thesis considered to be more than a month. The interactive 3D anatomy systems are not superior to traditional learning in the matter of medium-term and long-term memorization. Anatomy education systems have a short-term performance improvement confirmed by Preim and Saalfeld [13], but lack evidence for their medium-term and long-term impact on learning [16]. Chen et al. [21] performed in the field of medical mixed reality a search for papers considering the following topics; treatment, education, rehabilitation, surgery, training, interaction, mobile, display, registration, and tracking. The topic *mobile* had the least publications. Most research of the influence on learning with AR has been conducted on medical students. No mobile application has face pose estimation, an anatomy superimposition as AR with Head Pose Estimation (HPE), and interaction with rendered anatomy to navigate, select, and display anatomical information, which empowers embodied cognition.

The purpose of this master thesis is *to generate ARnatomy for mobile devices, which provides access for the general public to the education of anatomy with all the aforementioned benefits of AR*. Albeit the contained scientific anatomical information [4, 5] in *ARnatomy*, its private usage or its usage in the medical field can only educate and support users [1]. *ARnatomy* cannot replace the consultation of a physician.

1.2 Problem Definition and Aim of the Work

The mobile device usage along with time spent on the phone increases among the world population, whereby the former is reciprocal to the user's available cognition [8]. This is a problem for every mobile device user, since cognitive functioning is influenced negatively by mobile devices [9]. From the overall mobile device usage, media consumption has 95 % and within that 95 % social media is consumed the most [10]. Education on mobile devices has an active user involvement requiring more cognition compared to a passive mobile device usage. An *Android* mobile application with 3D, anatomical AR educating the general public is inexistent. The necessary technologies needed to build an immersive, mobile, AR education application exist. HPE, Google's ARCore [22] AR framework on a mobile device, an extension of ARCore named *Sceneform* [23], DBs with anatomical texts, DBs with anatomical models, and many APs on anatomy [13] are the required technologies conflated as one application. Based on the aforementioned information, this master thesis answers the following research questions:

1. *How can interactive AR be used in anatomical education for the general public?*
2. *How much and what anatomy can be learned in which time with the developed application compared to state of the art works?*

The application name *ARnatomy* is a portmanteau of one abbreviation of the used technologies and the topic anatomy. Capturing a human face triggers the AR superimposition enabling the user to interact, learn, and explore anatomy via simple taps, swipes, and pinch gestures on all 3D axes with one's own mobile device. A textual search is included, the entered characters show related anatomy suggestions and clicking a suggestion opens a detailed view. In the detailed view, the anatomical synonyms and definitions can be read and the hierarchy can be viewed. The anatomical hierarchy can be explored by using simple *up* or *down* popup menu taps concerning the previously found anatomy. At any time, the created anatomy hierarchy loaded through exploration can be viewed as AR visualization.

To achieve minimal latency, model loading from the DB, model rendering, model animations, and User Interface (UI) animations are all executed as asynchronous background threads synchronizing each other on attachment points to decrease the dangers of UI thread interruptions and bad UE. Asynchronicity is proportional to source code complexity, both depend on the number of asynchronous tasks executed and the single task complexity. With the implementation, the user can educate oneself about bones, muscles, organs, blood vessels, and nerves of the head. Further anatomical systems, regions, and layers concerning the rest of the body or lymph system, their circulation, et cetera are reserved as future work due to time constraints.

Anatomy rendering, AR, interaction on mobile platforms, their restrictions, and inherent benefits for learning require efficient communication and integration between each other, otherwise latency influences the interaction, AR experience, and learning process negatively. HPE, model mapping, and AR rendering have not been used together in an AR mobile application. Intuitive interaction and anatomical information representation are important for usability, learning experience, and progress and this master thesis aims to empower all latter three.

1.3 Methodological Approach and Contribution

The methodological approach consists of five major steps executed in sequential order. First, different APs on anatomy were examined to identify the macroscopic and regional APs [13], which are best fit for the education of the general public through a mobile AR application. Second, a model DB and textual DB was searched to have anatomical content and information in the application. Preprocessing of model DB and textual DB was a normalization and conflation to an own DB comprising the whole human anatomy hierarchy and all hierarchical information. Third, state of the art research about existing anatomy education tools, effect on learning of educational tools, learning of anatomy with education tools, effect on learning of AR, AR technology, HPE, all their features, and their interoperability. Based on the conducted research, the design of the application encompassing UI, model interaction, and the mental picture of anatomy

created through interaction with anatomy and learning. Fourth, implementation and testing of the application with the frameworks of ARCore [22] and *Sceneform* [23] as well as the DBs BP3D BP3D and FMA [5] with consideration of software design patterns, guidelines, and designs for the Back-End (BE) and Front-End (FE). Fifth, Informal Evaluation (IE) of the developed application with two case studies and questions afterwards for all eight participants. Visual facilitation of the complex human anatomy and its rendering is beneficial for the learning and comprehension of anatomy. Illustrative rendering can create meaningful renderings of anatomical structures and objects containing spatial and functional information of themselves as well as their proximity [24]. Compared to conventional rendering, more visualization opportunities for regions of interest are possible. A distinction is made between low level illustrative rendering, boundary emphasis and saliency, and high level illustrative rendering, visibility adaptations through transparency or removal. For a clear view and orientation, high illustrative rendering is utilized along with external labels [25] for the rendering of AR models.

With automatically facial landmark tracking, interactive models are rendered in AR allowing the user to perform a digital dissection [26] and an assemblage for discovery-based learning [27]. Discovery-based learning is active learning and exploration, which *constructs spontaneous knowledge* according to the major learning theory constructivism and embodied cognition [13, 15]. Active learning through interaction and deep immersion through AR bring a higher user satisfaction, increased attention, and a more ascending learning curve [13, 16, 17, 28] for the target group being interested individuals of the general public. Alternatives would be non-mobile anatomy education tools without direct interaction on the models, without AR for higher immersion, and without any synchronization with the user's face or body. Examples of other anatomy education tools lacking one, two or all previously mentioned features with synchronization of the user body, but without access for the general public, e.g. *mirracle* [29] and *WithTeeth* [30]. Anatomy education tools without synchronization of the user body, but with access for the general public exist too, e.g. *Zygote Body* [15], *The Online Anatomical Human* [31] and *The Open Anatomy Browser* [32].

The main difference and likewise advantage of the developed application to the aforementioned applications is the included constructivism and embodied cognition [13, 15] paired with direct, interactive manipulation and digital dissection of AR anatomy. In no other anatomical education application, the user can interact with anatomy models synchronized with the users's own body, which achieves a higher level of immersion. A minor, but still unique difference to other anatomical applications, is the application availability for the general public on a mobile device. No additional setup, as *Microsoft Kinect* in *mirracle* [29] for example, is required to start the developed application. Another minor difference to other anatomical applications is that the user can switch easily between three activities displaying first, the synchronized AR showing the overall facial anatomy, second, the single, isolated view of only one anatomy object, and third,

the detailed textual anatomy hierarchy view. Each single of the three activities was designed to create a single mental picture and to amend it through the switching to one of the other activities. Creating only one mental picture for the subject to be learned results in faster and a more efficient learning than creating multiple mental pictures for the same subject [33]. The anatomical education with the developed application should be easy, interesting, and enlightening. Participants in Preim and Saalfeld [13] showed appreciation for up to date anatomical information, i.e. high resolution renderings, textual keyword search, and automatic suggestion. The aforementioned anatomical information is included in the application for high usability and UE being a highly interactive, intuitive, non-complex UI for a low complexity and a low distraction from learning. To meet the requirements determined by Preim and Saalfeld, the newest 3D flat file model collection BP3D [4] with the attributes free, scientific, and anatomical is used for anatomy model rendering. The contained models of BP3D have an index linking them to FMA [5], a medical anatomy ontology DB. The BP3D and the FMA are normalized and migrated to constitute an own, single DB encompassing models along with their hierarchy, labels, synonyms, and definitions. Minimum latency is the basis for unrestricted, immersive usage preventing user dismissal at the beginning [34]. Building on top of minimum latency, a non-distracting, non-complex, intuitive UI aims for an interesting UE empowering user satisfaction, focus, and immersion for positive learning effects.

For low computational demand and to ensure a standard usability on a mobile device it is recommended to develop aligned to the *Google Java Style Guide* [35], *Google Developer Policies* [36], and *Google Material Design* [37]. The Java style guide ensures future readability and maintainability of the source code. Adherence of additional developer policies helps to include proven practices in the field and compliance with the guidelines and rules of *Google Play* [38] and to ensure a sparse quality standard for the BE. Similar for the FE, adherence with the material design creates an appealing UI proven in practice in many available applications as Threema [39], Airbnb [40], Asana [41], and many others. Maybe, this mobile device application will be available someday on *Google Play*. Compliance with the aforementioned guidelines and policies will be beneficial on the upload to *Google Play* as less issues, delays or code refactoring will hopefully be necessary.

Simple software design patterns, namely factory, singleton, observer, and builder patterns [42], are used. The integration of simple patterns is easier than more complex software design patterns [43, 44], because of the code structure from *Android* projects and the one from *ARnatomy*. An IE [45] with eight participants was conducted with *ARnatomy* to receive feedback on intuitiveness, usability, functionality, discovery-based learning [27], learning effects, gained anatomical knowledge, and to discover design flaws. Two case studies specifically designed for *ARnatomy's* IE had to be executed by all eight participants, who received brief instructions on what but not how to do the single case study components. The intention was to observe, if a participant can orient oneself in the application, is able to perform discovery-based learning, self-directed [46] anatomy

exploration, and to get more critic feedback in the interview afterwards. The contribution of this master thesis is to provide access to anatomical education for the general public with an AR mobile education application synchronizing interactive models with textual information through HPE with the user's own face for an immersive learning experience benefiting from constructivism and embodied cognition [13, 15].

1.4 Structure

The remaining master thesis consists of Chapters 2 to 7 and an Appendix. Chapter 2 treats related work as state of the art concerning the topics AR, its medical applications, mobile DBs, and HPE. Chapter 3 treats the methodology of the preparation and execution of application implementation. Chapter 4 treats the implementation from all used technologies and their mutual effect on each other. Chapter 5 treats the result of the development, its limitations, IE design, and the IE. Chapter 6 discusses the development result, the effects on learning, the evaluation result delivered by the IE, consisting of two case studies and an interview, answers thereby the two research questions, and contains the conclusion of this master thesis. Chapter 7 treats possible improvements, amendments, development directions, and application fields as future work. Chapter A contains all raw BP3D, all FMA column names, all projects, all issues created on *Github*, which affected this thesis as hyperlinks, and all IE questions. All abbreviations are written in their full length at their first occurrence in each chapter.

Related Work

The literature on Augmented Reality (AR), its subcategories, effect on education, usage in the medical field, face tracking, and Database (DB) technologies influenced the implemented application and gave innovative ideas for future work. The aforementioned topic order relates to the technology usage within this project and is intended for best comprehension. Section 2.1 treats the state of the art of AR encompassing AR in general, with the usage of AR in anatomy education, and the effects of AR on user education. Section 2.2 treats anatomy applications without AR. Section 2.3 treats anatomy applications with marker-based AR and markerless AR. Section 2.5 treats face tracking encompassing face recognition and Head Pose Estimation (HPE). Section 2.4 treats the anatomical model datasets found. Section 2.6 treats mobile DBs and their features and performance in Create, Read, Update and Delete (CRUD) operations. Section 2.7 treats current AR frameworks and gives an overview of their capabilities.

2.1 Augmented Reality

The first publication about AR was from Azuma in the year 1997 [3], but technology was not as far as the imagination of the researcher. Beside AR, other continuums with one or multiple virtual parts were imagined and are created today. First, virtuality, mostly referred as *Virtual Reality (VR)*, is generated by software and completely digital and virtual without any real objects. Second, Augmented Virtuality (AV), where a human or a real, non-digital or non-virtual object is placed inside a virtual environment [47]. Third, AR where virtual objects are placed inside a real environment creating an illusion of a coexistence from virtual objects in reality. Fourth, the unaltered, uninfluenced, non-digital, non-virtual reality itself perceived solely by our own, human senses without any other technology. In AR, the reality is captured by a camera and perceived through a display. The captured scene on the display is extended, amended or *augmented* by virtual objects rendered as superimposition. In other words, an illusion of a coexistence

between virtuality and reality is manifested. Mixed reality is a hypernym for AV and AR as depicted in Figure 2.1. Over time the realization of AR became possible on mobile devices, which can deliver a continuous video stream through their integrated cameras to a display wherein virtual models can be rendered.



Figure 2.1: Notions between virtuality and reality [48, 49].

With more computational resources of mobile devices and advanced environment tracking techniques, AR became first popular in companies developing products for the general public [50]. Challenges are realtime rendering and enterprise applications to make the technology more known with dedicated AR ecosystems as well as platform economy. Key factors for the success of AR are easy to use ecosystems, support for developers, and users who use the technology actively and do not have the technology just installed on their devices [50].

AR requires landmarks to be tracked and registered for orientation. Currently, two approaches named marker-based and markerless exist to maintain the realistic illusion and immersion created by AR. By relying on metric data from the mobile device's camera, sensors or data created by the tracking system, a plasticity of augmentations facilitating, guiding or adapting to user perception can be created programmatically [51, 52]. Within the context of AR, plasticity describes the application adaption to the environment, platform, content or the user. The diffuse and spectral lights used in rendering can be adapted to the detected ambient light intensity to create higher realism and more immersion in the captured scene.

Other AR plasticity are possible to change the User Experience (UE) or the AR based on sensory information and not only on user interaction. Adaptive illumination on AR renderings can be created, short distances to models can change model transparency to show its insides, details, display an info text, highlight the model by scaling [51], displaying its outlines, blurring the background, fade the background colour to greyscale, change the sound output volume, or utilize other effects. Sound recorded by the mobile device can change the rendering of models in all possible ways mentioned above. AR plasticity becomes more and more important as single AR applications transit on and on from one particular task to a persuasive UE [52]. AR can only be as good as the used

models, their rendering, and registration technology, especially in application cases where precision is needed as in the medical field, e.g.. Subsection 2.1.1 treats the state of the art in general marker-based AR without anatomy, Subsection 2.1.2 treats the state of the art in general markerless AR without anatomy, and Subsection 2.1.3 treats the state of the art concerning the effect of AR on learning.

2.1.1 Marker-Based Concepts and Applications

Marker-based AR needs fiducial markers in the real world detected by a camera. The Oxford dictionary defines fiducial as (*especially of a point or line*) *assumed as a fixed basis of comparison* [53] for an algorithm performing fiducial landmark tracking. In the AR context, *fiducial* refers to *an accepted Point of Reference (PR)* from the application meaning markers can be nearly anything from what camera position and rotation in space can be derived by a corresponding marker detection algorithm [54, 3, 55, 56]. Knowledge of camera and marker position enables rendering objects relative to the current camera perspective resulting in the illusion of AR in the device display. The act of fiducial marker tracking for AR registration formed the term *marker-based AR*.

Within the marker field, a distinction between different marker-based technologies is made. Marker-based technologies can be barcode-based, template-based, and model-based, whereby all three technologies can be used for AR registration. Barcode-based AR markers can be One Dimensional (1D), e.g. Universal Product Code (UPC) code depicted in Figure 2.2 and European Article Numbering (EAN) codes, and Two Dimensional (2D), e.g. Denso's Quick Response (QR), International Data Matrix's Data Matrix, United Parcel Service (UPS)'s Maxi Code, Zebra's Ultracode, and colourZip Media's *ColourCode* In Figure 2.3 [57].



Figure 2.2: Example of a 1D marker [58], which can be used for AR orienting its renderings relative to the 1D marker position in the captured scene by the device camera.

The first barcode generation has been 1D, depicted in Figure 2.2, and is still widely used. Black lines with different widths on an arbitrary background being the *zero* and *one* bits form a certain geometrical pattern, which can be converted into a binary format for recognition and decoding. Recognition of 1D barcodes is fast for machines due to their simple design, but comes with disadvantages, visual entropy encoding limitations compared to its

successor, the second barcode generation, no visual information for humans on first sight, and unreadable by untrained humans. The second barcode generation has been 2D and has the advantage of a higher entropy compared to 1D barcodes, more information, more information density, and more data types are supported. The third barcode generation is the second one in colour, has even more entropy than single coloured 2D barcodes, and is designed for one single barcode of the third generation within an observed area. Nearby recognition is assumed and therefore coloured 2D barcodes as well as image codes are susceptible to scale changes or distortions [57]. AR markers have a higher distortion resistance, which results in a more tangible interaction and better recognition stability of tracking software up to 30 markers. The entropy of AR markers had to shrink in order to gain more stability against rotation and compared to barcodes, the information capacity is small. In Figure 2.3 different markers, their recognition and decoding process, depicted as arrows, show different marker types.

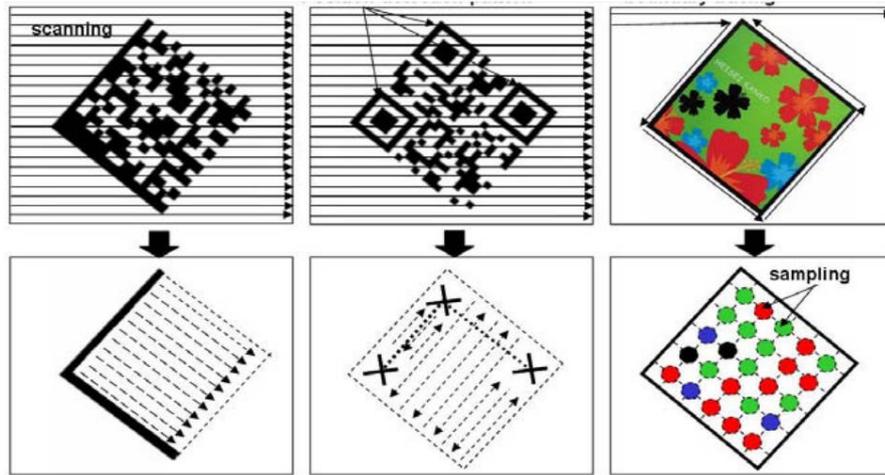


Figure 2.3: Left: Decoding procedure and direction of the data matrix inside a 2D marker. Middle: Decoding procedure and direction of the data matrix inside a QR code. Right: Decoding procedure and direction of the data matrix inside a colour code [57].

One may think by observing Figure 2.3 markers have to be square or rectangular, but markers can be any shape a tracking algorithm can detect, e.g. circles, triangles or a company logo can also be used as marker. For robustness with fast detection black and white markers having a high contrast and a bold frame are the majority [59]. Barcode and AR marker technology was fused to create *ColourCodeAR* [57] to have the benefits of a higher encoding entropy and better tracking. Continuous recognition and tracking of AR markers is needed for smooth application execution and a good UE. A small marker with a high recognition range, meaning easy and fast detection, is the requirement, which constrained the search for the best barcode technology used for the combination with AR.

The novelty of *ColourCodeAR* [57] detection is that a lot more marker identifications can be made compared to barcodes not using different colours. *ColourCodeAR* can contain up to 13 times more information than QR codes and the smallest detected marker from a fixed 50 centimeter distance can be a 15 millimeter square. The smallest detected *ColourCodeAR* marker from a varying distance can be a 3 millimeter square and the detection distance is enlarged by approximately 10 centimeters. *ColourCodeAR* increments the side angle from 55 to 90 and has a detection range magnification from 30 to 90 side viewing angles [57]. Any markers with angles from 30 to 55 cannot be detected with standard colour codes but with *ColourCodeAR*, they can be detected.

The improvements result in more robustness and more performance speed compared to other colour coded markers as well as more supported scanning positions. Users tend to start the scanning from the most comfortable scanning device position for the users but not for the scanning process. Quad vertex detection detects the vertices of a quad and *ColourCodeAR* uses the quad vertex detection algorithm in each frame. Previous quad vertex positions are reused on a tracking loss [57], and if the usage of previous quad vertex positions is successful, the colour code decoding with the *Kanade-Lucas-Tomasi (KLT) feature-tracking algorithm* is performed.

Template-Based or *template markers* are image markers and markers depicting a silhouette. Within template-based markers, features are extracted and compared with stored marker templates form a DB. Template-based markers require a sophisticated design and implementation, since their correct scanning is often unreliable through unconsidered similarity properties. Markers concealing their contained information by stereograms with robust decoding for AR are a new emerging field founded by Nguyen and Yeap [59]. A stereogram, autostereogram or *magic eye picture* is a 2D image, which displays a hidden, floating object when viewed from a particular angle. Viewing from a particular angle is also known as cross viewing or parallel viewing. The stereogram consists of reiterated patterns, where each pattern's distance to a view point is proportional to another view point. The angle between the two view points results in a different focus point behind the image for each single view point, which creates the perceived depth illusion of stereograms. Most stereograms are designed for the mean view point distance between human eyes, which can be considered as the aforementioned view points.

Reading information with a computer from a stereogram is an *ill-posed inverse optical problem* [59] because of the matching similarity and structural ambiguity of the reiterated patterns used. The stereogram used as marker for AR can be printed on a regular 2D black-border square with an 10 % thick border for recognition. The residual 90 % of the image is again separated in 50 and two 25 % areas, where the 50 % is a fixed center image with the stereogram. The two 25 % images are left and right reiterated greyscale overlay pattern halves creating the stereogram beside the fixed center image as depicted in Figure 2.4.

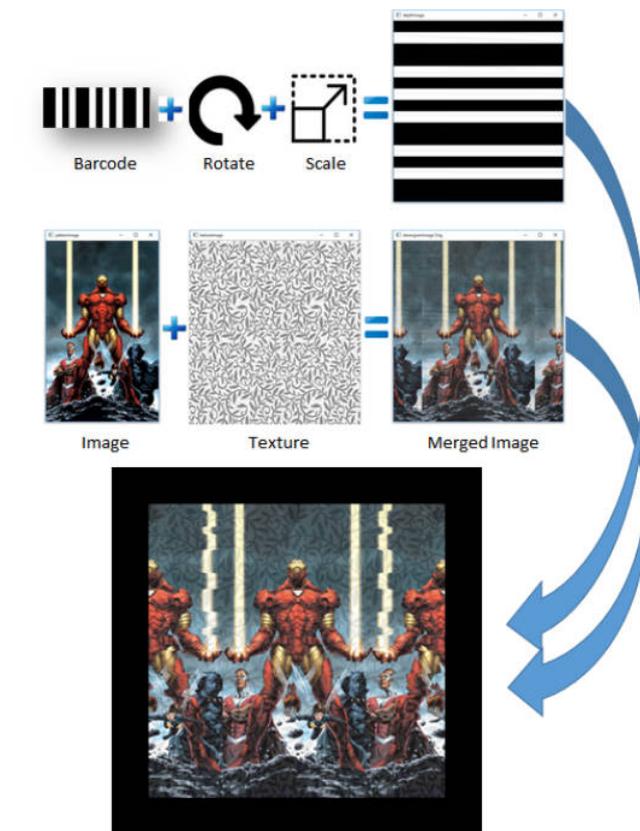


Figure 2.4: Creation process of a stereogram [59].

Advantages of stereograms [59] are the informative, meaningful, human-readable center image, the information inside the two side images readable by a computer, and center image as well as side image independence from each other. Further advantages of stereograms [59] are the encoding support of multiple 1D or 2D barcodes, decoding with approximately 95 % similarity to its original, no discrepancy on high image blurs, contrast or brightness levels, heavy raindrops decrease the similarity to 85 %, sand hard noise, being image disruptions, decreases the similarity to 82 %. Disadvantages is more generation complexity, more computational demand, and more expensive colourful printing ink on colour images. The effect when left or right information image parts are missing through occlusion or partial damage was not treated by Nguyen and Yeap [59]. The AR *StereoTags* can transport information to their users who only have to take a glance at the markers to know their context or contained information, if the picture represents the content of the marker or the marker's context. Companies could use *StereoTags* as additional advertisements and information transportation for users or as shipment data transportation for delivery services. AR is not limited to indoor environment, outdoor AR uses Global Positioning System (GPS) for orientation, and device sensors to know in which direction the current Field of View (FoV) of the camera

is aligned [60]. The location accuracy and the users perception can be improved, even when the target outdoor object is occluded by another object, especially in urban environments. In some cases, AR reportedly compromised the depth perception negatively. Images of physical objects, being in urban environments buildings or statues, can be used as fiducial AR markers. The object's exterior is photographed beforehand and the extracted features are then matched with the features extracted from the current video stream. If a match for the physical object is found, the corresponding AR content is shown.

Small environmental illumination changes can cause severe malfunctions of outdoor AR. The environmental illumination variables, as object photograph date and time, object illumination, current illumination inside the video stream, viewing angles, headlights of cars, weather as rain, snow, dust, smog, camera light sensor, and camera configurations, influence the outdoor AR registration process [60]. During the night, the influence of all variables on the outdoor AR registration process is increased. Kasapakis et al. [60] make outdoor marker-based AR more robust by taking sensitivity of illumination from outdoor AR into account. If the normally executed physical object registration from outdoor AR fails, an FoV estimation is activated as additional physical object registration. The physical object in the urban environment is seen, an accurate AR FoV projection upon the actual side of the detected, physical object is performed via geolocate raycasting as depicted in Figure 2.5. In geolocate raycasting, rays are sent from the GPS location of a device into the direction of the device's magnetic sensor [60]. If the ray intersection points hit building polygons, the device's FoV can be estimated. Building polygons are polygons of buildings in previously captured images stored on an DB.

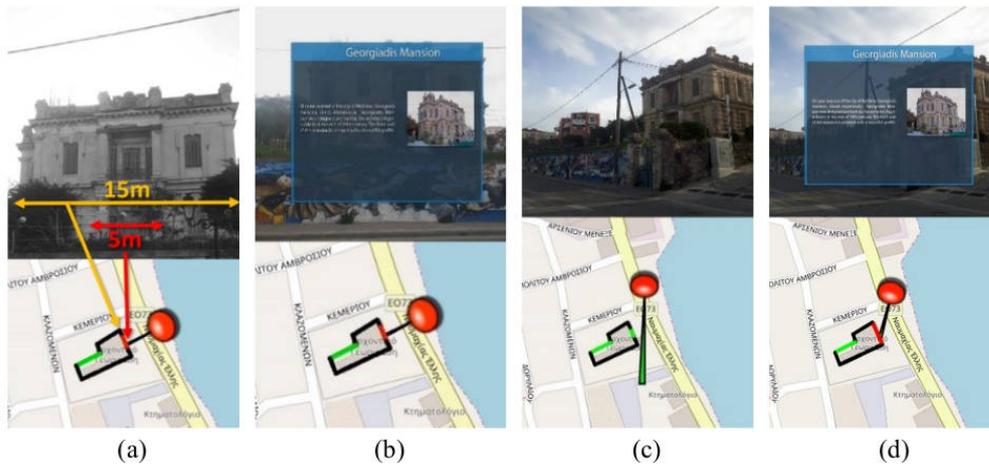


Figure 2.5: Outdoor AR: a) Polygon side red center adjustment. b) AR content shown due to correct building identification. c) No content due to incorrect FoV not hitting the center of the buildings facade. d) False positive match as a) due to perspective distortion [60].

The geolocate raycasting should hit the center of a physical object facing to the device

for a high FoV estimation precision. With the FoV estimation, inaccurate GPS locations can be counterbalanced and sometimes be recognized in the first place. Only buildings with clearly visible sides, excluding smaller built-in facade objects, can be detected with geolocative raycasting. With geolocative raycasting, the illumination fluctuations in urban environments are decreased [60]. Other marker-based AR registration approaches are model-based AR, which are not to be confused with markerless, also known as feature-based, AR treated in Section 2.1.2. In model-based AR registration, the system tries to register one or multiple models with the camera-captured scene by comparing extracted features with preprocessed ones from all supported models [56]. The registered model features serve as fiducial markers for model-based AR rendering.

As clear differentiation between model-based AR, marker-based AR, and markerless AR, model-based AR is considered to be a subcategory of marker-based AR. Markerless AR generates own, structured features from the video stream, as coordinate map for example, and does not use any preprocessed model features as model-based AR does. The classification consequence from the differentiation between model-based AR, marker-based AR, and markerless AR is that the paper from Vlaminck et al. *A Markerless 3D Tracking Approach for Augmented Reality Applications* [61] has the word *markerless* in its title but performs model-based AR and is therefore treated in this section. Time of Flight (ToF) cameras measure distances with the time of flight method. Light impulses are sent and the camera measures the time of an impulse to come back to the camera from an object. By knowing the speed of light and the flight time of a light impulse to come back, the distance from the camera to the object can be calculated. Depth information from a handheld sensor, mimicking the view of a head-mounted Time of Flight (ToF) camera, is used to calculate the position of the sensor relative to the observed model as depicted in Figure 2.6.

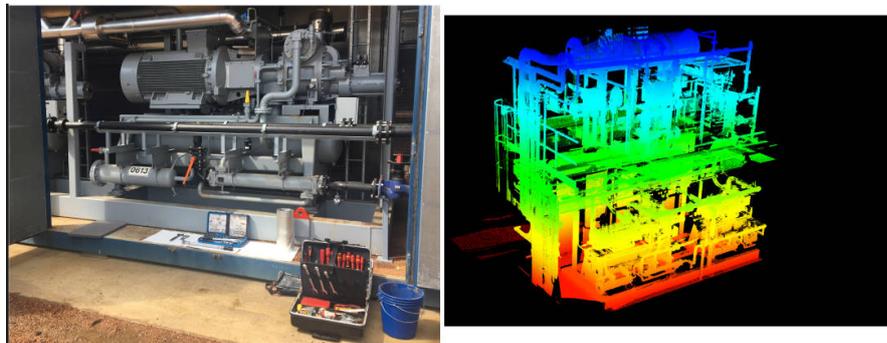


Figure 2.6: Left: The object in reality. Right: The generated model ground truth used for AR registration [61].

A ToF camera measures the distance, i.e. depth information, to an object by emitting and receiving light impulses. The depth information registered by the ToF camera gives an inaccurate camera pose relative to the observed object. Due to the inaccurate determined

camera pose, the derived view area of the model is also inaccurate. Preprocessing in the form of a full model laser scan and further scan data processing is required for more accuracy. A full object laser scan creates the object's Three Dimensional (3D) model. Surface elements (surfels) is in 3D computer graphics an alternative to polygonal modeling. An object is represented by a dense set of points or viewer-facing discs holding lighting information. From the a point cloud of object's 3D model surface element (surfel)s are created and utilized as assistance to gain speed and accuracy in localization and orientation [61]. The latter is achieved though creating an octree of the point cloud where each level has a lower resolution. On each depth scan an octree traversal is performed through the different resolutions until the actual viewed area with a resolution down to one centimeter is found. The one centimeter resolution is sufficient to calculate the viewed area of the model. Further scans utilize the previously known location and search in the current and subtrees first for a faster view area location and orientation.

The previously treated preprocessing of a 3D model and a full object laser scan emphasizes the model-based and rather complex fiducial marker-based AR tracking. Vlaminck et al. [61] compare their approach with actual markerless AR tracking technology, namely Large Scale Direct - Simultaneous Localization and Mapping (LSD-SLAM) and Oriented Features from Accelerated Segment Test (FAST) and Rotated Binary Robust Independent Elementary Features (BRISQUE) Simultaneous Localization and Mapping (ORB-SLAM), treated later in Subsection 2.1.2. In the benchmark video material the 3D Computer Aided Design (CAD) models are missing for the Vlaminck et al. [61] solution making the accuracy and speed inferior to LSD-SLAM and ORB-SLAM. A detailed evaluation with the state of the art is impossible, since static laser scans of the physical accessible industrial machinery used is currently unavailable. The performance is acceptable for AR applied in industry and on machinery, where digital plans of the used machinery are known. One application case may be the guidance of a layman without any knowledge about his task with precise visual instructions [61].

An advantage of the approach by Vlaminck et al. [61] is that environments with low or no illumination have no impact on the performance due to the use of solely depth information. But relying on depth sensor is a disadvantage too, most standard mobile devices in the year 2019 still lack depth sensors shifting model-based AR registration approach further into the industrial field of application. The newest registration technologies in AR are markerless being one of the most difficult and important technologies in the AR field, due to its superior performance in rotation, orientation, lighting, and partial superimposition detection compared to marker-based AR [2].

2.1.2 Markerless Concepts and Applications

The registration process of markerless AR depends on real world information or natural features, which allow an application to create a higher immersion due to the lack of fiducial markers as required by marker-based AR [2]. A fiducial marker has interactivity

limitations, decreases the intrinsic immersion in AR, and draws the users attention away from the application and its content to the markers. Since a marker only serves as fiducial point for the AR registration, but has no contribution to content, application logic, nor immersion, the user can just to start a markerless AR application without preparation, e.g. placing a marker somewhere. The diffuse sunlight of a grey sky is sufficient for markerless AR, but dark places where no texture can be detected by the AR registration process are still a problem. Observing AR from the software engineering perspective, marker-based AR is less complex than markerless AR, since recognition and registration rely on the marker detection in marker-based AR instead of the feature detection in markerless AR. Markerless AR technology must generate a precise, virtual, 3D map with coordinates from natural features to eliminate latency and rendering displacements.

The 3D map generation in markerless AR is also called *registration* as in marker-based AR. The 3D map coordinates make the calculation of the camera pose and the positioning of AR models possible. The latter are realistically positioned as superimposed illusion seen through a display. For the 3D map, iterative recognition refinements, sequential coordinate updates, relocation on tracking loss, loop closure and error corrections are possible [62, 63, 64, 65, 66]. As literature publications, the complexity of both approaches, their extrinsic as well as intrinsic problems suggest, markerless AR was researched and developed after marker-based AR [3, 47, 56]. Depending on the implementation and usage of the extracted features, markerless AR can be more flexible and faster as marker-based AR.

Observing AR from the user perspective, Brito and Stoyanova [54] researched the effect of marker-based and markerless AR technologies on the user. The user arousal when interacting with marker-based and markerless AR is the same but the latter created more positive facial reactions. Positive reactions are related to the easier setup of markerless AR without any third objects, i.e. markers. Starting markerless AR is easier as starting marker-based AR and appears to the user more intuitive and more integrated in the real world. Markerless AR has consequentially a higher interactive experience as marker-based AR, and if the user is familiar with the rendered content, the user knowledge about the content and markerless AR symbiotically cause a better UE [54, 56]. In other words, markerless AR has a higher associated usability, and content familiarity reinforces the effect magnitude.

One advantage of marker-based AR is the easier content relocation, a user just has to position the marker somewhere else and the content follows accordingly [47]. In a markerless AR application, the marker positioning is considered to be a feature, which has to be programmed intentionally. Users interacting with markerless AR rendered products tend to recommend the product or brand on a higher rate. The higher product rate may reside in the higher associated usability and partially comes from the higher suggested innovativeness of the product by markerless AR [54]. All the aforementioned disadvantages of marker-based AR along with the advantages of markerless AR contributed to the

development decision to use markerless AR in this master thesis. Visual Simultaneous Localization and Mapping (V-SLAM) [64] in general generates a continuously updated map of extracted feature points from a camera video stream of an unknown environment and provides therefore a coordinate system that can be used for camera location and AR model positioning. Compared to the state of the art approaches in marker-based and markerless AR registration, 3D map information generated via ORB-SLAM [67, 64] achieves higher performance and higher accuracy. The two techniques FAST and BRIEF constituting *Oriented FAST and Rotated BRIEF (ORB)* of ORB-SLAM have high performance and low computational demand. FAST is used for the search of corner key points who match visual features in realtime detection systems. Hence, FAST [68] is well suited to solve the AR registration problem, which is the tracking of objects and the mapping of features from the real world into a virtual representation, i.e. mostly a 3D map. The single corner orientation of detected features is determined with the centroid technique [69], which delivers one dominant result.

BRIEF [70] is a feature description using binary tests between picture elements (pixels) inside a smoothed subimage of the camera-captured reality. The high robustness of BRIEF to perspective distortions, illumination changes, and blur comes with a weakness to in-plane rotations [69]. An exhaustive search is used to remove redundant information from the in-plane rotations and to generate a better signature for the key points inserted into the 3D map. The combination of FAST and rotated BRIEF with V-SLAM assembles ORB-SLAM [64]. The detected key points by *ORB*, as seen in Figure 2.7, are used to create and afterwards improve the 3D key point map of the current camera-captured scene.

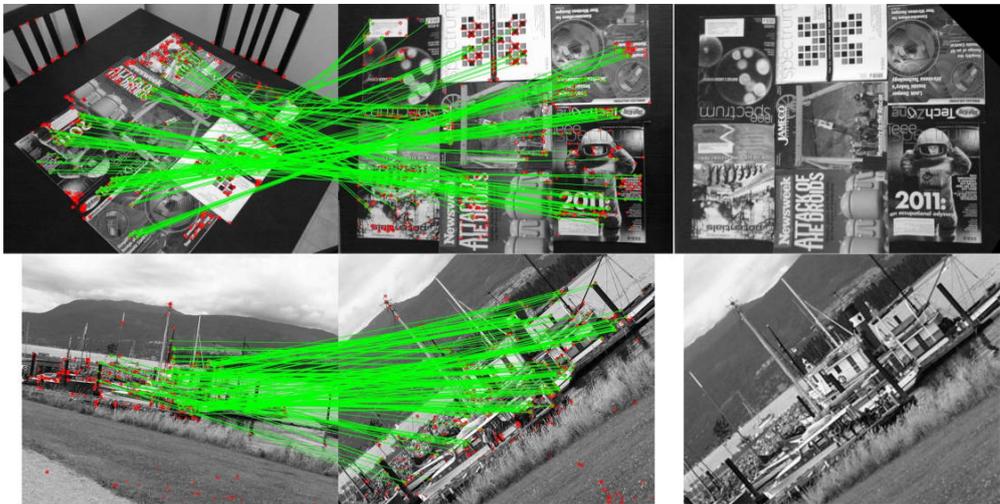


Figure 2.7: Green lines indicate matched key points identified by ORB on a view point change. Red dots indicate unmatched points [69].

Gao et al. [67] add a 3D dense map to the ORB-SLAM 3D key point map for the

markerless AR registration stabilization. The virtual models seen through a display have a more stable location and a smoother movement. Iterative Closest Point (ICP) is an algorithm employed to minimize the difference between two or more point clouds. The Hough voting algorithm is added to the V-SLAM for better detection and recognition rates of objects and Iterative Closest Point (ICP) is applied for higher accuracy in the model's transformation matrices. With the ICP, the modified ORB-SLAM can detect objects, know the object location and camera location simultaneously. ORB-SLAM provides the generated and continuously refined 3D coordinate system in realtime, which makes ORB-SLAM a suitable technology delivering AR the coordinates required for model rendering. The jitter phenomenon in general makes rendered models jump in different directions on a small distance frequently, e.g. the models jitter. The jitter phenomenon in AR originates small but different tracked position registrations on each frame, which is resolved by ORB-SLAM [67]. However, the need for depth information is a disadvantage as most computers or mobile devices lack a depth camera.

Loop closing in digital maps is when a map path intersects with itself or another path and the former is conflated with one of the two latter *closing* the path loop. Relocalization in digital maps is localizing a position, which was known beforehand but was lost. Map reuse is reusing or refining already inserted key point coordinates of a map. Bundle Adjustment (BA) is the optimization of line of sight bundles in a 3D scene or 3D map recorded by multiple cameras, where each camera can have a different perspective. Mur-Artal and Tardós [63] created an open source project ORB-SLAM 2 [63] encompassing loop closing, relocalization, and map reuse, and a lightweight mode for map reuse without map updates, no Bundle Adjustment (BA) is performed. In the field of AR registration the ORB-SLAM 2 [63] technique has the highest accuracy and performance and is applicable on monocular, stereo, and Red Green Blue Depth (RGBD) camera systems. Hence, laptops and mobile devices with cameras are in the application range of ORB-SLAM 2 [63]. The Back-End (BE) is executed on a dedicated thread and is based on BA, enabling the system to run on standard Central Processing Units (CPUs), and to build a global sparse reconstruction of the created and refined 3D map with a relatively low computational demand similar algorithms. If the Front-End (FE) , also running on a dedicated thread, input is a monocular or stereo video stream, the missing depth information for a dense map can be derived when the camera is moved. One camera view and the corresponding, refined sparse 3D map is depicted in Figure 2.8.

In the graph theory, spanning tree includes all vertices in a graph with the minimal number of edges. A covisibility graph is formed by connecting all consecutive key frames captured by a camera in a map to a spanning tree. The upper picture of Figure 2.8 depicts the local map key points as green dots. The lower picture of Figure 2.8 depicts the global covisibility graph key points as black dots and the local map key points as red dots. In lower picture of Figure 2.8 the key frames from the camera positions forming the covisibility graph are depicted as blue dots for vertices and green lines represent edges. The generated map from ORB-SLAM 2 [64] with test video footage was equal

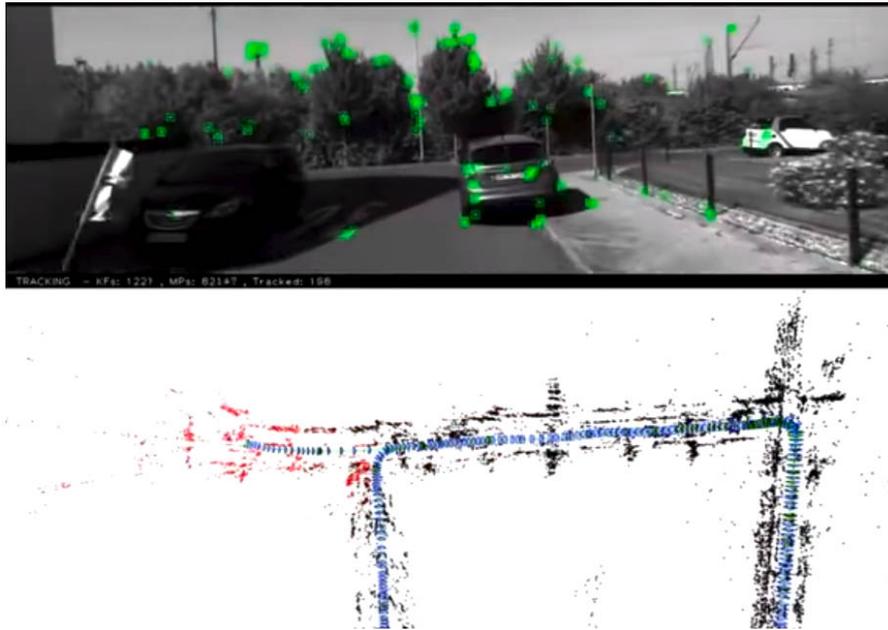


Figure 2.8: Visualization of the generated ORB-SLAM 2 map from a moving vehicle [71].

to the ground truth map. Other V-SLAM systems have tracking interruptions with the same test video footage or fail on initialization [64]. A lightweight mode is included for long-term localization solely running on one thread in a known environment without big environmental changes.

Especially on mobile devices having low resources the aforementioned lightweight mode could be used as AR registration method [63]. Camera localization in the lightweight mode is performed by matching 3D points from the previous frame with the current frame forming a continuously refined 3D map making the key point coordinates more precise. Scale Drift (SD) is an interpretation change over time legitimately applied to entries inside a certain scale [72], e.g. a map, model, measure or criterion. Through scale drift, gathered data differs over time more and more from its ideal or reality. A delayed initialization performs a delay until enough data is gathered for the initialization to start. The lightweight monocular version of ORB-SLAM 2 has Scale Drift (SD) and problems with high distances inside camera-captured frames. The problem with high distances originates in the inability of creating key points and knowing their distance from just one frame to another frame because of delayed initialization. A self driving car with ORB-SLAM 2 on a highway has delayed initialization and would need additional depth information for its start. Smaller environments, a room for example, favor a lightweight ORB-SLAM 2 usage as no long distances exists nor a delayed initialization happens. Additional stereo cameras or depth information make ORB-SLAM 2 more robust to delayed initialization with a zero SD and achieve in 9 of ten cases a higher accuracy than

the current state of the art [63]. On a monocular ORB-SLAM 2 application, accuracy is gained through loop closing, which diminishes the SD [63, 72]. But, on a video sequence going 800 meters into one direction, the SD accumulates and the translation error raises up to 10 % of the 3D map key points. Qualified feature point selection [64] assumes that the majority of all extracted feature points belong to the background of a camera-captured scene. Only background feature points of a camera-captured scene, which are identified as pairs within the current and previous key frames, are used for camera pose calculation controlled by the covisibility graph. With the qualified feature point selection, Wang and Shang [64] improved the monocular ORB-SLAM 2 accuracy and robustness for dynamic environment changes by qualified feature point selection. The reversed idea originated in the optical flow algorithm mostly used to calculate the x -direction and y -direction velocities of foreground feature points within a covisibility graph. The K-means clustering is used to minimize the feature point cluster to a prototype of itself resulting in less computation time [64]. Only the cluster with the most matching feature points from the covisibility graph is used for camera pose calculation as depicted in Figure 2.9.

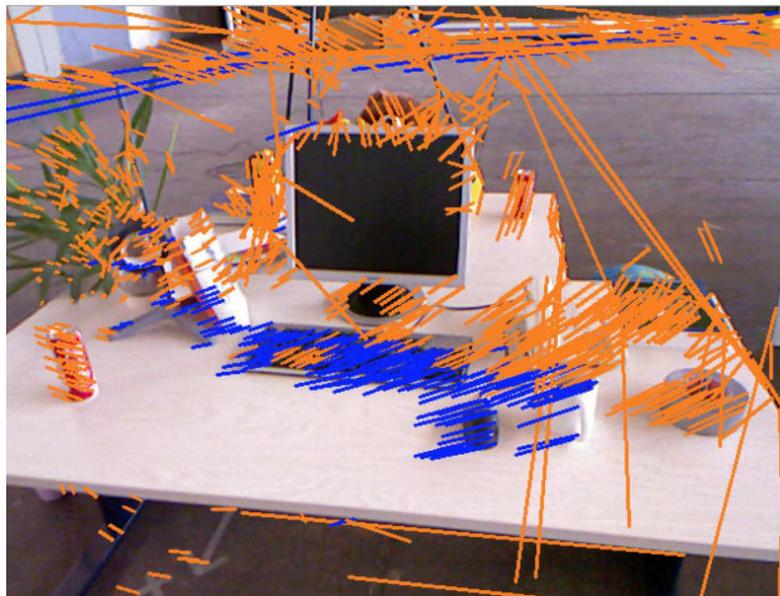


Figure 2.9: Each single line end is a matched feature point. The blue line clusters are used for camera pose calculation and the orange lines are excluded [64].

For more robustness to object superimposition, more rotation invariance, smaller positioning errors, and excluding extreme values a K-means classification on x values and y values of a key point in the covisibility graph is used. Monocular ORB-SLAM 2 with qualified feature point selection is the state of the art in constructing a map in each scenario with tracking the cameras position in an unknown environment [64] in terms of accuracy. Manly intended for robotic mapping and navigation, the capabilities of ORB-SLAM 2 can enhance multiple other fields requiring knowledge of their surroundings, one of the

best examples is AR. Large Scale Direct - Simultaneous Localization and Mapping (LSD-SLAM) is a direct monocular SLAM technique: Instead of using landmarks or feature extraction, it directly operates on image intensities both for tracking and mapping [73]. Other realtime monocular tracking or mapping approaches besides V-SLAM achieving nearly the same performance are rare. ORB-SLAM 2 relies heavily on feature extraction, inside the field of V-SLAM there is another technique who operates directly on image intensity named LSD-SLAM. An advantage is to use all the information contained within an image and a certain robustness to environment with few texture [65].

Akin to ORB-SLAM 2, LSD-SLAM can be used with monocular, stereo, and RGBD camera systems. A disadvantage is the high computational demand, whereby the performance of LSD-SLAM on CPUs is poor. By prepending a filter to remove pixels with an insufficient gradient the performance of LSD-SLAM on a CPUs rises and making LSD-SLAM usable without a Graphical Processing Unit (GPU). As ORB-SLAM 2 has three threads, LSD-SLAM is consists of three components. The first component tracks the constant velocity of pixel and their motion model. The second component performs a depth map estimation between the camera pose and the current key frame from previous key frame iterations. The third component performs covisibility graph optimization in the form of loop closure and key point minimization similar to ORB-SLAM 2. The final created depth image is depicted in Figure 2.10.



Figure 2.10: The final build depth map at the top with the corresponding real scene at the bottom [65].

More robustness to rotation and more speed are gained in comparison to the traditional LSD-SLAM, but in terms of accuracy in relation to computational demand ORB-SLAM 2 is still more efficient. A new binary feature with a more advanced trade-off between efficiency and accuracy for a speedup of LSD-SLAM was developed by Yang et al. [66]. Two threads are required in LSD-SLAM by Yang et al. [66], one for tracking and the other

one for mapping. Pyramidal Lucas-Kanale (PLK) finds the location of a certain pixel area from one image in a sequential image by the area's intensity. PLK computes a velocity vector for the area known as optical flow. Window size is proportional to accuracy and robustness, bigger windows smooth the area, smaller windows capture more motion. The pyramidal version blurs and shrinks the image recursively. From the pyramids peak down to the biggest image the standard Lucas-Kanale is applied and each result is the initial value for the next pyramid level [66]. The first thread is the Pyramidal Lucas-Kanale (PLK) with depth and spatial constraints as optical flow, who estimates the camera pose and its translation and rotation in 3D space at 30 Hertz. The Inertial Measurement Unit (IMU)'s angular velocity dictates image resizing resulting in high resolution image selection for high blur degrees and smaller images for low blur degrees as stabilization [66].

The second thread is the depth mapping, which continuously updates the inverse depth map and compares the map to the previous frame and its distance. A threshold for the distance triggers a new key frame generation and a depth map update. The Frames Per Second (FPS) controls reciprocal the relevant image gradient and therefore the accurate depth estimate as well as the optical flow accuracy. In other words, the higher the FPS are the smaller the key frame images become. Tracking loss activates immediately relocalization, which is trying to find a matching key frame in the already generated covisibility graph caching only several hundred key frames but not millions as other image retrieval approaches. The features correct location is reliably found in the new frames coming from a video stream with translation and rotation of the camera and enables the building of a depth map similar to V-SLAM. The feature and key point translation errors in the generated map of Yang et al. [66] come close to the improved monocular ORB-SLAM 2 accuracy on the TUM dataset of Wang and Shang [64] and range to five centimeters, but with double the speed.

In terms of speed, it is probably true that the state of the art in constructing a map of the camera's position in an unknown environment is the Yang et al. method [66]. In terms of accuracy, the claim of Yang et al. [66] that their method is state of the art is doubtfully. The doubt arises through the ORB-SLAM version used for comparison by Yang et al [66]. A correct performance comparison in accuracy and speed with the state of the art monocular ORB-SLAM 2 [64] was not done, but rather claimed by Yang et al. Because, Yang et al. [66] published their work in November 2018, whereas their compared ORB-SLAM method [62] was published in 2015, reintroduced with refinements as ORB-SLAM 2 in 2017 [63], and improved in its monocular usage for accuracy by Wang and Shang [64] in August 2018. The Wang and Shang ORB-SLAM method [64] was not used for comparison by Yang et al. [66] making their claim to be state of the art brittle. As no evidence can disprove or prove the performance superiority in speed or accuracy of both systems, both approaches remain as the state of the art with a slightly better tendency to Wang and Shang [64]. The Wang and Shang method [64] is the newest, currently most efficient ORB-SLAM and not the first ORB-SLAM [62] used

for comparison by Yang et al. [66].

2.1.3 Effects on User Education

The human anatomy is complex and acquiring knowledge about all involved physiological systems is alike. Traditionally anatomy education is performed by cadaver dissection, which is a systematic exploration through removal of muscles, fat or other tissue of a conserved human cadaver to learn more about gross anatomy [26]. Within the philosophy of education, *constructivism* assumes that learners construct knowledge out of their experiences influenced by already constructed, prior knowledge and experiences often created by the learner's social and cultural environment. Embodied cognition is a theory assuming that the motor system, the perceptual system, bodily interactions with the environment, and the assumptions about the world shape cognition itself. The human motor system performing during a cadaver dissection influences human cognition [74]. First, a full 3D view is given during a cadaver dissection and second, tactile experiences are made both equal to active learning, which creates spontaneous knowledge according to the major learning theory constructivism and embodied cognition [13, 15].

Constructivism and embodied cognition have furthermore the benefit of a lower cognitive load than traditional learning. Cadaver dissection is a validation of already acquired knowledge and more elaborations and explorations for further knowledge gain and insight can be made. Obtainment, preservation, and dissection with professionals of a human cadaver is expensive and direct empirical evidence for the effectiveness of dissections for learning anatomy is inexistent [26], and done because no practical alternatives exist. When the visualizations of AR become more detailed and representative in human anatomy, the traditional dissection on a real human body may be exchanged with a dissection on a virtual human body. Heretofore, inherent embodied cognition during cadaver dissection and its practice over many decades in universities speak strongly as justification.

The general public cannot make real cadaver dissections, rather digital ones on the individual's own body captured by a mobile platform video camera. Some of the benefits of a real cadaver dissection, namely full 3D view, active learning via interaction, and more elaborations during a traditional cadaver dissection as well as explorations, could be taken along with the AR digitalization, which has the benefits of inexpensiveness, maintenance, expandability, filtering, and fewer real world resources [26]. Acquisition costs of human cadavers are high, but will pay off in the long term, since low costs for software changes, manpower, less time for preparing and designing new features as well as software parts are inherent [27]. AR can create a highly extensible view, which depends on the used models, model texture, and anatomical information through rendered models and information not interfering with the learning process [27]. As many theories harvesting the benefits from AR seem to be sublime, but the empirical research about how AR affects learning gives a fragmented image and no concrete answer [26]. The immersion of AR plays an important role in the researched state of mind from user during learning in various works

[26, 75, 27, 76]. Immersion of learning, meaning the extent of several, involved senses in the learning process, is proportional to the engagement of learning [76]. Benefits of learning with AR [27, 76] are motivation increase, meaning more eagerness and interest to engage with new technology, learning content, and physical interaction as driver for augmented emotional engagement. Further benefits of learning with AR [27, 76] are attention increase, concentration increase, meaning a higher concentration degree induced through physical interaction, and a preserved satisfaction increase by the user. Similar teaching concepts to learning with AR are student-centred or discovery-based learning, meaning students are more independent and educators guide and adjust the learning process instead of educating students by content imparting and collaborative learning [27]. Important to note is that depending on the subject AR is not always the best choice, e.g. learning to play a music instrument for example. AR individualized the exploration, knowledge gain, instructions, supported independence of the exploration, and created an invitational environment for communication and cooperation.

Creative learning is additionally supported by AR. Problem-solving, absorption, and exploration abilities with new knowledge are improved by learning with AR. Anatomical, spatial abilities and memorization of spatial contents are improved of participants learning with AR and decreased drop-out quotes of subjects in learning facilities where spatial knowledge is needed. The memory of participants is improved, during questionnaires more content could be memorized with AR in comparison to traditional memorization [27]. Empirical research on the effect of AR on learning [26] suggests that AR's merit in educational systems is in the field of impaired people. Munoz-Montoya et al. [77] discovered that users who learned with AR remembered virtual objects and their location significantly better in a previously unknown environment than other participants who used photographs of the same scene.

A meta-analysis on the educational effectiveness of 3D visualization learning software [75] encompassing 32 identified papers having 77.8 % usage of random control tests on the concerned subject of learning with 3D visualizations was conducted. Within the analysis, the correlation of learning benefits by learning the content with 3D visualizations instead of traditional learning is confirmed. Especially with anatomical content, knowledge, spatial abilities, mental rotation, and rationalness are significantly influenced by learning with 3D visualizations for women and men. Compared to nearly all other educational methods, learning with 3D visualizations delivered better spatial knowledge acquisition, but factual superiority of 3D visualization learning is unconfirmed due to inconsistent outcomes in all analyses of the learning [75]. Nevertheless, a general trend in anatomical learning towards learning with 3D visualizations is observed and better results are reported in user satisfaction and the perceived effectiveness of the learners compared to traditional teaching methods [27, 76]. Many studies [16] found an improvement in identification and localization of anatomical structures through learning with volumetric 3D visualizations of participants. The subset of participants having the poorest spatial

ability in pretests had a significant disadvantage in later tests concerning learning with 3D visualizations [16]. Limitations are that most participants were junior medical students and the rapid change of 3D visualization programs may lead to different outcomes in the future. Participants's short-term performance in anatomical tasks meaning accuracy and time needed for completing improved too. Khot et al. [16] showed the opposite, digital objects are inferior to physical ones in the case of pelvic anatomy, VR is as effective as 2D depictions. Khot et al. [16] conclusion is traditional learning of normal anatomy is superior to computer based 3D learning. Researchers who invented the *Anatomy Glove Learning System* [16] state their system has the same effect on knowledge and introspection confidence of students as learning hand anatomy traditionally. While other studies [16] show the same equality on learning 3D or traditionally anatomy structures, it is important to note that the participants who participated in 3D learning modalities reported a higher satisfaction.

Of the 30 papers [16] identified considering 3D learning from over 90.000 papers, only two participants were from more than one school and only eleven papers had over 100 participants. Most papers do not explain their theoretical basis, do not give a logical explanation as justification for their findings, nor provide evidence for the validity of their methods. Randomized controlled studies, controlled studies or crossover studies are used for support of their research design. Impact assessment is based on testing knowledge learnt by answering quiz questions, theoretical examinations, and practical examinations. But the two questions “*Has 3D learning benefits compared to traditional learning?*” and “*What are medium-term and long-term effects of 3D learning?*” remain unanswered. Unfortunately and altogether, the quality of most studies in the current paragraph giving multiple results on 3D anatomy learning lack a profound and definite basis [16] as the various results suggest. Limitations are that the PubMed, EMBASE, and the Web of Knowledge DBs were searched with seven keywords and only the English language excluding *allied health disciplines such as nursing, physiotherapy, and occupational therapy* [16] as well as other languages. Khot et al. [16] are confident alongside of their paper's limitations that the unlikeliness of outcome changes through adding more data is significant and remaining. Two proven statements can be made, first, 3D learning increases short-term accuracy and speed of fulfilling anatomical tasks, and second, the interest of research in 3D learning grows as the increasing paper publication count in recent years shows.

Only one study by Sommerauer and Müller [78] was found juxtaposing short-term and long-term effects of learning with AR. The 23 participants in the study by Sommerauer and Müller [78], being 22 pupils and their mathematics teacher, was tested for short-term and long-term effects. Tested content was not in the medical nor anatomical field but the mathematics field during an exposition. On 12 exhibits, marker-based AR was installed. Only a significant effect of learning with AR on short-term but not on long-term could be shown. However, Sommerauer and Müller [78] did not treat or show their AR installations.

A bias of the AR design and its installation have to be taken into account when referring to Sommerauer and Müller [78], especially, because the design of AR influenced the medium-term memorization in another study [79].

Learning effects of a mobile marker-based AR learning environment was compared with learning effects of a mobile learning environment and the former let participants achieve 8.1 % less errors after a medium-term test [79]. The short-term and medium-term learning effects with the two learning environments was not compared to traditional learning [79], restricting the results to marker-based AR compared with traditional learning. Sommerauer and Müller [78] who investigated short-term and long-term learning effects state that using a whole class as participants and that the long-term test was part of their lessons may have contributed to the final outcome, a positive short-term learning effect, too. As the proven strengths of learning with AR are an increment of spatial abilities and positive short-term effects, a study investigating short-term, medium-term, and long-term with spatial AR learning content, as the developed application *ARnatomy*, may deliver insights medium-term or long-term learning effects.

In medicine the 3D understanding of spatial anatomy and its relationships is important. Text provides complementary information wherein complex processes are presented textually, their attributes and structures visually as well as spatially [33]. A combination of 3D models with correct textual information should to use the strengths of textual and model information. 3D rendering offers more exploration possibilities, since its 3D dimensions are not projected into the 2D space as illustrations in books. The remaining of three-dimensionality eliminates the cognitive challenge of re-projecting 2D illustrations in books into the 3D space within the students minds [16]. Rotation and manipulation of 3D anatomy from different angles and views helps to get faster and easier insight and understanding of anatomy. Research results led to the recommendation that linkage to definitions and details should be dynamic, secondary media as labels and annotations, in applications containing anatomical models [33]. Important are highly crosslinked and hyperlinked definitions, labels, and annotations for a well established interconnection of information between all anatomy parts.

Creating a link between text and models is crucial to have one view only with translations of disappearing and appearing content to not break concentration and attention. The aim of developers should be to keep cognition in the created mental picture and trying to create one mental picture with texts, models or other modalities instead of creating multiple separate mental pictures referring to each other. The previous identified benefits may not apply to each application as the application's interaction, User Interface (UI), and information rendering can be different [33]. Careful development, knowledge about AR intrinsics, their effects on users, Software Development Kit (SDK)s, and the SDKs capabilities are highly important to profit from the learning effects from AR as much as possible.

2.2 Anatomy Applications without Augmented Reality

Leading mobile applications rendering human anatomy are from the vendors *Visible Body* [80], *3D4medical* [81], and *PocketAnatomy* [82]. The aforementioned vendors so not have any scientific publications discussing their data, design, interaction, and evaluation [13]. The websites of the aforementioned vendors and their videos show detailed surface models, anatomical definitions with a full range of 2D and 3D touch interaction, no AR, and the intention to educate the general public. A free online visualization of 2D and 3D human anatomy imaging data targeted at medical students and experts is Online Anatomical Human (OAH) [31], which can be accessed through a web browser. The OAH enables interactive exploration for the user, adding landmarks, region, line annotations, shareable hyperlinks, and leap motion, which enables the control of the visualization entirely by hand gestures. Sliders allow users to dissect the models simultaneously on all 3D axes and show the model's Computed Tomography (CT) images, and Magnetic Resonance Imaging (MRI) in the dissected area.

Smit et al. [31] state that the number of specialists except radiologists viewing CT images, and MRI rises. Hence, it is fruitful to give everyone including medical students and doctors the ability to learn mental reconstruction and association of 3D anatomy with 2D images. OAH provides online courses for learners and the landmarks and hyperlinks are intended as teaching and communication methods, but lack the support of AR for embodied cognition. The Open Anatomy Browser (OAB) is an open source, web-based, volumetric imaging 3D model visualization with controllable clinical MRI intersections on each of the 3D axis. Geometric models, textual metadata, textual hierarchy of anatomical structures, sharable hyperlinks, dynamic shared views for multiple users, and collaborative interactive sessions are included [32].

A handful of locally developed human anatomy regions is viewable and publicly available in atlases. But the 2D MRI intersections and no textual definitions of the selected or viewed anatomical structures limit the audience to personnel educated in the radiographic field. AR is unsupported. The visualizations of OAB and their controls are not optimized for mobile devices. Inspired from *Wikipedia*, OAB users can create the information audited by consistency checks and supported by version history. Halle et al. [32] do not give the description, if only scientific users with medical background are allowed to add anatomical content or not. An application to train liver surgeons in computer-based planning is the *LiverSurgeryTrainer* [83], which supports decision making in preoperative learning. Radiological visualizations are paired with volumetric information, an interactive 3D liver model, and video material from actual surgeries to provide a learning platform for surgery and resection planning. The training steps form the workflow. First, a case selection from 13 predefined cases categorized by diagnosis, difficulty, and therapy. Second, the diagnosis going through anamnesis, examinations, and medical imaging. Third, the resection planning consists of vessel anatomy judgement, virtual resection line definition, and is depicted in Figure 2.11. Fourth, the analysis encompasses expert recommendations

and real surgery.

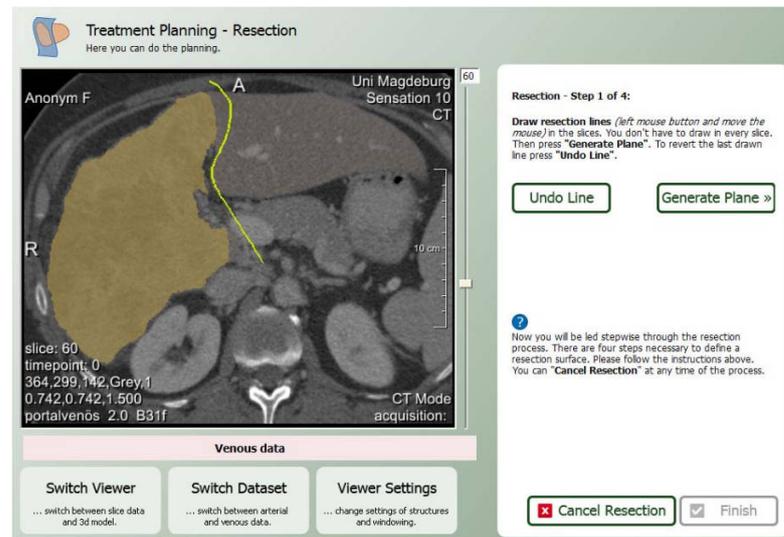


Figure 2.11: The yellow line represents the resection line and for novices the tumor is highlighted [83].

All cases and information was validated in terms of correctness and relevance by experts to improve the quality. As shown in Figure 2.11, radiological information is depicted in 2D with textual cues what the students should do. A view change in the planning step from the 2D radiological image to the 3D model depicting each time the currently drawn interactive resection lines. The resection surface and volumes are calculated automatically and presented. The resection date is important to give feedback if the remnant volume size is correct, vessel cuts or safety margins are violated. The planning result is then finally represented in the 3D model. Students can directly compare their resection results in terms of 2D, 3D models, *anatomical variants*, *number of resection surfaces*, *volume of remnant and resection liver parenchyma* [83] with the expert ones. As multiple solutions exist, no direct assessment of the students input is given.

Similar work without VR is conducted by Saalfeld et al. [84], where realistic 3D vessels or their 2D representations can be sketched with bloodflow simulations as well as operational access paths of vessels. Application fields are treatment planning, patient education or consultation, and general fast treatment option visualizations in 3D. Each single vessel is constructed by the program around a hand drawn, editable, smoothed center line. The smoothing is necessary, since the stylus used on a *zSpace 100* handheld can have tracking jitter causing the sketched lines to be irregularly. The handheld is capable of rendering stereopsis at a rate of 120 Hertz with additional glasses needed for the users. The stereopsis limits the usage of the application only to rather rare stereopsis handhelds and not to the majority of handhelds as pads, mobile devices or even computers. Vessel regions can be highlighted or cut for better visualization and annotated by 3D labels or

notes hovering in the air or directly on the vessel surface. In computer graphics, metaballs are organic-looking 3D-dimensional objects. The metaballs smoothly blend into each other and allow users the restructuring of the vessel surface and to create branches as depicted in Figure 2.12.

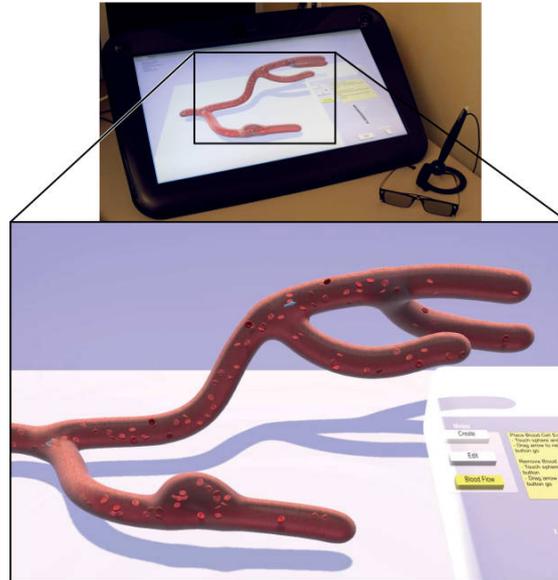


Figure 2.12: A sketched blood vessel with its metaballs and the handheld [84].

Real patient data can be imported and is automatically drawn to enable fast real data exploration, usage, and amendment [84]. The bloodflow adapts according to the vessel structure and shows the bloodflow impact on the vessel structure and especially on vessel diseases. The bloodflow simulation is a *realtime grid-based (Eulerian) Navier-Stokes simulation on the GPU* [84] along the center line with fast surface collision detection. The bloodflow simulation is considered to be sufficient to visualize the repercussions of aneurysms and stenoses.

2.3 Anatomy Applications with Augmented Reality

AR has many application cases in healthcare, e.g. showing nearby defibrillators, surgeon assistance, sport applications, education, health report to or by patients, and anatomy illustration during consultation [1]. Current healthcare applications are treated with marker-based AR in Subsection 2.3.1 and with markerless AR in Subsection 2.3.2.

2.3.1 Marker-Based Approaches

Including anatomy visualizations within realtime video streams creates anatomical AR. The anatomy models are mapped to key points a camera-captured human shape. Navab et al. [29] use the depth camera of *Microsoft Kinect* for positioning models of photographic

and CT datasets volumetric rendered as superimposition on the user body, which suggests the illusion to look into the own body. Target groups are teaching personnel and medical students, but not the general public. The application from Navab et al. [29] cannot run on mobile devices due to its usage of *Microsoft Kinect*. Textual information and additional 3D models can be displayed for more information transport.

Interaction methods and gestures are implemented to select sagittal layers of the rendered model volume. A user has to move his hand from left to right to perform a sagittal slice, front or back to select coronal layers, and up or down to switch between the sagittal and coronal layer modes. Navab et al. [29] state that they use a frosted glass metaphor, where the screen is viewed as a frosted glass. All objects on the glass are clearly visible and all others are blurred relatively to their distance to the glass. In the context of illustrative rendering, frosted glass belongs to high level illustrative rendering [24] with a novel interaction method. Low level illustrative rendering changes the appearance of objects to back- or foreground specific information, e.g. hatching generations, silhouette, crease line, or colour changes. High level illustrative rendering changes the visibility of objects to back- or foreground specific information, e.g. adding, removing, clipping or making parts transparent [85, 86]. Combinations of low level and high level illustrative renderings are possible. Missing functionality in Navab et al. [29] is deformation relative to user movement, textual definitions, and public access.

Similar to the *mirracle* project CT, the Living Book of Anatomy (LBA) [74] captures limb movement, speech or orofacial movement with Microsoft Kinect (MK) and visualizes information about involved muscles superimposed onto the user image shown on a display. The difference of LBA [74] to the *mirracle* project [29] are the intended target groups, namely the general public and medicine students, but both target groups need at least the MK camera. The MK limits the application area mostly to the home or the university, where a MK camera can be set up. A calibration to generate user specific bounding shapes of anatomy is needed for the LBA [74]. *Linear blend skinning* transforms vertices within a mesh by a *blend* of multiple transformations determined by bones inside the mesh. The bones are restricted in their range of movement and rotation and the mesh vertices move only when the corresponding bone moves. Further deformation of anatomy models in LBA [74] is applied via *linear blend skinning* [74] and enables realtime anatomy animation and mesh deformation with user movements. The 23 bones of the MK skeleton are used for mesh deformation on each frame. Different modes of anatomical superimposition with the MK camera are depicted in Figure 2.13. The usage of a screen during an operation is a dilemma. A near display in the region of the patient needs to be sterilized, which has the hazard of patient pollution. A distant display increases cognitive load due to looking forth and back causing eye adjustments [87]. Improvement of this situation is AR, where datasets, PET or CT images from the patient are fused with the operative area on a screen or on the patient himself. Common is the patient data projection without any orientation or guidance of virtual objects. More accuracy is achieved by a projection with



Figure 2.13: a) Skeletal model used b) Knee flexion muscle activity c,d) Skeleton superimposition e) Multiple organ superimposition f) Single selected organ superimposition [74].

previously set up markers. During surgeries the registration forms a reoccurring problem, since cameras need to be calibrated in case of patient movements or new patient data. In Figure 2.14, an image overlay on the patient's skin is depicted.

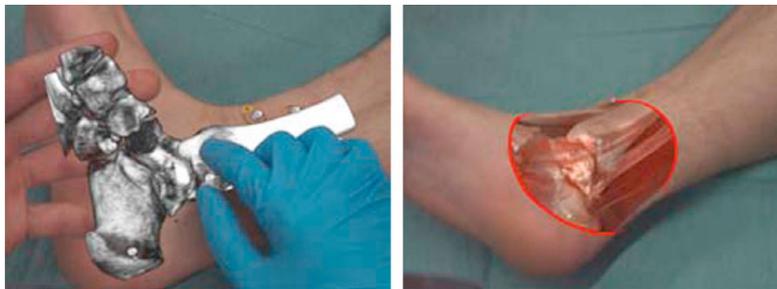


Figure 2.14: Left: Ankle with the drill entering as small yellow dot is visualized. Right: Focus and context rendering with shading as an additional depth cue of the ankle [87].

AR registration on regions with more entropy or large skin areas is easier than a registration on small skin areas, because more characteristics and features can be used for the registration. Small skin areas are frequently encountered in surgeries forming a problem for precise AR registration. The most used technology is the tracking of artificial markers fit to the digital renderings to minimize errors and better adoption of AR. Disadvantages of marker-based AR are limitations in interaction, tracking of markers during movement, deformation, scaling, misalignments, inaccuracy threshold, time, and the visual as well as the computational control [87, 2]. But new, markerless approaches of AR the medical field are being developed continuously, which have improved capabilities compared to marker-based AR. The following Subsection 2.3.2 treats five medical markerless AR state of the art applications.

2.3.2 Markerless Approaches

With the marker-based approach, fiducial markers need to be positioned on a patient or on a Region of Interest (RI) for registration. Markers in a big number can be expensive,

can distract the medical staff during their work, need to be manufactured, and positioned in the first place before the application can register them. Markerless AR is less invasive than marker-based AR [88]. Especially tracked animals can be distracted by markers and biasing their performance, medical treatments, and the final results. Alternatives to marker-based AR are skeleton fitting or regression training based on derived features of the objects to be registered. Fitting skeletons or active contour models have a high accuracy and performance but have a high development complexity, which limits their scope. Regression training using computational generated features achieve the best performance in pose estimation for humans. A special case in human pose estimation is HPE, which is used in this master thesis and treated in Section 2.5.2.

The delivered landmarks from the pose estimation are used for markerless AR positioning. The training and benchmark datasets are large and can have more than 1.000 entries, which can result in huge feature extraction files and high computational demand. Mathis et al. [88] utilized the state of the art transfer learning with the feature detector of the *DeeperCut* pose estimation algorithm [88] to achieve outstanding results on human limb detection with only approximately 200 training images. A Residual Neural Network (RNN) builds on pyramidal cells similar to the ones in the cerebral cortex. RNN utilize skip connections or shortcuts to optionally jump over one or more pyramidal cells. The feature detectors used in transfer learning are consist of enormous deep Residual Neural Network (RNN)s pretrained with *ImageNet* [88] containing more than 14 million images.

The training with few hundred frames results in specialized, stable feature detectors able to recognize and track anatomy successfully. The tests included changing illumination, dark transparent side walls, shadows, and distortions from a wide-angle lens on video material of a mouse walking and sniffing along a scent path [88]. The training set consisted of 100 frames with one mouse visible and allowed the RNN to detect multiple mice and their single body parts successfully with a maximum feature point detection variation of five pixels. The recognized anatomy features were small, e.g. the mouse's snout was 30 pixels wide and the video width was 654 pixels. If the training dataset contains multiple mice with occlusions, a better tracking result in terms of accuracy is achieved as with only one mouse. RNNs trained to recognize more body parts outperform RNNs trained to recognize less body parts with nearly half of the precision in pixels. In Figure 2.15, the detected mouse body orientation is visualized via lines and determined by its snout, ears, and tail.

Applying the achieved results and knowledge of transfer learning in the medical field, one question arises: How small can markers became and their placement areas be? A video can be zoomed or captured with special lenses to show small content zoomed on a screen. On zoomed visual information, markerless AR can do its enhancements without any camera recalibrations, if the patient moves or new data is available as opposed to marker-based AR. Mathis et al. [88] performed an exemplary test by tracking all fingers of a mouse within a monocular video stream. The constraints of transfer learning lay beyond

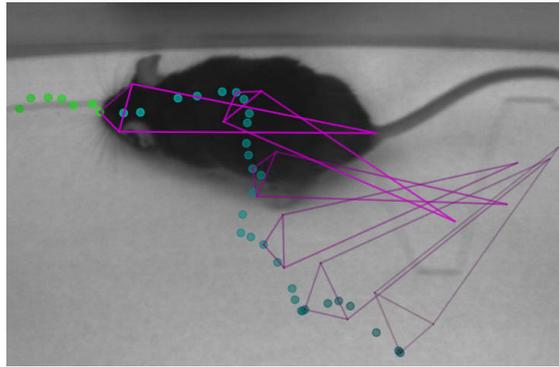


Figure 2.15: The green points show future snout positions and the cyan points past snout positions. The magenta coloured lines show the body orientation determined by four vertices [88].

the executed laboratory tests of Mathis et al. [88] due to the outstanding performance in a challenging environment having strong varying background and reflective walls. Many use cases in the medical environment, surgeries or medical video data lack varying background or reflective walls. So, the expectations on tracking speed and tracking accuracy from Mathis et al. [88] are very high as the lack of deflections should cause a small tracking speed or tracking accuracy increase.

On-patient data visualization in an AR environment can be made more robust via an improved registration method. Macedo and Apolinário Jr. [89] focus on the 3D superimpositions of medical 2D pictures or data via Volume Clipping (VC) . Continuous conflated regions or whole images on the z-axis can create a volumetric 3D data and models from 2D CT images or 2D MRI. Direct Volume Rendering (DVR) gets a 3D representation of the volume data directly. The data is considered to represent a semi-transparent light-emitting medium, which is used slice by slice for 3D volume generation. The algorithm used is known as front-to-back Direct Volume Rendering (DVR) , a single raycast through the already clipped bounding box of the 3D model is performed on a single rendering pass to select the correct 2D CT image or 2D MRI. The 2D image is blended with the Red Green Blue (RGB) data coming from the video stream for a realistic appearance, and occlusions are highly important too for realism. If a single raycast is on the outside the clipped volume the ray is discarded and the video stream pixel occludes the image pixel. Otherwise, the raycast travels through the volume and the image pixel occludes the video pixel.

Due to different greyscale values of bones and soft tissue, each single one or both can be identified and rendered differently in VC. Bone can be rendered more realistically and soft tissue can be blended with the background for bone highlighting or vice versa. Focus and Context Visualization Paradigm (FCVP) shows the medical data on the patient's body as

focus region, provides a better perception and immersion of medical data for the viewer as superimposition than without FCVP. The treated procedure is named Focus and Context Visualization Paradigm (FCVP) and shows the medical data on the patient's body as focus region, which is superior to superimposition [89]. With FCVP, the visual perception of the physician is improved. The lack of additional markers or other tracking hardware is beneficial for a quick usage by just starting the application. But accuracy remains a major problem in markerless AR for medical applications. In educational applications lower accuracy is sufficient. It is only a matter of time until the accuracy is high enough for medical usage, since works as the aforementioned Mathis et al. [88] improve accuracy continuously.

With FCVP and the 3D depth information from the video stream captured by MK, not only the volumetric model created by 2D image data but also the video stream can be clipped to see the background behind a user. The background was saved beforehand. A ray traverses on the z axis. If the ray is in the volumetric model region, the beforehand saved background is not rendered but the anatomy of the user. The medical image is clipped too and its foreground is blended with the video background for more realism as depicted in Figure 2.16.



Figure 2.16: Left: FCVP with smooth contours on the user. Middle: Visible background with a superimposed CT image. Right: Volume clipping of the user by MRI data [89]

The depth data from the MK camera with its *KinectFusion* algorithm are both processed with a GPU eliminating their deployment on mobile devices. The high computational demand of VC, rendering CT images, and rendering MRI makes the solution [89] inapplicable for the general public. Solely relying on depth data from any depth camera would extend the application support to mobile devices, which have a depth camera included. The numbers of mobile devices having a built-in depth camera is still low. The majority of the general public carries mobile devices with monocular cameras and this master thesis aims exactly for this majority as target group. Another limitation, besides the need for an GPU and a mobile depth camera, may be the VC, the rendering of CT images, the rendering of MRI, and not with 3D anatomy models synchronized with a face mesh as this master thesis does. Because, CT images and MRI are inappropriate for the education of the general public as laymen lack the medical skills to read radiographic or cross-sectional images. An installed MK system with VC software is applicable in public places or medical buildings. Stationary computers have a GPU and are capable of processing the accruing depth data. Incorporating the current state of the art in HPE,

treated in Section 2.5.2, could make the application more accurate due to more precise landmarks and eradicate the need for depth information. The HPE is solely performed on monocular video data. Most hospitals contain a surgery room performing operations and research strives to develop better operating techniques. Orthopedic trauma surgery focuses on minimally invasive operations, which require multiple 2D fluoroscopy intra-operative X-ray images to understand the 3D spatial relations in the operating area. Difficult mental mapping and extensive training is required to perform effectively and even experienced surgeons report *longer procedure times, increased radiation exposure, multiple tool insertions, and surgeon frustration* [90].

Amirkhanov et al. [30] developed *WithTeeth*, an application capable of previewing dental treatment by rendering different teeth models in realtime with AR utilizing the virtual mirror approach. The purpose is early feedback for the patient to raise endorsement as well as trust in the treatment itself and in its outcome. Supported input formats are pictures and videos. Teeth are sessile in the maxilla and mandible and have many properties, pose, meaning positing and orientation, form, colour, and texture. The identified and manipulated properties are colour and shape. Patients can change teeth models and colour until they are satisfied, the result may be communicated to the dental technicians lowering traveling, treatment, time, and costs. After an adjustment by the technician the final feedback is presented to the patient's device, who then can decide, if he undergoes a treatment or not.

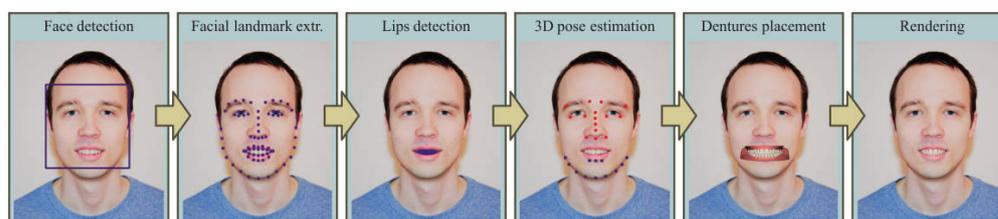


Figure 2.17: The processing pipeline of *WithTeeth* [30]

The algorithm depicted in Figure 2.17 consists of several steps and is applied sequentially in realtime on each frame [30]. Step one is face detection, step two is landmark extraction, step three is inner-lip landmark determination, step four is head and model pose estimation, step five is denture positioning and orientation, and step six is rendering of the teeth model. The foreground plane is the picture showing the face with a transparent inner-lip area, followed by the virtual teeth model as middle plane, and the original picture as background plane. The threefold plane rendering technique makes the tongue visible behind the rendered dental models in the final image. Single tooth superimposition is currently unsupported and may be implemented by a visibility mask and prior knowledge of the patient's teeth layout. The tests showed zooming is advantageous to see dental details, but currently unsupported and may be achieved with adding more facial landmarks around the mouth. Missing real teeth are unsupported, if a patient misses a teeth the skin above may be concave, which could be counterbalanced with skin

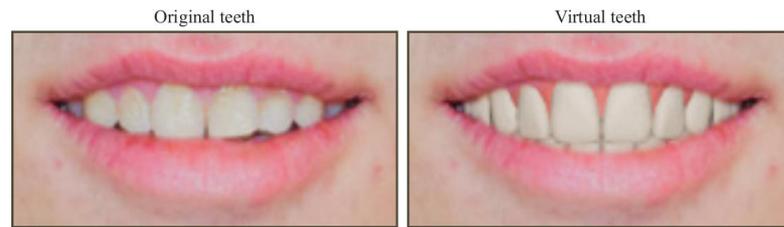


Figure 2.18: Final result of *Withteeth*, zoomed in on teeth superimposition by virtual denture [30].

deformation. The usage of *OpenCV* [91], *dlib* [92], and the *Unreal Engine* [93] makes *WithTeeth* platform agnostic and unavailable for the education of the general public. As a minimum for the education of the general public, adding dental information and a FE, on a mobile application for example, is required. Then, interaction with the teeth and the education about them would become possible. If the interaction of single teeth would be supported too, information for education on prophylaxis, dental bridge, repercussions of diastema, implants, dental prosthesis ecetera could be added.

Hajek et al. [90] developed to their knowledge the first markerless AR solution for intra-operative target anatomy designed for surgeons and not the general public. The application is an *in-situ* tracking based on 2D X-ray images, rendered as AR in an optical see-through Head-Mounted Display (HMD) to superimpose the anatomy of interest. A C-Arm (CA) intensifies scanned images, mostly fluoroscopic ones, primarily during surgical or orthopedic procedures. Figuratively depicted, the patient or object to be scanned lies inside the recess of the letter *C*, which is the name origin of the C-Arm (CA). Operation observers, residents, and personnel moving the CA delivering 2D X-ray images can understand the actions of the leading surgeon better through marking the *in-vivo* anatomy on the HMD.

The coordinate system for the AR visualizations are obtained from V-SLAM algorithm receiving additional depth information from a *Microsoft HoloLens* [94] as tracker facing away from the patient and additionally mounted on the CA. In a standard CA position and both poses of the tracker and X-ray cameras are known. The tracker and the X-ray camera are offline calibrated once by hand and then the data from both systems are joined to have a translation matrix between V-SLAM coordinates as well as the 3D volume generated by 2D X-ray images. With the known data from the tracker and the X-ray camera, AR becomes possible and is visualized in the form of volumetric superimposition of real anatomy with the 3D volume. User defined landmarks and annotations as well as lines indicating the angle and direction of the X-ray who generated the images can be rendered additionally. In Figure 2.19, a radiopaque tubular structure with a radius of five millimeters serving as phantom anatomy is depicted as well as the *in-vivo* anatomy rendering on the outside of the foam tissue.

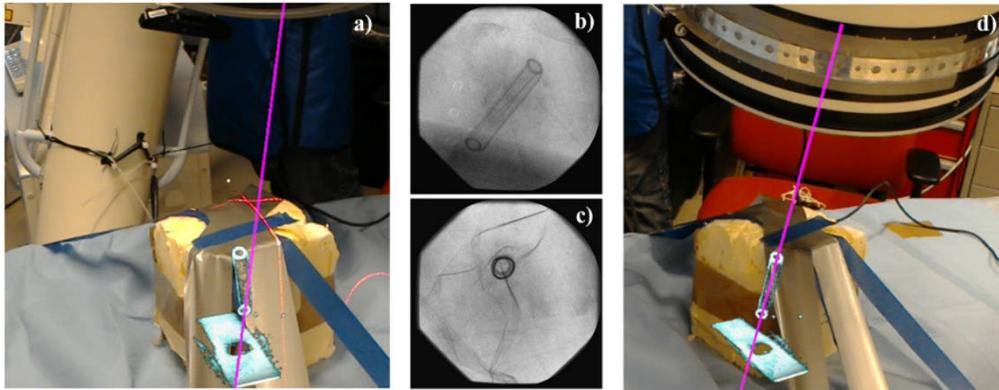


Figure 2.19: Image a) Operator perspective seeing the AR with an incorrect CA alignment. Image b) Side X-ray image of alignment in a). Image c) Top X-ray image of alignment in a). Image d) Operator perspective seeing the AR with a correct CA alignment [90]

Hajek et al. [95] state that other approaches using preprocessed models of scenes would be affected through CA movement, especially when the CA changes its orientation. The difference from the Hajek et al. [95] approach to others is that an a preprocessed model of the room with only the operation table inside is used instead of the fully equipped room. Tools or people are just considered as occlusions or surfaces. With the RGBD configuration, the pose of the device rendering AR is known in the coordinate system and a radiation visualization can be calculated on each new frame [95]. An improvement would be to incorporate monocular ORB-SLAM 2 [64] to increase accuracy, robustness, and to remove therefore involved RGBD sensors making the application available to far more devices and personnel. As further improvement, the AR radiation visualization of Padoy et al. [95] should be fused with the AR 2D X-ray image superimposition on the anatomy of interest by Hajek et al. [90]. Users would profit of all benefits from both applications, additional radiation awareness, X-ray image superimposition, and correct CA image angle visualization.

An Electromagnetic Field (EMF) is a physical field produced by electrically charged objects. With the visualization of invisible entities, the education about invisible entities become possible, e.g. any physical fields as Electromagnetic Field (EMF) s along with their properties could be visualized. a user may select the location of his Wireless Local Area Network (WLAN) router via his mobile device and then the mobile device visualizes the WLAN as AR EMF with the field visualization colour depending on the WLAN signal strength. Other field visualizations, as Bluetooth or atomic radiation with a “*Geiger-Müller Tube*” [96] for example, should work too.

2.4 Model Datasets

The quality of perceived AR can only be as high as the used models. Leading model vendors selling access to their artistically, medical, scientific inaccurate datasets [13] for visualization are *Digimation* [97], including data from the free Visible Human Project (VHP) [98], and *TurboSquid* [99]. Digimation provides nearly 600 and TurboSquid delivers nearly 10.000 medical models about the human body and both are intended for the general public. In contrast to the former two, VHP consists of scientific accurate *cross-sectional cryosection, CT images, and MRI obtained from one female cadaver and one male cadaver*. The CT images and MRI are captured in 0.33 mm intervals for the female and in 4 mm intervals for the male one. VHP can be accessed by the general public and its data is mainly intended for science. Artistically created and scientific accurate anatomical human datasets are *BioDigital* [100], *ZygoteBody* [15], and *Body Parts 3D (BP3D)* [4]. BioDigital is an online cloud service including visualizations of anatomy, diseases, treatment, and interactive 3D models. The data from *BioDigital* is accessible through payment via an Application Programming Interface (API). Own applications can be build with the data from *BioDigital* for the general public, for medical students, and medical personnel due to the high scientific detail of the models.

ZygoteBody is a 3D interaction, web-based human anatomy online visualization with restricted possibilities and without AR. Features as high resolution 3D anatomy, additional content, advanced navigation, notes, exact viewing angle, dissections, adding anatomy to a view, removing anatomy from a view, and saving everything as a state becomes available with a paid premium account. *ZygoteBody* is intended as teaching and learning support of human anatomy online and provides no full digital access to the DBs. The target group are medical students as well as medical personnel, but everyone who pays can learn. BP3D is a flat file collection, which provides in its current version number three 1523 single, free, anatomical, 3D objects of a human male. From all mentioned DBs or flat file collections, BP3D is the only free scientific one for the general public and science. A free viewing tool named *Anatomography* is available for screenshot creation with the desired angle, zoom, colour, opacity, and *Anatomography* lacks definitions, mobile design, as well as AR.

2.5 Face Tracking

Rendering anatomical AR content as superimposition in front of a face in a video stream requires the facial recognition and tracking beforehand. Landmarks of a detected face must be tracked frame by frame for a correct anatomical model rendering as well as the current head rotation in 3D space. Several algorithms for face detection and HPE exist. Subsection 2.5.1 treats the state of the art in face recognition and Subsection 2.5.2 treats the state of the art in HPE.

2.5.1 Face Recognition

An public available computer vision library named *OpenCV* [91] is widely used in practice for many years. *OpenCV* offers the *Eigenface*, *FisherFace*, and *LBPFace* algorithms for face detection. All three algorithms rely on feature extraction and their classification against a training set with Nearest Neighbour Classification (NNC). The Sparse Representation Classification (SRC) is a face recognition classifier using pixels without feature extraction and the l1 optimization for less influence of illumination and occlusions [101]. Shen et al. [101] use Sparse Representation Classification (SRC) for face recognition and outperform the three algorithms in *OpenCV* in recognition accuracy and recognition speed. The advantage of SRC is its direct pixel usage without feature extraction, more robustness to illumination changes, and occlusions.

A pixel dimensional dictionary is build from column-vectorized images of the same size as the images and for each subject, a subdirectory is created. Classes are formed and a test image is found inside the dictionary by solving for unkown variables in a linear equation denoting the affiliation to a class. Ideally is a test image in a vector-spanned space of one corresponding class. If the spanned class number is large and the ideal condition holds, the test image denotes a sparse vector. Solving the aforementioned linear equations is computationally demanding due to the pixel dimensional dictionary. Random projection matrices can be applied to reduce needed computational resources and preserve accuracy. For an AR application, where oneself or other humans in motion are captured, pure face recognition is insufficient. No face landmarks are detected and relying only on recognized sparse face data is insufficient for model mapping.

Face recognition can be effective and robust against various noises or low resolution. Nugroho and Kusumoputro [102] achieve strong noise recognition rates above 95 % and 98 % on clear images. The fuzzy technique converts all images to their greyscales, apply Gaussian, Poisson, and salt-and-pepper noises, for each single face image of a person. A person must be depicted in each single pose to detect. Finally, all generated images of one person are grouped into an one set crisp-vector reducing 2D vectors into one-dimensional vectors. Triangular fuzzification with the lowest, mean, and highest greylevel intensity values, as depiced in Figure 2.20, of a pixel on all sets of face images are applied to transform their crisp vectors to fuzzy ones.

It seams the state of the art in face recognition requires predefined images of users [101, 102, 103]. No single work has mentioned extracting the needed images from the video stream of the front camera of a mobile platform, when the application or a task in an application needing face recognition is running. HPE can recognize faces and delivers face landmarks and their rotation in 3D space, which can be used for further processing, but incorporates a higher computational demand than face recognition.

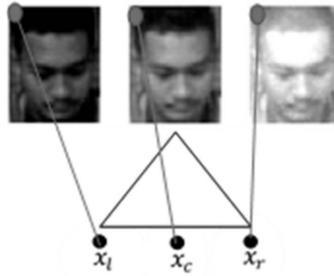


Figure 2.20: Triangular fuzzification with the lowest (x_l), mean (x_c), and highest (x_r) greylevel intensity values [102].

2.5.2 Head Pose Estimation

Inferring the head orientation via an image or video relative to a camera is HPE [104]. HPE can be divided in two categories, i.e. coarse and fine, where the data of both is detected in an unconstrained 2D or 3D space. The coarse HPE category detects the head pose directions roughly [105]. The fine HPE category detects the three angles, i.e. yaw, pitch, and roll, depicted in Figure 2.21, as well as the face orientation encompassing typically all or a subset of the landmark points aligned with eyebrows, eyes, lips or the jaw.

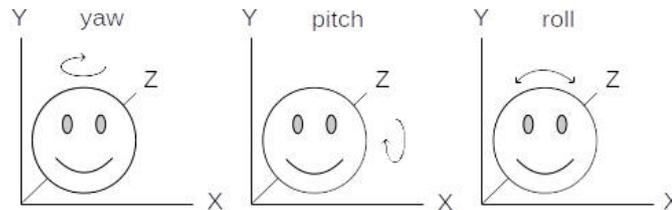


Figure 2.21: The 3D axes with their rotations shown on a sketch of a face.

Cascaded Convolutional Networks (CCNs) are a consecutive interconnection of single Convolutional Neural Networks (CNNs) without feedback. Thus, the adjective *cascaded* in the name of CCN. Multi-task Cascaded Convolutional Networks (CCN) are used in the field of inter-frame correlations and by Florea et al. [104] to perform fast face detection and HPE via deep trained CCNs. The CCNs use the link between detection, alignment, and vicinity of facial landmarks for performance gains. Convolutional Neural Networks (CNNs) are a class of deep neural networks, wherein neuron connectivity patterns resemble the organization of an animal visual cortex. CNNs are designed to require minimal preprocessing. Three stages form the multi-task deep Convolutional Neuronal Networks (CNN)s for face detection. An image pyramid with different sizes of the target image is created, which is not labeled as step by Florea et al. [104]. For completeness, the image pyramid generation, required in step one, and the pretraining of a deep neuronal network with face images, required in step four, should be declared as initial step zero. A *proposal network* generates detection proposals, which can be

discarded or processed, for faster overall detection of a CCN. Non-Maximum Suppression (NMS) is an edge thinning technique applied to find *the largest edge*. Step one determines the face Bounding Boxes (BBs) with a very shallow and fast proposal network and a Non-Maximum Suppression (NMS) merges superimposing BBs. Step two performs BB deletion through a refine network. Step three detects the final BB with five face landmarks with an output network. Step four feeds BB data into the pretrained deep neuronal network for face recognition. All steps combined form the HPE method by Florea et al. [104].

Pitch, roll, and yaw prediction is conducted via scaling of the last fully connected network layer that initially has a dimension of 1000. Only the luminance of the single frames is used. The already found BBs are fed into a classic machine learning multi-layer perceptron, and two deep learning neuronal networks, i.e. *VGG16* [106] and *ResNet50* [107], detect the face. A luminance image contains only brightness information without any colour information. The two neuronal networks performed better than the multi-layer perceptron, but only with fine tuned data and the usage of consecutive luminance frames. Face landmarks moving a lot, as the mouth for instance, increase error rate as well as yaw rotations. A yaw rotation makes the two irises appear to be asymmetrical inside the face BB, which causes false predictions [104]. Adding nose information as countermeasure in the training process decreased false predictions. Neuronal networks can outperform classic machine learning algorithms with a nine times lower yaw error rate and a four times lower pitch error rate.

A new state of the art description in face detection and HPE is achieved by Wu et al. [108] with CCNs. Better performance than other works comes from the focus of Wu et al. [108] on extended feature fusion in the multi-task learning process and training process. Shallow layers used in the multi-task CCNs contain features more suitable for localization, which respond to corners as well as edges and serve as important structures for HPE. Deep layers used in the multi-task CCNs contain features more suitable for learning, i.e. any color information, properties, landmarks or landmark regions. The baseline framework in the multi-task CCNs is adapted accordingly to the shallow layers and deep layers. Shallow layers, deep layers, and in-between layers are fused in the last CNN enabling it to benefit of features of the previous deep layers, which increases accuracy through the more superficial features of shallow layers. The face detection speed and accuracy gains nothing from feature fusion of shallow layers and deep layers, but the HPE speed and accuracy increases. The multi-task CCN from Wu et al. [108] is the new state of the art in HPE.

The algorithm structure of Wu et al. [108] is similar to the work of Florea et al. [104] and has notable different details in step two and three. Step two is an optimized CNN, its input is a BB coming from step one. By the calculation of the BB center, each BB content is cut out of the original BB as a square with a side length of 20 pixels. The output of step two is a vector containing a binary no face classification score, a binary face classification score, BB regression offsets, and the three pose estimation angles. A threshold on the binary no face classification score and binary face classification

score decides, which BBs is further processed by applying NMS and a calibration with regression vectors to receive a sharper image. Due to false positives, yaw, pitch, and roll angles are ignored in step two. Applying NMS in step two and not in step one is different from most other approaches in HPE, e.g. Florea et al. [104]. Step three is a fine detection and pose estimation CNN, another threshold filters the view remaining BBs coming from step two resulting in a minimized inclusion of false positives. The highly complex network processes the final BBs with high accuracy outputs of yaw, pitch, and roll angles determining the final HPE [108]. The subimages denoted by the BBs include always the RGB channels and not any extracted features, as luminance for example. The inclusion of all RGB channel information in all involved CNNs by Wu et al. [108] is a further difference other state of the art works. The training images are scaled to 10, 20, and 40 pixel side lengths of a square enabling the Wu et al. [108] approach to detect small faces in an image accurately as depicted in Figure 2.22.

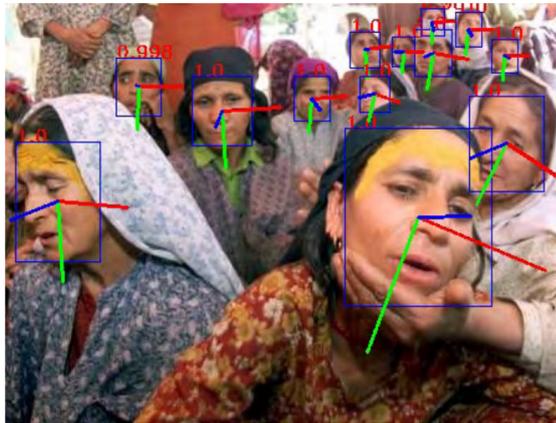


Figure 2.22: Multiple detected faces with their 3D rotational axes: x red, y green, and z blue [108].

In the *ARnatomy* application, the coordinates, face landmarks, and the yaw, pitch, and roll angles from HPE are used to position the anatomy models queried from the mobile DB.

2.6 Mobile Databases

Storing data between different program invocations is named *persistence* [109]. Rational DBs persist data as relations, i.e. a presentation in tabular form meaning a set of tables [109] consisting by rows and columns, and offer relational operators to users or clients to manipulate the persisted data. Applications having high data usage depend non-relational or relational DB as persistence optimization and speedup. The DBs are queried via raw DB operations or framework APIs providing DB operations as abstraction. Persistence influences not only stored data, but rather execution time, energy consumption, and programming complexity. Energy economics remains as one of the key considerations in

mobile application development, since application energy consumption surpasses battery capacities [110]. Research literature on efficient energy usage addresses programming patterns, modeling, component usage, best practices, lifecycle, general program analysis, and detailed program analysis but misses detailed persistence studies. Song et al. [110] tested eight DB frameworks designed for mobile devices with limited resources for their energy consumption, performance, and programming effort on an LG LS740 smartphone, with one Gigabyte (GB) of Random Access Memory (RAM), eight GB Read Only Memory (ROM), 131.2GHz quad-core Qualcomm Snapdragon 400 processor, and *Android* 4.4.2. The tested frameworks are *Android* Structured Query Language Lite (SQLite), *ActiveAndroid*, *greenDAO*, *OrmLite*, *Sugar ORM*, *DBFlow*, *Java Realm*, and *Paper*, using the SQLite DB engine as well as *Realm*, which is a *NoSQL* DB engine.

Energy consumption is calculated by the time current of the device battery. The *Monsoon Power Monitor* revealed the average current and the current during execution time, which is higher than the average current. The current during execution time is determined by the load put on the CPU, hard disk, and RAM. Performance is measured with the overall execution time of a whole DB transaction and the involved DB operations. The frameworks using a SQLite engine with the *Android* API measure only their CRUD operation times. Uncommented Lines of Code (ULOC) is a software metric used to measure the programming effort put into a computer program by its uncommented source code line count. The software metric Uncommented Lines of Code (ULOC) is applied to measure the programming effort of a computer program. Despite other existing measurements, ULOC is still frequently used in literature to measure programming effort.

An adopted version Java DB benchmark named *DaCapo H2* [111] is used for testing. The H2 DB engine is replaced with SQLite or *Realm* to be able to use the benchmark in the *Android* environment. *DaCapo* contains high amount of data, which change to simulate bank transactions involving data from a real production environment. *DaCapo* encompasses of 12 complex tables, 11 relationships, in total 120 columns, and complex DB operations being transactions, batching, and aggregations [110]. With the *DaCapo* benchmark the results are as follows: *ActiveAndroid* has the best performance and most programming effort both originating in its unexceptional use of the raw SQLite API and the least amount of DB operation invocations due to its caching mechanism.

DBFlow has the worse performance and less programming effort than *ActiveAndroid*, but more than *greenDAO*, and is considered by Song et al. [110] to have the best balance between performance and programming effort [110]. *greenDAO* has middle performance and the lowest programming effort. *Sugar ORM* and *Java Realm* had the worst performance. The same data inside a transaction can cause different CRUD and other DB operation invocations on different frameworks. Hence, the *Android Object Relational Mapping (ORM) benchmark* involves a more fine grained micro benchmark than the macro benchmark *DaCapo* on the low level CRUD operations. The CRUD operations are

additionally taken into account for the evaluation of execution time, energy consumption, and programming complexity. With focus on the low level CRUD DB operations, the results for the frameworks on the *Android ORM benchmark* are ordered descending by their best performance: *Android SQLite*, *greenDAO* and *DBFlow*, *OrmLite*, *Java Realm*, *Paper*, *Sugar ORM*, and *ActiveAndroid*. *Paper* and *DBFlow* have the lowest programming effort and the latter can be generally recommended having the best balance between performance and programming effort in relying only on CRUD operations. When the dominating operations are select or delete, *Sugar ORM* should be omitted, on insert operations *DBFlow*, and on update operations *Java Realm*. The evaluation of the ORMs has different results with different testing granularities.

For a more practical approach than testing just the CRUD times and counting the ULOC, a numerical model named Performance, Energy consumption, and Programming effort (PEP) is introduced [110]. PEP includes weights for the developer's preferences. The properties of each framework in the acronym PEP are all included in the numerical model weighted with two parameters denoting the preference performance or programming effort of a developer and the developer's expectation of the transaction volume in the DB. For *DaCapo* 40 and 1,500 and for *Android ORM benchmark* 1,025 and 20,025 transactions are executed. *DBFlow* has the best balance between performance and programming effort with high transaction volume on the *DaCapo* benchmark, *ActiveAndroid* the best performance with most effort in the PEP model. *Paper* has the best balance between performance and programming effort with a low transaction volume on the *DaCapo* benchmark. Above 10,000 transactions in *Paper*, the scalability becomes a problem and *Android SQLite* or *greenDAO* are recommended. *Paper* has the best balance between performance and programming effort with a low or high transaction volume on the *Android ORM benchmark*. Only on high transaction volumes, *Android SQLite* has the best performance with most effort in the PEP model.

2.7 Software Development Kits

As foundation for the development of an AR application, a software framework or a SDK is required. Developers need to know the capabilities and constraints of a framework to create the best application possible. In Table 2.1 short overview of currently popular and maintained AR frameworks with their most important features is given. All SDKs in Table 2.1 have a *Unity 3D* plugin except *AR.js*, which is developed only for the usage on the Internet. As treated in Section 2.1.2, V-SLAM with its derivations is the state of the art in markerless AR.

With time constraints, it is nearly impossible for a developer to quickly get an overview of the SDKs, which use the most advanced SLAM or V-SLAM techniques. Basic AR functionality for creating simple applications are free of charge and more complex and efficient applications using V-SLAM or other key features force the developer into payment. Free SDKs with V-SLAM and markerless AR are only ARCore [22] and ARKit

<i>Name</i>	<i>Marker-Based</i>	<i>Markerless</i>	<i>Free</i>	<i>Payed Extension</i>	<i>Open Source</i>	<i>Unity</i>	<i>Cross Platform</i>	<i>SLAM</i>
ARCore [22]		x	x			x		x
AR.js [112]	x		x		x		x	
ARKit [113]	x	x				x		x
ARToolKit [114]	x	x	x		x	x	x	
EasyAR [115]	x	x	x	x		x	x	x
Kudan [116]		x	x			x	x	x
Pikkart [117]	x		x	x		x	x	
Vuforia [118]	x	x	x	x		x	x	
Wikitude [119]	x	x	x	x		x	x	x

Table 2.1: Overview of AR frameworks with their most important features [120, 121, 122, 123, 124, 125].

[113] from the two technology companies *Google* and *Apple*. The similarities of ARCore and *ARKit* are of the usage of monocular V-SLAM combined with device sensory data, ambient light estimation, vertical plane detection, and horizontal plane detection.

During research on which technology will be used in this master thesis, *Google's Android* documentation seemed to be more readable and details easier to find. Time constraints were a huge factor in the decision making process for this master thesis. The Java Native Interface (JNI) is a foreign function interface programming framework that enables Java code running in a Java virtual machine (JVM) to call and to be called by native applications and libraries written in other languages such as *C*, *C++*, and *assembly*. A Native Development Kit (NDK) is used to generate native code bibliographies from another programming language. ARCore [22] is programmed with Java and native *C++* code can be included too via Java Native Interface (JNI) or Native Development Kit (NDK). The documentation for the native inclusion of *C++* was found a lot faster for ARCore [22] than for *ARKit*. Additionally, I have already more than seven years experience with Java and *C++*. *ARKit* can be programmed in *Apple's* own language *swift* or *objective-c*, two programming languages I'm unfamiliar with.

The embedding of face recognition via *C++* libraries was cumbersome and costly in terms of time, even with profound knowledge of the involved technologies, programming languages, found documentation, and workarounds. ARCore [22] only partially supports marker-based AR by allowing only 2D template-based AR recognition, whereas *ARKit* supports full marker-based AR. Since only the markerless AR was considered to be

used in this master thesis, the marker-based capabilities of both SDKs were negligible. To use ARCore [22] one has to simply download sample files from the corresponding *GitHub* repository and the *Android Studio* Integrated Development Environment (IDE). To use *ARKit*, one has to register as *Apple* Developer to get an Identification (ID). As finding general information and finding the documentation for ARCore was faster than for *ARKit*, getting another ID and account was unwanted due to the overhead of registration, password creation, password storage, and already more required time for general search for *ARKit* than for ARCore.

Another reason for using *Android* were the lower prices of ARCore [22] supported mobile devices compared with *Apple* mobile devices and the higher prevalence rate of *Android*. For ARCore [22] usage a supported device was needed, and the cheapest one, being the *Nokia 7 Plus*, was chosen. The same would have been applied to the usage of *ARKit*, an *iPhone* supporting at least an *iOS* version 12. A private *Android* phone was already available before the ordered *Nokia 7 Plus* was received. The private *Android* phone enabled an early programming start of *ARnatomy* without ARCore [22] to save time. Later, when the *Nokia* phone arrived, *ARnatomy* was solely implemented on it. *Android* had a global market share of 87.3 % in 2018 [126]. A developer who wants to reach as many users as possible should target the majority first and then the biggest minorities. With all aforementioned considerations and experiences accumulated, time constraints, and the overall ease of use witnessed in the *Android* ecosystem, a decision for the usage of ARCore [22] was made.

Methodology

The methodological pipeline of this master thesis is depicted in Figure 3.1.

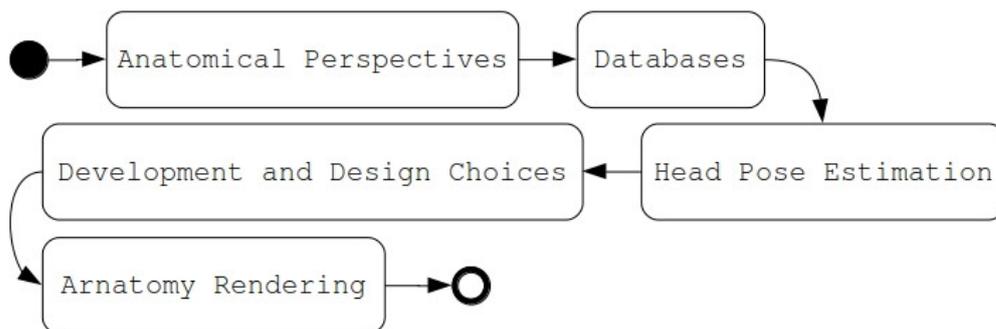


Figure 3.1: The methodological pipeline of this master thesis as Unified Modeling Language (UML) flow diagram.

Section 3.1 treats the familiarization with anatomy and Anatomy Perspectives (APs) on anatomy, whereupon all sequential decisions in this master thesis were made. Section 3.2 treats the investigation of the model Database (DB) adjudged for applicable in relation to the model data format and data structure. With a generated and optimized DB, the search of Head Pose Estimation (HPE) techniques and their implementation for minimal rendering of one single model with the Augmented Reality (AR) framework ARCore [22] was performed. Section 3.3 treats the technology investigations, installments, and changes for interoperability, which were executed for a realtime and robust HPE. Section 3.4 treats the development and design choices, which influenced the gained knowledge and experience about the identified technology to be used for the overall application with consideration of software design patterns, guidelines, and designs for the Back-End (BE) and Front-End (FE) . Section 3.5 treats the implementation and the different, nested

technologies, which required several decisions on structure, data flow, User Interface (UI), and visualization from the back- to the FE, based on the development and design.

3.1 Selection of APs

The study of organisms along with their parts is decidedly complex, has manifold interrelations, and various manifestations, which all precipitate multiple APs an observer can occupy. In this master thesis, solely physical or spatial anatomy is treated without its influences from other circumstances, illnesses or anatomical changes of any form or cause. Physical or spatial anatomy can be amended and newly categorized in non-spatial and spatial anatomy to gain more insight in doing so. Non-spatial anatomy knowledge includes terminology, taxonomy, and functions of structures. Spatial anatomy knowledge includes position, orientation, extent, and shape of structures. Additionally to non-spatial and spatial anatomy, Preim and Saalfeld [13] identified more differentiated AP:

- *Macroscopic*, wherein anatomy is observed with the unaided eye, essentially a visual observation during or after a dissection.
- *Microscopic*, wherein anatomy is observed with light or electron microscopes allowing spectators a subcellular observation.
- *Regional*, wherein anatomy parts are observed with emphasis on their regional and biological interrelations with other anatomical parts or systems, e.g. the head, thorax, blood circulation, blood-cerebral barrier, or the brain.
- *Systemic*, wherein anatomical systems are observed within the whole body, e.g. cardiovascular system, immune system or endocrine system.
- *Radiographic*, wherein anatomy is observed with different kinds of medical imaging as Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) for example.
- *Cross-Sectional*, wherein anatomy is observed in the transverse body plane, which is from left to right or right to left.
- *Surgical*, wherein anatomy is observed with focus on its relevance during an operation to fulfil set objectives.
- *Clinical*, wherein anatomy is observed in terms of practical, applied medicine for state of health changes.
- *Comparative*, wherein anatomy is observed in aspects of morphology, topology, orthology, and pathology in different organisms.

The requirements of an anatomical model DB change with the selected AP. The majority of the above listed APs on anatomy are for medical professionals, highly interested individuals without any medical profession or patients, and not for laymen of the general public. Microscopic anatomy would increase the time and programming effort tremendously, since subcellular AR renderings with zooming in and out for an illustration would be required. Open Graphics Library (OpenGL) is a cross-language, cross-platform Application Programming Interface (API) for rendering Two Dimensional (2D) and Three Dimensional (3D) vector graphics. Cellular AR rendering and subcellular AR rendering would require a development of an own particle Open Graphics Library (OpenGL) system for AR with interaction. Systemic anatomy would require a full body pose estimation algorithm with enough landmarks for correct synchronization to be able to visualize all the single systems of anatomy. If only HPE is executed for the systemic anatomy, only parts of anatomical systems would be visible or full anatomy systems would be excluded, e.g. the genitourinary system.

Radiographic and cross-sectional anatomy would require too much resources for a mobile device, as on the fly volumetric rendering from the 2D image data to 3D models is computationally demanding [86, 89]. Preprocessed 3D model generation from sequential 2D radiographic or cross-sectional images is possible. Preprocessed 3D models would lack the correct predefined position inside the human body, all generated models would have no relative coordinates to human anatomy. Each preprocessed 3D model would have to be positioned manually, which is a highly time expensive task. Surgical and clinical anatomy would constrain the target group to highly trained professionals already having a vast knowledge about anatomy due to their education and training. Surgical and clinical expert staff may have the need for applications facilitating their profession and not educating them in their profession. Comparative anatomy is the AP with the two highest requirements on a DB. First, a link between anatomy models of different organisms is needed and second, a decision, which from the all aforementioned APs are chosen for juxtaposition. All APs except the macroscopic and regional AP build upon the former two AP as required subset. So, all APs depend on the macroscopic AP and the regional AP and the latter two APs form the necessary starting point in the anatomical education of the general public. The macroscopic AP and the regional AP were selected for their implementation in the developed mobile application named *ARnatomy*.

The names, definitions, and a hierarchical structure available as models on a DB are required as an anatomical information minimum for all aforementioned APs. Only the free, scientific, public available, flat file model collection Body Parts 3D (BP3D) satisfies the requirements of the macroscopic and regional APs. Without the information of macroscopic and regional anatomy, all other APs lack context, relation, orientation, and observable content every human can make without any aid for the eye. For example, the macroscopic and regional perspective of vessels in the cerebral region is depicted in Figure 3.2.

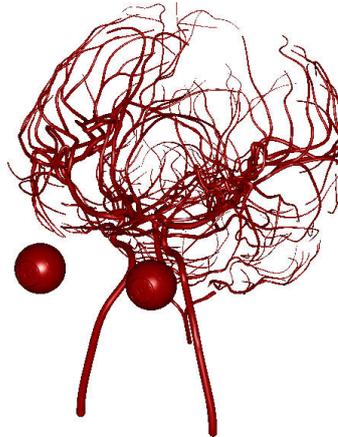


Figure 3.2: Macroscopic and regional perspective of vessels in the cerebral region with iris and choroid from the eyes as Point of Reference (PR) from the BP3D flat file collection [4].

Application extensions including any of the microscopic, systemic, radiographic, cross-sectional, surgical, clinical, and comparative APs may be made upon the foundational knowledge macroscopic and regional anatomy are constituting. As already treated in Section 2.4, from *Digimotion* [97], Visible Human Project (VHP) [98], *TurboSquid* [99], *BioDigital* [100], *ZygoteBody* [15], and BP3D [4] only the last one is free, scientific, and available for the general public and science. BP3D has additionally already anatomy model positioning in its own coordinate system for the human body. All BP3D models can be downloaded from a dedicated online preview, and all models of BP3D are linked to the Foundational Model of Anatomy (FMA) DB [5]. Only the macroscopic and regional APs are coherent with the models of the free, scientific, and public available BP3D flat file collection. Any other AP selection would have made the BP3D flat file collection insufficient and would require the usage of the scientific *BioDigital* [100] or *ZygoteBody* [15] DBs.

The models of BP3D receive through FMA definitions, synonyms and a full anatomy hierarchy of the human body. Macroscopic and regional APs along with the applicable DB create advantages as well as disadvantages for programming, learning, and interaction. As treated in Section 1.1, interactive 3D anatomy systems are equal to traditional learning and allow the learners faster buildup of spatial abilities through manipulation. The macroscopic and regional APs categorize anatomy, which can be represented through simple models. The macroscopic character of reducing the anatomy to its parts observable with the unaided eye is equivalent to reducing the complexity anatomy model arrangement to a minimum. Using rather simple models than sophisticated models imposes less cognitive load on the user and less foreknowledge is required for the learning of the rendered anatomy model.

Without additional APs, model interaction complexity and interaction paths remain simple and the user's attention and cognition can focus more on the anatomy macroscopic and regional exploration. In a highly detailed and complex anatomy visualization, users would require more cognitive capacities for orientation and interaction. If any interested user who wants to know more or specific aspects of anatomy should be supported in the future, other resources would be needed to be utilized as references to anatomical lexicons. The naturally restricted number of models through macroscopic and regional APs has a positive influence on the computational demand, overall loading time, and simplicity of the rendering technique. Other constitutive APs on anatomy have a negative influence on performance, as the model number along with their details, hierarchy depth, and rendering specialization would exacerbate the issue of limited resources on mobile devices. With more supported APs, a higher demand of programming skills is required to create an application with no visible latency. Additionally to programming skills, more needed time for programming is the result of supporting more APs, as more dependencies, required by the additional models, have to be considered with more potential error sources. An introduction of a new anatomical perspective in a mobile application tends to make software structure, its internal relations, malfunctions, and planning far more complex. Hence, only the macroscopic and the regional APs, primarily imposed by the two identified DBs, are in use.

3.2 Selection of Databases

As BP3D is the only flat file collection found to fit the requirements imposed by the macroscopic and regional APs, the search for the most efficient format and structure for mobile environments came to the conclusion that *Android SQLite* is the best choice. As empirically tested by Song et al. [110] and already treated in Section 2.6, *Android SQLite* has the best performance when focused on low level Create, Read, Update and Delete (CRUD) DB operations. The programming effort of *Android SQLite* is located in the higher field of the average Object Relational Mapping (ORM)s tested by Song et al. [110]. Only the read operation of the CRUD are needed in 99 % of the application during runtime, reducing the programming effort solely to few, simple, direct, and minimal queries for numbers, strings, and models. The remaining 1 % is the creating DB operation of writing already chosen anatomy indices in a dedicated table with one column named *recents* to have a recent anatomy list. Despite the higher effort Structured Query Language Lite (SQLite) has compared to other persistence frameworks [110], its integration in *ARnatomy* was chosen due to the following advantages.

Android supports SQLite, since API version one and it works out of the box, no additional dependencies are needed, which keeps the application size and the computational demand low. An official *Android* documentation of its API to SQLite exists, and established a lot of resources, questions, experiences, and caveats on the web over the years. No additional knowledge gain from tutorials or documentation was required as the author of this master thesis had already knowledge of SQLite, only the *Android* API documentation had to be

read. Writing or reading data from an SQLite3 DB can be 35 % faster than writing or reading from the file system [127]. Object Relational Mapping (ORM) is a programming technique for converting data between incompatible system types via a virtual object database for object data conversion and transmission. Operation on a low level enabled own creation of data retrieval without the management by an additional dependency adding, energy increasing, size increasing, and time intensive ORM [110, 128]. All upper advantages saved time required by other application features.

The usage of ORMs can create an overhead for CRUD operations, especially on more complex object structures to load objects consisting of other objects, including reference circles in the DB, can make DB queries heavy and therefore slow. ORM caching, relations, additional dependencies, declarations, configurations, source code adaptations, and their optimizations have a high time and resource investment. ORMs should be carefully treated when new APs ought to be supported, since APs make the loaded objects more complex. Nevertheless, adding other APs could change the currently minimal programming effort with SQLite as the model and data structure would change too.

Especially the more detailed APs as microscopic may have higher transaction volumes and would move the choice from SQLite to *Paper* or *DBFlow* [110], which had the best balance between programming effort and results if the majority of queries are *select*. If more than one other AP ought to be supported, transaction volume, scalability, used resources, and energy consumption become more important factors and impact proportionally the increased effort making *Paper* the best choice in terms of balance. Best performance in terms of CRUD operation speed and energy consumption has again *Android SQLite* with the most effort. The lowest performance has *Sugar ORM* and *Java Realm* due to their caching mechanisms and high computational demand as well as high energy consumption. After the AP assessment concerning DBs, the BP3D flat file collection and FMA DB structure and their interrelations were investigated.

3.2.1 Body Parts 3D

BP3D contains only anatomical models, lacks textures for rendering, and is not designed for an extension via additional information or simple texts. If a DB extension for more supported APs is required, a third-party DB utilizing the FMA indices is advised. Flat-Files are ordinary, unstructured files on the operating system's file system. The relations of single BP3D models with one another or with FMA datasets are defined in additional `.txt` flat files. So BP3D is a flat files collection and not a DB. Transforming a flat files collection into an SQLite DB in this master thesis is beneficial as overall model scaling and conversion are mandatory to use the models in the AR framework ARCore [22]. Different ethnicities and races have different anatomical features making the best fitting humans for the BP3D Japanese, as the models from BP3D are created by and based on Japanese people. Other ethnicities than Japanese ones may experience superimposition discrepancies. A solution would be the deformation [129] of anatomical models to the

detected face proportions and the mobile computation resources are limited as well as the time constraints for this master thesis reserving model deformation for future work.

3.2.2 Foundational Model of Anatomy

The BP3D `.obj` file commentaries contain a Primary Key (PK) to the FMA DB. A conflation of FMA and BP3D can directly be made and is unidirectional, since FMA has more than 100.000 entries and BP3D only 3.493, and only the latter links to the former. The authors of BP3D do not have sequential updates meeting the sequential updates of FMA. As BP3D has its version limits and is the only DB fit for this master thesis, the decision to use macroscopic and regional APs imposes DB $1:m$ relations between an anatomy model entry and its synonyms, definitions, and hierarchies.

3.2.3 Conflation to Own Database

An own SQLite DB optimized for efficiency and small size is created from the raw BP3D text and object files, which are linked with the reduced FMA DB. Adding APs would add more relations in the DB as well as all involved datasets. Further, adding APs would cause more preprocessing in the form of model scaling, format conversion, and merging of single DBs into one monolithic DB are mandatory. In Figure 3.3, an Entity Relationship (ER) model depicts the own, created, minimized, unified *ARnatomy* DB.

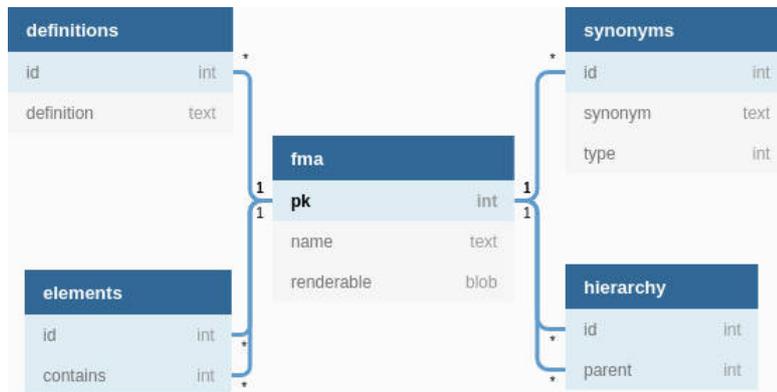


Figure 3.3: ER model of the created *ARnatomy* DB from the conflation of BP3D and FMA.

All other tables in the own DB than the table FMA in Figure 3.3 are added via an own written bash script iterating the generated FMA SQLite DB to insert the needed data into the BP3D SQLite DB as conflation. The table *fma* is the most important one, as it contains unique entries of the now unified FMA Identification (ID) as PK and the column *renderable* as the corresponding BP3D model. The column name *renderable* refers to the now converted model format and the wording in the ARCore [22] framework documentation and source code used for rendering. All other tables than *fma* contain

foreign keys adding properties to the PK of the table *fma*. The tables *definitions* and *synonyms* only add optional or multiple definitions or synonyms to an *fma* table entry. The table *elements* contains other *fma* table PKs and states thereby, which *fma* table entry consists of other *fma* table entries. The table *hierarchy* defines the original FMA hierarchy, which is equal to the anatomy hierarchy in the human body. Read and create queries with SQLite to the generated DB worked immediately and during application testing no delay was experienced. Consequentially, only the native *Android* SQLite API [130] is used for DB transactions. The DB is freely extensible due to the nature of SQLite. New data may be added to the own DB when APs are added to the application, without any changes of already existing DB or source code structures.

3.2.4 Proposal of the Universal Anatomy Index

The work with the BP3D and FMA and the own created DB showed that no concept is applied for the single indices of both. In the following, a proposal for an indexing concept named Universal Anatomy Index (UAI) for past, present, and future anatomical DBs is made. The intention of UAI is to standardise anatomy indices as International Organization for Standardization (ISO) number, foster DB merging, and to have a numerical logic inside all single anatomy indices. A website to search UAI may be installed to show the corresponding anatomy models and to provide hyperlinks to anatomical encyclopedia. The digits of an UAI number are uniquely defined by the established anatomy taxonomy as described in the following paragraphs:

Digit 1: Defines the region of anatomy from the bottom to the top of a human body equal it's numerical value: 0 Feet, 1 Legs, 2 Groin and Gluteus, 3 Abdomen and Lumbus, 4 Thorax and Dorsum, 5 Arms, 6 Hands, 7 Neck, 8 head.

Digit 2: Defines the type of anatomy: 0 Connective Tissue, 1 Adipose Tissue, 2 Fibrous Connective Tissue, 4 Cartilage, and 5 Bone.

Digits 3 and 4: Define together the 10 main systems of anatomy: 00 Circulatory, 01 Digestive and Excretory, 02 Endocrine, 03 Integumentary and Exocrine, 04 Immune and Lymphatic, 05 Muscular, 06 Nervous, 07 Renal and Urinary, 08 Reproductive, 09 Respiratory, 10 Skeletal, and 11 Hematopoietic.

Digits 5 and 6: Define together the main terms of anatomical location: 00 None, 01 Superior, 02 Inferior, 03 Anterior, 04 Posterior, 05 Medial, 06 Lateral, 07 Proximal, 08 Distal, 09 Central, 10 Peripheral, 11 Superficial, 12 Deep, 13 Dorsal, 14 Ventral, 15 Cranial, and 16 Caudal.

Digits 7 - n: Define the actual anatomy in an order considering their already assigned index digits.

As an example, the index *8510001* may identify the frontal bone. The last digit in the previous example is arbitrary chosen and all other digits before have a descriptive character. Memorizing the logic of the first six digits makes the UAI human readable and fast to decode. UAI has the potential to uniquely identify digital anatomy models worldwide. References, updates or linkage to other DB would be easier as the indices would be equal. Fields and people often using digital representations of anatomy save time, descriptive metrics, additional labeling, and a general overhead. Search and identification are facilitated and may be verified by an online mechanism on a web page. Interoperability between technologies using anatomical models is improved.

The numbering is the core issue, multiple trained professionals with proper knowledge about anatomy need to validate and verify the structure in the aforementioned paragraphs or may change it for a better mapping of anatomy to a final unchangeable index. A change in the UAI after its public release is impossible, since the public release would define the purpose of uniqueness and standardization. The indices may be used with marker-based AR to render the anatomy on books, on screens or other platforms and recognized by Quick Response (QR) codes or function as initialization for markerless AR application start. In general, further standardizations and cross linkage in health care, its institutions, the organizing programs and data models in use, patient records, as well as patient consultations can be made accordingly and with one common feature, the UAI.

3.3 Selection of Head Pose Estimation Technology

Asynchronous model loading, HPE, and the model's synchronization with the tracked face in position, rotation, and scaling on mobile device can become a problem due to the high computational demand. As the optimized, minimal, condensed, monolithic SQLite DB was finished, the state of the art in HPE was marked a start of implementation. An investigation of how the single models from the DB can be rendered on their correct facial positions coming from HPE technology in ARCore [22] was conducted. Efforts were made to integrate one of the current HPE papers, which claim to be the state of the art using Cascaded Convolutional Networks (CCN)s or Convolutional Neuronal Networks (CNN)s [104, 105, 108]. Other authors of identified works [101, 102] having not state of the art results and still a good performance in speed and accuracy were contacted too. Unfortunately, the result of the started investigation on state of the art HPE technology was that the source code could not be retrieved nor executed as third-party dependency. For ongoing progress and development another third-party software was utilized.

3.3.1 Third-Party Dependency

The work from Bertók and Fazekas [103] was identified as third-party dependency. For face detection on a computer and on an *Android* mobile device, Bertók and Fazekas [103] claimed to have the same source code and to provide access to the roll information, pitch information, jaw information, and facial landmarks needed for model rotation and

model positioning. Computer source code was already finished and the *Android* source code was made public, and in an early development status. Until the change of the development status from Bertók and Fazekas [103], various programming tasks, UI design drafts, implementation, bugfixing, and writing for *ARnatomy* and this master thesis was conducted. At some point in the own development of *ARnatomy*, the integration of the third-party HPE dependency could no longer be prolonged and a master thesis stagnation settled in. An own detailed time intensive investigation of the C++ source code repository on *Github* named *face-api* from Bertók and Fazeka [103] was conducted. The findings of the own investigation showed that the code from Bertók and Fazeka [103] is a wrapper [42] of an open computer vision library named *OpenCV* [91] combined with a machine learning library named *dlib* [92]. Combined, *OpenCV* and *dlib* offer tools for HPE in C++. Especially *dlib* utilizes the work of Kazemi and Sullivan [131] for fast face alignment. A decision was made to implement an own API of both public available, open, and free libraries *OpenCV* as well as *dlib* on the C++ side of *ARnatomy*. The facial landmarks coordinates, coming from the C++ side, are used for the anatomy model synchronization on the Java side. The own C++ implementation consumed a considerable amount of time and promised to be manufacturable and independent.

3.3.2 Dlib

The algorithm of Kazemi and Sullivan [131] used in *dlib* [92] for face recognition is a cascade of regression trees performing on picture element (pixel) features, which are sparse subsets of intensity values denoted by facial landmarks. As regression introduces geometrical invariance into the process of indexing through the different regression steps, no further prerequisites are needed. The regression estimates lie naturally in a linear subspace of the initial estimate being the average shape of all training data constrained by a generic face detector Bounding Box (BB). Regression learning is performed via triplet generation consisting of the original image, its initial shape estimate, and an update step to fit the given shape, being a model having facial landmarks. Each initial shape estimation is uniformly sampled with gradient tree boosting through a sum of square error losses and provided as input for the next regression until sufficient accuracy is achieved. The cascade regression face detection reduces the detection label in one millisecond and supports partial or uncertain labels as output too [131].

An Application Binary Interface (ABI) connects two binary program modules via an interface defining how data structures or computational routines are accessed in a low-level, machine code. The opposite is an API. Various caveats, configurations, and workarounds for special version compatibilities between the needed technologies Native Development Kit (NDK), *cmake*, *OpenCV* [91], *dlib* [92], Java Native Interface (JNI), and the target mobile phone *Nokia 7 Plus* Application Binary Interface (ABI) configurations were only found via exhaustive search on web pages about help, questions, errors, and issues, where previous developers figured correct configurations by time consuming trial and error. Without documentation of programs, trying to understand the programs features,

properties, input, and output is can only be done with trial and error. Experiences from other developers with similar problems and their workarounds or solutions shared on the Internet helped to have progress in the NDK and *dlib* configuration phase. Official documentation of *Android* NDK with JNI formed only theoretically ideas and no practical outcomes nor minimal examples.

Meticulous software engineering, debugging, refactoring, and source code optimization let to an asynchronous, harmonic interplay of all used technologies. The interplay of technologies was to slow, *ARnatomy* had latency making its usage and the debugging impossible. Optimization techniques were searched and own approaches for more speed and less overhead between Java, NDK with JNI, and C++ were tested. Fortunately, a new version publication of ARCore [22] relinquished all aforementioned grievances and disadvantages.

3.3.3 ARCore

The reasons for the usage of ARCore on *Android* are fast found extensive documentation, NDK support, higher customization possibilities of *Android* than *iOS*, a cheaper *Android* mobile devices, higher world market share of *Android*, minimal example application downloads for development start, a more familiar Integrated Development Environment (IDE), and seven year experience in Java programming all treated meticulously in Section 2.7. The research on the state of the art concerning AR frameworks determined the usage of ARCore [22] as AR rendering framework before the first bit of development was conducted. On the 15.02.2019 ARCore [22] released a new version supporting face tracking and providing an 3D mesh face mask with 468 vertices and the mask's rotation synchronized on each frame with the tracked root face node in the ARCore [22] scene. After short tests all decelerating and complex dependencies being, NDK, *cmake*, *OpenCV* [91], *dlib* [92], and JNI, which made the source code cumbersome and brittle, were removed for the fast realtime HPE of ARCore [22] on the front camera of a mobile device. The single vertex world coordinates of the provided mesh face mask are read and the anatomical AR models are set accordingly without any latency. The final outcome of the development is depicted in Figure 3.4 on the left and the frontal wireframe used for model positioning on the right.

Beards, glasses, eating with cutlery, chewing, drinking from cups or glasses, and long hair beside the eyes do not have any effect on recognition performance and positioning of the face meshes with ARCore. The standard horizontal and vertical plane detection mode of ARCore on the back camera is additionally used in *ARnatomy* for single anatomy model rendering as depicted in Figure 3.5.

With the single anatomy view activity depicted in Figure 3.5, the user can focus solely on isolated anatomy for better learning. Figure 3.4 and Figure 3.5 show the full potential of ARCore version 1.7.0 rendering the macroscopic and regional AP. In the 2D model preview without AR and the back camera 3D AR rendering depicted in Figure 3.5 colour

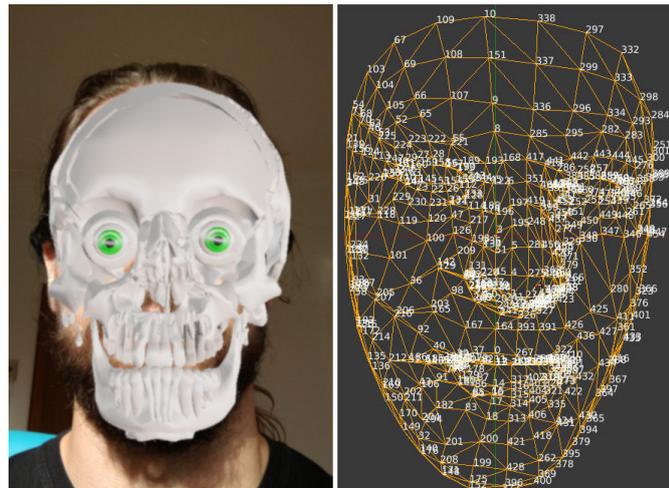


Figure 3.4: Left: AR bone and organ models synchronized with the detected face. Right: Generated image of the ARCore .fbx frontal face mesh indices in wireframe view [132].



Figure 3.5: View of the frontal bone 30 centimeters above the horizontal floor plane with ARCore points for orientation.

errors are observable. There is no certainty that shader errors are the cause, since in the front camera all rendering has no colour errors albeit the model creation and rendering source code is in all three cases the same. The disadvantages of ARCore [22] and its HPE are that both are only indirectly accessible via the AugmentedFace class and the facial mesh, all other source code is proprietary and with ARCore 1.7.0 only the frontal camera can be used with HPE by ARCore.

For now, only HPE is supported by ARCore limiting the superimposition of anatomy to the head. No coordinates are provided for the neck, tests showed that realism, immersion, and accuracy suffer by using jaw coordinate as root location and adding an offset for

correct neck positions. A full body pose estimation can be included in the future by a third-party framework [133, 134, 135, 136, 137, 138, 139, 140], the delivered coordinates should to integrate themselves seamlessly into the ARCore position and rotation data. The most discrepancies were experienced on x axis movements or rotations of the neck and the head. When the head moves the neck has a lot less movement, especially its lower part connected to the shoulders. Additionally, some neck anatomy is missing in the BP3D flat file collection, i.e. the external carotid artery. This master thesis follows the mentioned reasons against supporting neck anatomy or other anatomy than the head anatomy and only supports the latter.

3.4 Development and Design Choices

With the knowledge and experience about the identified technology to be used, choices for further application development and their implementation were made. Back-End (BE) refers in software engineering to the data access layer of a piece of software, or the physical infrastructure or hardware. The BE counterpart is the Front-End (FE). Front-End (FE) refers in software engineering to the data presentation layer of a piece of software, or the physical infrastructure or hardware. The FE counterpart is the Back-End (BE). The application can be subdivided into two big areas, i.e. the BE and FE. The BE provides data from the DB and the framework or API the FE is connected to. The FE presents and renders the data of the BE.

3.4.1 Back-End Development and Design

The native *Android SQLite* API is amended through a *factory* software design pattern [42] for asynchronous queries to the DB with no FE latencies. The asynchrony of the BE is intended to let UI thread of the mobile device, the FE, render a smooth anatomy model rendering, animation rendering, and latency free interaction creating the basis for usability and User Experience (UE). Using an API for the DB access enables BE and FE decoupling for minimal data structure changes and code refactoring inherent with additional AP extensions. No design choices were made for the DB, as the DB was created due to the data format `.sfb` imposed by ARCore. Only models having the ARCore format SceneForm Binary (SFB) can be rendered by ARCore as AR. The conversion of a model to an `.sfb` file performs optimizations as vertex reduction, vertex position normalization, uniform scaling, and barycenter relocation for rendering `.sfb` efficiently on a wide range of devices. Through the `.sfb` conversion, the need for fast, organized model retrieval was required and SQLite has the best performance [127, 110] of all considered DB solutions [109, 110]. During the own DB generation, the focus was laid upon the DB optimization too for excluding future DB optimization iterations. Hence, the requirement of an overseeable, minimal in structure and size DB was fulfilled before it was required by the mobile application development, mobile application design, and mobile application environment.

Basic data structure of an anatomy entities and their data flow through the BE and the FE are treated in the remaining paragraphs of this section. The anatomy source code objects filled by DB data should have one core model, which is a pure DB table representation with few additional meta data info, e.g. if the source code object is fully or partially loaded. Further usage or renderings of the anatomy objects in the FE should use the *wrapper* software design pattern [42] to easily extend the anatomy objects as needed and to attach more meta information required for rendering in the FE. Especially the relation of BP3D and FMA mapped into the generated DB will need to make use of the *wrapper* pattern or container node objects. The container node objects should create, load, and contain the anatomical hierarchy represented in the BE as a tree. The anatomy objects within their hierarchy need another link to their final rendering object in ARCore [22], since only then the textual information about an anatomy entity, anatomy appearance, hierarchy, and its final AR visualization are coupled with each other and can then also be decoupled.

Due to the anatomy hierarchy in FMA and in the own DB, models along with their hierarchy need to be queried in a background thread recursively. A mechanism to determine the hierarchy levels to be queried is necessary, loading the overall hierarchy of just one model may load various other models, which are not needed. Loading the overall model hierarchy increases the DB transaction volume proportionally to the data inside a model hierarchy. As all models are stored as Binary Large Objects (BLOBs), loading the full hierarchy requires a computational demand far too high for a mobile device. Based on the above requirements for the BE, *ARnatomy* is efficient and open for further extensions and more supported APs in the future.

3.4.2 Front-End Design and Development

The FE visualizes data provided by the BE and should have the same basic programming attributes as the BE, high performance, accuracy and low latency. As the FE presents data to the user, additional attributes for the design and the development of the FE are required. Identified attributes required by the user for learning is easy comprehension, low cognitive load, automatic start of anatomy AR rendering, intuitive interaction, and immersive interaction. The identified attributes for the FE are additionally responsible for a good UE and a good usability. A simple interaction has a low cognitive load, which is beneficial for the user as he can focus more on the visualized content as well as learning. The visuals should be reduced to the mere anatomy for minimal distraction allowing the user to focus the attention more on learning than on visuals or interaction. Less distraction and the opportunity of more focus facilitates the discovery-based learning through higher immersion [27]. Additionally, the comprehension of anatomy spatiality is improved [77] and the short-term memorization of the learned content as well as spatial performance is increased [16]. Three activities, Face Activity (FA), Hierarchy Activity (HA), and Single Activity (SA), should be implemented with smooth and fast transitions between them for preserving the generated mental picture and to not generate multiple

mental pictures of the same content [33]. In the FA, when the user looks into the enabled front camera the face of the user is superimposed with synchronized AR anatomy models automatically. The model interaction relies on simple swiping to remove anatomy out of the line of view to dissect the rendered anatomy into deeper layers. The swiped anatomy is added to a cache list, which holds anatomy and enables the user to readd anatomy back into its original position via swiping the anatomy itself. Tapping on an anatomy at any time shows FMA information or renders further possibilities to access the information. Tapping must show at least the anatomy name for simple external labeling [25], learning [27], exploration, and interaction. Long taps show all information available in another view as rendering the full anatomy information would occlude too much space on the screen.

Rendering big labels occluding large portions of the screen may cause lower usability as the interaction possibilities would be consumed falsely by the big labels. With rather big labels, the user has to move his head out of the camera view, which may be eliminated by the ARCore [22] HPE or labels may accidentally capture interaction gestures by overlapping other AR target visualizations. Hence, only simple non-superimposed, external labels [25] containing only the anatomy name in the FA should be rendered. Anatomy filtering by anatomy type or anatomy region must be added in the form of AR to be neutral to the generated immersion. Anything negatively affecting immersion affects UE negatively [27, 76]. Filtering criteria for anatomy regions in the face [141] ordered occipital are frontal (forehead), ocular (eyes), nasal (nose), buccal (cheeks), oral (mouth), mental (chin), and cervical (neck). Filtering criteria for anatomy types [141] in the face, determined by the BP3D are bones, muscles, vessels, nerves, organs, and teeth. As bones and teeth are similar and the latter are normally seen frequently, bones and teeth are separated into two filtering types as depicted in Figure 3.6.

In the HA, the FMA hierarchy is not rendered with ARCore [22]. Instead of AR, only anatomy texts are presented allowing the user to browse through all name, synonym, description, and model datasets similar to a file explorer. If the optional anatomy model is found in the DB, the anatomy model is rendered in 2D within a rectangle as preview without AR. Fast transition to the FA and the SA preserving the currently viewed anatomy model are accessible via a tap. The presented anatomy hierarchy can be amended at one level upwards, which always loads the parent with all its children, or downwards, which loads just the children. A left margin multiplied with the nodes current hierarchy level represents the parent child relation of the single, textual anatomy representations.

In the SA, the user uses the back camera of the mobile device to capture a surface whereon a single anatomy is rendered in AR. The single anatomy rendering is intended for an isolated observation and free interaction with anatomy without distractions or other anatomical context as already depicted in Figure 3.5. Swiping interactions repositions the viewed anatomy on the detected surface to give the user countermeasures for recovering from bad model and surface contrasts. Tapping on anatomy shows all anatomy information available in the DB, since no other anatomy is rendered and no

3. METHODOLOGY

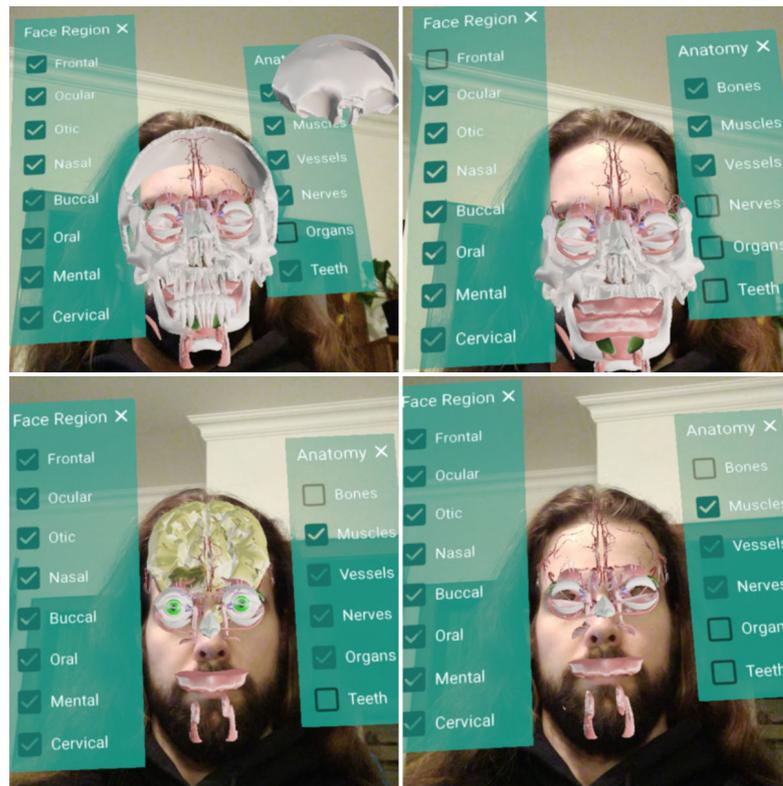


Figure 3.6: Anatomy region and anatomy type filtering through the AR filter options beside the detected face.

other anatomy can be occluded.

The UI was reduced to a minimum for an intuitive, non-complex, noninvasive, immersive appearance to only render content necessary in the moment to let the user focus as unimpeded as possible concerning the UI on the AR anatomy. A Floating Action Button (FAB) performs the primary, or most common, action on a screen. It appears in front of all screen content, typically as a circular shape with an icon in its center. In this master thesis, the actions of a FAB are overloaded depending on the current status. A Floating Action Button (FAB) lets the user access the FMA hierarchy by providing a textual search with automatic suggestions according to the *Google Material Design* [37]. The search FAB location is standard on the bottom right for fast access without a full finger stretch. A text search field is in *ARnatomy* a field where the user can enter text to search anatomy. The search FAB action is expanding or, if already expanded, starting the anatomy text search field. A collapse of the text search field can be done by the text search field via an inline close button. A tap on a single suggestion or search result shown above the text search starts the HA to view detailed anatomy information and to explore anatomy hierarchy. A menu FAB is allowed by the *Google Material Design* [37],

if appropriate to the cause. The menu FAB location is on the bottom left and expands to a stack of actions related to the current activity being the FA or SA. With the BE and FE development and design choices made, their implementation started.

3.5 ARnatomy Rendering

The activity workflow is depicted in Figure 3.7.

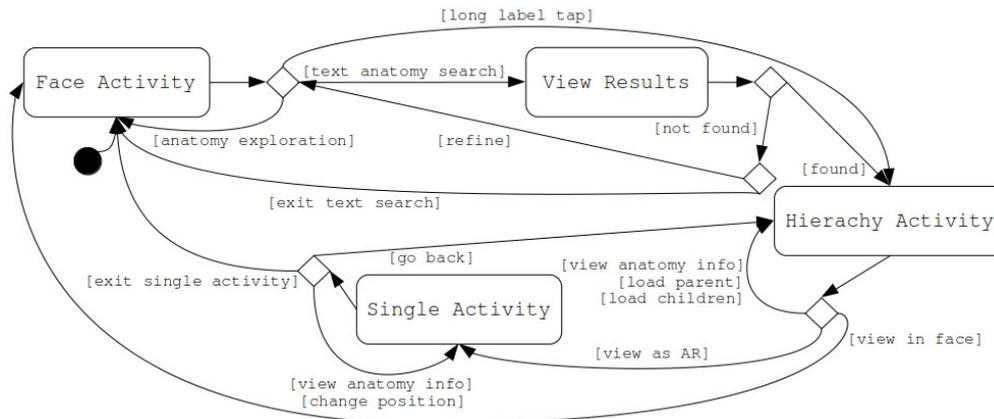


Figure 3.7: UML activity diagram showing the workflow of the FA, HA, and SA. The view results are determined by the user through the search and allows users's to go from the FA to the HA.

The FA is the main activity wherein the user can interact with 3D rendered AR models synchronized with his face. The HA shows additional hierarchical information and 2D rendered models of search results or labeled anatomy from the FA. The SA can only be accessed by the HA and renders single 3D AR models on detected planes for isolated exploration.

Focus ought to be laid upon activity translations to preserve the generated mental picture from the user as long as possible for benefiting from constructivism and embodied cognition [75]. An AP amendment would effect the three single activities, as reasonable translations between the new AP and all existing activities would have to be implemented to preserve the created mental picture in the user's mind. For example, the introduction of radiographic or cross-sectional APs in the form of CT image or MRI volume segmentation [86]. The segmentation volume could be cut by the ARCore face mesh as in standard Volume Clipping (VC) [89]. Contained in the volume segmentation would be the single anatomy objects already positioned through all the combined information from the single images. The inherited anatomy positions would exchange the current dependency on preprocessed .obj models loaded from the own DB for on the fly volume segmentation or preprocessed data models. Further ARCore [22] version upgrades impeded by *Sceneform*

Issue 613 [142] would be eliminated too. Dedicated works [143] leverage medical image data as CT and MRI into cinematic renderings, depicted in Figure 3.8, which has a higher resolution than the BP3D flat file collection currently used.

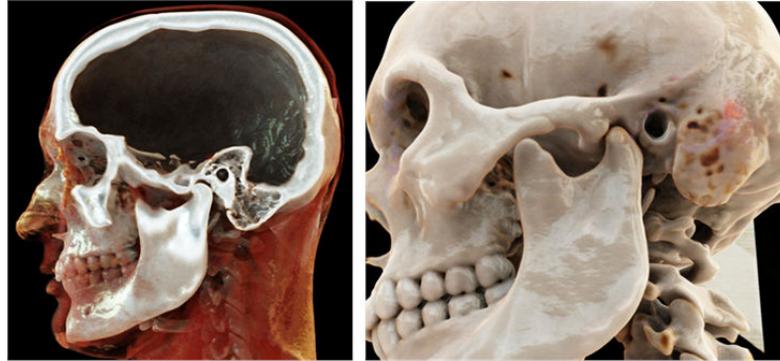


Figure 3.8: Left: Skull interior with the midline as clipping plane. Right: Bony surface rendering. Left and right: Renderings of a non-contrast-enhanced CT [143].

The high resolution can be a problem for mobile devices as it requires more resources in terms of Central Processing Unit (CPU) time, disk space, and Random Access Memory (RAM) space. As a possible solution is that the rendering resolution may adapt automatically to the current available resources of the device. ARCore's 3D mesh face mask with 468 vertices may use few carefully selected synchronization points to synchronize the segmentation volume with the tracked face with rather low computational demand. AR volume segmentation as idea to help patients in understanding or medical staff during consultations with radiographic or cross-sectional images remains as future work. Laymen in the general public would lack the required knowledge to learn or gain insights of radiographic or cross-sectional images. Activity development was in reversed order as the user visits the activities depicted in Figure 3.7, since the SA has the least amount of complexity and the FA the most. However, in the following an elaboration of the three activities and the purpose of their rendering in the same order as the user visits them. Section 3.5.1 treats the FA, Section 3.5.2 treats the HA, and Section 3.5.3 treats the SA.

3.5.1 Face Activity

In the FA, the user can view HPE-synchronized 3D anatomical AR models with their names. The FA depicted in Figure 3.7 uses the front camera and superimposes the tracked faces with anatomy models rendered in AR and allows the user to explore his anatomy on the own body. On first application start up during the DB extraction, the time is used to give the user simple clues on interaction via a dialog as depicted in Figure 3.9 on the left.

All other features beside the introduction dialog in the application are designed to be intuitive, explorable, and hopefully self explanatory. The FA is the main activity and

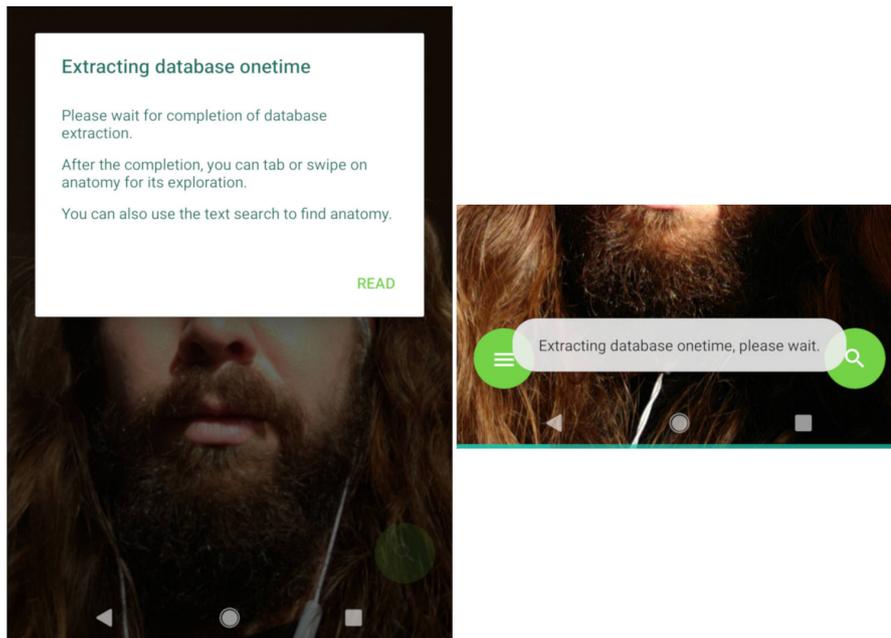


Figure 3.9: Left: The introduction dialog for basic AR interaction clues. Right: The *Android Toast* message shown on search FAB tap to indicate that the database extraction is ongoing.

first activity shown on each start up, the HA and SA can only be accessed through the FA. On each new face detection the anatomy translates in one second on every device out from the back of the face to its front. Intention of the back-to-front anatomy translation is to create a mental picture of the own, synchronized, facial AR anatomy and to stir imagination, curiosity, and interaction.

The camera in the FA is always the frontal camera tracking the 3D face mesh in the captured video stream. Position and orientation of the camera and models are relative to the face mesh, always known, and updated. As ARCore is proprietary, one assumption is that the single coordinates of each frame create and update a facial covisibility graph. A generation of a covisibility graph is similar to Oriented Features from Accelerated Segment Test (FAST) and Rotated Binary Robust Independent Elementary Features (BRISQUE) Simultaneous Localization and Mapping (ORB-SLAM) for constant knowledge of the position and orientation and more robustness with some smoothing algorithm to prevent jittering. Phong lighting is an empirical model of the local illumination of points on a surface. It describes the way a surface reflects light as a combination of the diffuse reflection on rough surfaces with the specular reflection on shiny surfaces. The illumination on all 388 rendered face models is Phong lighting. The stationary directional lighting is used in ARCore to create more depth without any shadows, since real anatomy is very dense and inner anatomy does not cast shadows on other inner anatomy. Having

no anatomy shadow receivers lowers overall computation, particularly important on mobile devices with many models. Specular lighting and reflections are excluded, since natural anatomy has no metallic surfaces and the computational demand would rise with the number of rendered models. ARCore and *Sceneform* [23] contain environmental High Dynamic Range (HDR) light estimation, as depicted in Figure 3.10, since version 1.10 [144] and the *Sceneform* Issue 613 [142] limits *ARnatomy* to use *Sceneform* [23] version 1.7.0.

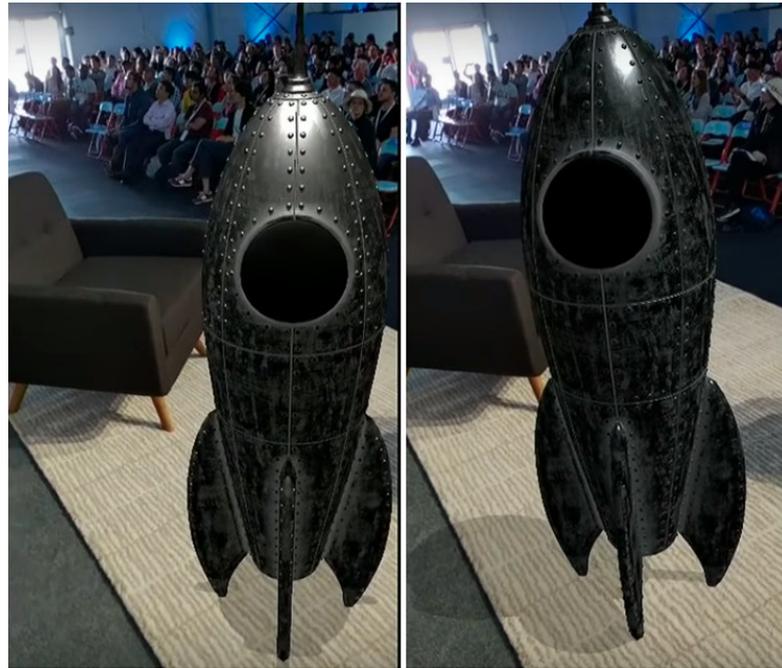


Figure 3.10: Left: Standard Phong lighting with shadows of a metallic model in ARCore. Right: The same as left, but with environmental HDR light estimation, note the specular light and model shadow adjustment to the natural lighting in the captured scene [145].

The light estimation would make the currently stationary directional lighting dynamic in terms of intensity, direction, and attenuation for more realism and immersion, even without specular lights. The rendering colours of the single anatomy types [141] are adapted from *Anatomography* of BP3D [4], which are similar to reality. Different colours for different anatomy types enhance the memorization of appearance, relations, differentiation, position and orientation [146]. The single colours used for anatomy types are depicted in Table 3.1.

As treated in Section 2.1.3, the immersion in the AR FA comes with benefits for learning being increase of interest, motivation, and eagerness. Further advantages of immersion are increased augmented emotional engagement [76], attention, concentration, satisfaction, and spatial abilities [27]. The positive effect on short-term accuracy, speed in fulfilling spatial anatomical tasks [16], and overall short-term performance improvement

Anatomy	Colour	HTML-Code	Anatomy	Colour	HTML-Code
Bones & Teeth		FFFFFF	Gland	●	173B0B
Muscles	●	FA5858	Ventricle	●	2E2EFE
Brain	●	F4FA58	Cartilage	●	CCFFE6
Artery	●	610B0B	Cornea	●	00FF00
Nerve	●	FA58F4	Iris	●	000000

Table 3.1: Rendering source code colours of the mobile application similarly selected to the *Anatomography* of BP3D [4] colours to differentiate anatomy easier.

is established [13, 78]. Other effects as faster memorization or long-term effects as one may assume, are not [16, 78]. Nevertheless, the increasing interest in effects of AR on learning in recent years will certainly continue and bring more insight on the currently sparse researched medium-term and long-term relations concerning 3D visualizations and their effect on learning [76, 16]. The interaction of *ARanatomy* is designed to be intuitive and immersive, the user can tap on an anatomy to show its external label or swipe to translate the anatomy automatically into the cache list. Tapping as selection or event start and swiping as special action are the standard interactions known by most mobile device users, therefore *ARanatomy* has them as input methods.

In general, external labels are positioned on empty screen space and are linked via a line with the concerned objects [25]. In general, internal labels are placed directly in a readable manner on the objects they are labeling depending on the object's surface and geometry. External labels are invoked by tapping on an anatomy and are implemented as the *Mühlner and Preim method* [25], which drastically reduces label position adaptations due to the mobile environment and front camera usage. Mobile environments have limited computation resources and screen sizes, the screen space is occupied with circa 50 % of the user's head leaving little to no space on the bottom, left or right. The head of the user takes a lot of space on the phone leaving room for labels only above the head.

The natural mobile device orientation during front camera usage with hands is vertical, as the width of the small mobile devices, e.g. mobile phones, today are designed for grabbing a mobile device by one hand. Most users tend to capture themselves from a skewed carinal angle, meaning from the front of the abdomen to the head, so only the space above the top of the head remains as external label visualization area. Therefore, the mobile device width and most user's mobile device holding habits during application usage and design, impose a vertical orientation. When the mobile device has vertical orientation, the user's body is on the bottom edge of the mobile device and only space left for labels is the area is above the user's head. Even in the landscape orientation mode meaning holding the mobile device horizontally, the device distance to the user is limited by the length of the user's arms. The length of the user's arms is too small to create enough room above the user' head for label positioning.



Figure 3.11: Coloured anatomy model labels in the FA.

In an external label, only the anatomy name is shown due to limited space inside a label caused by the small area above the user's head depicted in Figure 3.11. If a user wants to see more textual information about an anatomy being synonym, description or hierarchy, a long click, double click on the label or the anatomy activates the HA. The HA is treated in Section 3.5.2 and displays the synonym, description, and hierarchy of an anatomy. The external label foreground colour is based on the luminescence of background colour. If the background colour gets to bright, the foreground colour is set to black to ensure a readable label text. The number of labels determines the colour of the labels, their lines, and the corresponding anatomy will have. Hue Saturation Value (HSV) is an alternative representation of the Red Green Blue (RGB) colour model. The determination of the label background colour is based on the Hue Saturation Value (HSV) colour space. The label count is multiplied with an odd coefficient modulo the 360 degrees of the HSV cylinder resulting in the hue value used for a label with fixed *saturation* and *value* as depicted in Figure 3.12.

For an additional illustrative highlighting, all unlabeled anatomies are rendered transparently. Internal labels [25] are not utilized, since they would cause clutter in the image and due to time constraints. The on the fly projection of text on an arbitrarily formed geometrical object consumes additional mobile resources, time for implementation, and may be labour-intensive due to the limited texture capabilities of ARCore and *Sceneform*. The *cache list* contains anatomy removed from the face. The cache list is located above the user's head and allows the user to scroll through the removed anatomy through

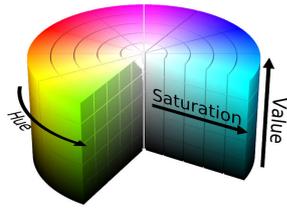


Figure 3.12: The HSV colour cylinder.

swiping to the left or to the right. Adding to or removing anatomy from the cache list is invoked by swiping over an anatomy to represent the capability of digital dissection from the user's own anatomy on his own face as depicted in Figure 3.13.



Figure 3.13: The cache list above the head from the FA with the left frontal gyrus, right frontal gyrus, and frontal bone added.

The idea originated in the circumstance that the human motor system influences the human cognition during a real cadaver dissection [74]. A cadaver dissection requires careful slicing the cadaver with dissection tools whose cutting motion is intended to be represented by the swiping gestures on the mobile device screen. Small motoric involvement of a swipe may not contribute to the anatomy learning process, but active, visual, immersive exploration of the own anatomy has a positive effect on learning [26, 75, 27]. The cache list contains anatomy removed from the face, cached anatomy, and give the user the ability to position the cached anatomy automatically back into the face by a downward swipe on the cached anatomy. The downward swipe is equal to the direction of the original anatomy position and to the mental picture of adding anatomy

back, *downward*, into the face as depicted in Figure 3.14.

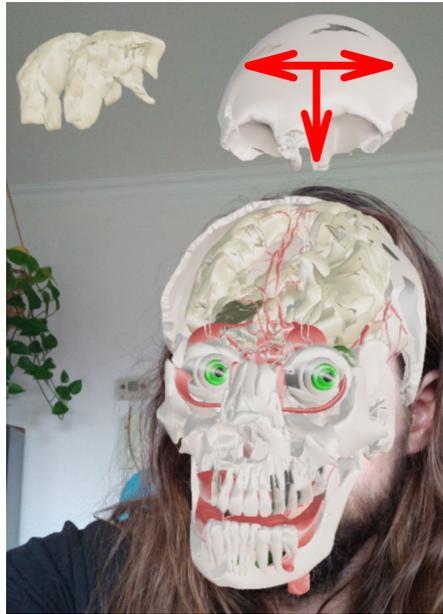


Figure 3.14: Annotated arrows on anatomy in the cache list to illustrate the possible swipe directions. A downward swipe would locate the anatomy back into the face. A left swipe or right swipe would scroll through all anatomies in the cache list to the left or right.

The mental picture of left or right swiping the cache list originates in the *downward* swipe, left or right swiping iterates the cache list. Some of the cached anatomy objects may be rendered out of the screen. The cache list item rendering beyond the screen is necessary, since displaying the overall, growing cache list would create an information overflow on the screen. The information overflow on the screen would draw too much attention away from the face, which ought to have the focus of the user for learning. Tapping in a cache list displays the label directly in front of the anatomy, since not enough space is available to position the label somewhere else.

The capability of moving anatomy from its position across the screen inspired various authors [147, 148] to create a 3D puzzle metaphor and puzzle games. In the puzzle games, anatomy is chaotic, explosion alike, or comes in part by part and the user has to reassemble the anatomy. Depending on the game, the overall compound anatomy can be investigated before its disassembling. The playful component of disassembling or reassembling anatomy helps the user to gain spatial knowledge and spatial abilities [148]. The first intention for this master thesis concerning puzzle games was to include a small 3D puzzle anatomy game, wherein one anatomy model appears above the user's head to indicate its availability for positioning. Then, the user would have to move the anatomy close to its correct position, where the anatomy would snap into its correct and

final position. Positioning other anatomy parts would continue until all anatomy parts are correctly positioned. The primary argument against the puzzle implementation on applications for mobile telephones is the finger coverage of screen space. On a handheld device, the user's face and the user's body are always on the bottom screen edge as well as the finger dragging anatomy models to their destination. The finger which will perform the drag will also mostly obfuscate the bottom screen edge where the anatomical drag destination will be. The user would have to move always his finger during a drag below the actual drag destination, which results with the beforehand obfuscation in poor interaction and UE. The secondary argument against the puzzle implementation is that during dragging, the user may occlude the device camera with his hand or any finger eliminating the HPE and the anatomy synchronization. The overall session and puzzle game state would have to be preserved and recoverable, as the game gets interrupted. Recreating the puzzle game state is highly time intensive to implement with ARCore. A restart of HPE needs a high computational demand and depending on the device processing resources, the time needed would be noticeable latency, which would tamper the interaction and decrease the UE already negatively influenced from the primary argument even more.

Derived from the primary and secondary argument, 3D AR anatomy puzzle games should only be implemented on mobile devices having bigger screens than standard mobile telephones to compensate the finger screen obfuscation. As *ARanatomy* is intended for the general public and to support as much mobile devices as possible, an own implementation of the 3D anatomy puzzle in AR remains undone. The FA has a different rendering technique in comparison to the hierarchy and single activity. An obfuscation of all anatomy models behind the 3D face mesh from ARCore inside the user's head results from the not changeable highest rendering priority of the 3D face mesh. As countermeasure to the not not changeable priority of the 3D face mesh, a high level illustrative rendering technique [85, 86], transparency, controlled by z axis translation through the pinch gesture is implemented.

The user can translate the overall head anatomy via a pinch gesture on the z axis in and out of the head. Single z axis threshold values set a focus area and context area on the z axis. Anatomy in front of the 3D face mesh is the focus area. Anatomy far in front of the 3D face mesh beyond the focus area is in the context area. In the focus area, the interaction with anatomy works as previously explained. In the context area, the anatomy is rendered transparently and does not react to taps. Contextual anatomy can be removed with via swiping and context anatomy translates taps as raycasts rostral until one or no focus anatomy is hit. Anatomy behind the 3D face mesh is not rendered as it would be obfuscated by other rendered anatomy, as depicted in Figure 3.15, and by the ARCore face mesh itself.

Further decrement of the transparency depicted in Figure 3.15 is restricted by low level details discussed in Chapter 4. In the following, the mere intention of transparency usage



Figure 3.15: Visualization of transparent context revealing anatomy located deeper inside the head.

is treated. The user can translate the overall anatomy on the z axis via pinch gestures. The primary intention of the z axis translation was the countermeasure for the 3D face mesh obfuscation. The secondary intention of the z axis translation is to give the user the ability to explore deeper anatomy inside the head. The tertiary intention of the z axis translation is to highlight the anatomy in the focus area through rendering all the anatomy beyond the focus area, being in the context area, as transparent. Interaction on focused anatomy behaves as mentioned before, context anatomy translates taps rostral via raycasts until one or no focus anatomy is hit. Swipes on context anatomy adds context anatomy to the cache list, since in some cases a contextual anatomy can obfuscate focused anatomy. With the filtering options treated in Section 3.4 and depicted in Figure 3.16 as AR rendering, anatomy can be filtered with direct feedback on any selection of the options.

To keep the immersion high, the cognitive load low, and to avoid additional head movements for the user, the two filter panels of anatomy region and type are rendered on the sides of the head close to the anatomy they are filtering. If the whole body is supported in the future, the filter panels ought to be scaled to the size of the currently visible body and rendered in empty space, similar to external labels, and without the connection line.

The text search expands when clicked on the search FAB button. The user can enter anatomy terms to search the DB, similar names will be suggested above the text search field. A tap on a search suggestions activates the HA, which shows detailed information about the anatomy. The expansion of the FAB text search disables, hides, and replaces

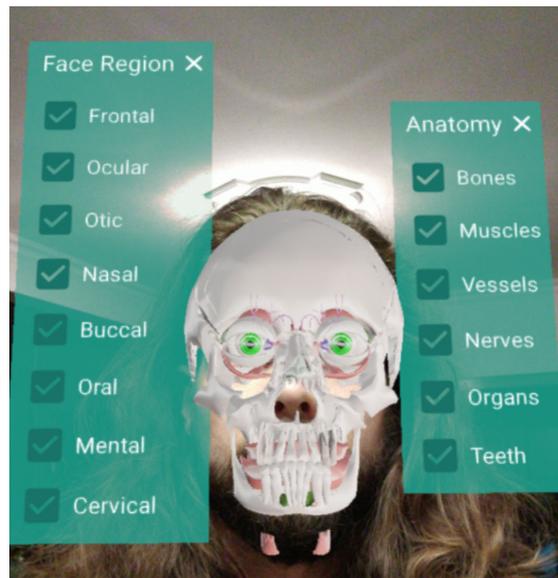


Figure 3.16: All checked FA filtering options rendered in AR beside the detected face.

the menu FAB, treated in Section 3.4, with the inline close button of the text search field. A tap on the close button collapses the search text field back into the search FAB and the menu FAB becomes therefore visible again. All results found by the text search are displayed as alphabetically ordered anatomy name grid where the rows have synchronized row heights and widths for an unified appearance. To indicate that a model is contained in the DB for a search result, an eye icon is added inside a grid cell as depicted in Figure 3.17.

The text search is only available in the FA, because the HA and SA function as informal extension of FA without any face synchronization. The second FAB in the FA, the menu FAB, is overloaded, meaning depending on the application state, the menu invoked by a tap on the menu FAB contains different items. The actions of the menu FAB are application restart, change between FA and SA, and settings. If a face is currently tracked, a fourth action appears in the menu FAB to disable or enable the aforementioned AR face filtering options depicted in the previous Figure 3.16.

The reason for the minimal UI with only the search FAB and menu FAB is not to overload the screen and the users cognition. If the user restrains the permission for the mobile camera, a black screen only with the text search will be shown without any AR. To access previously found anatomy faster, a table in the DB is created programmatically for saving the anatomy indices on the `recents` table. If the text search field is empty, the names of the anatomies in the `recents` table are shown for fast access. A tap on a recent anatomy results in the same as a tap on a search result, the HA is started and shows detailed information about the taped anatomy.

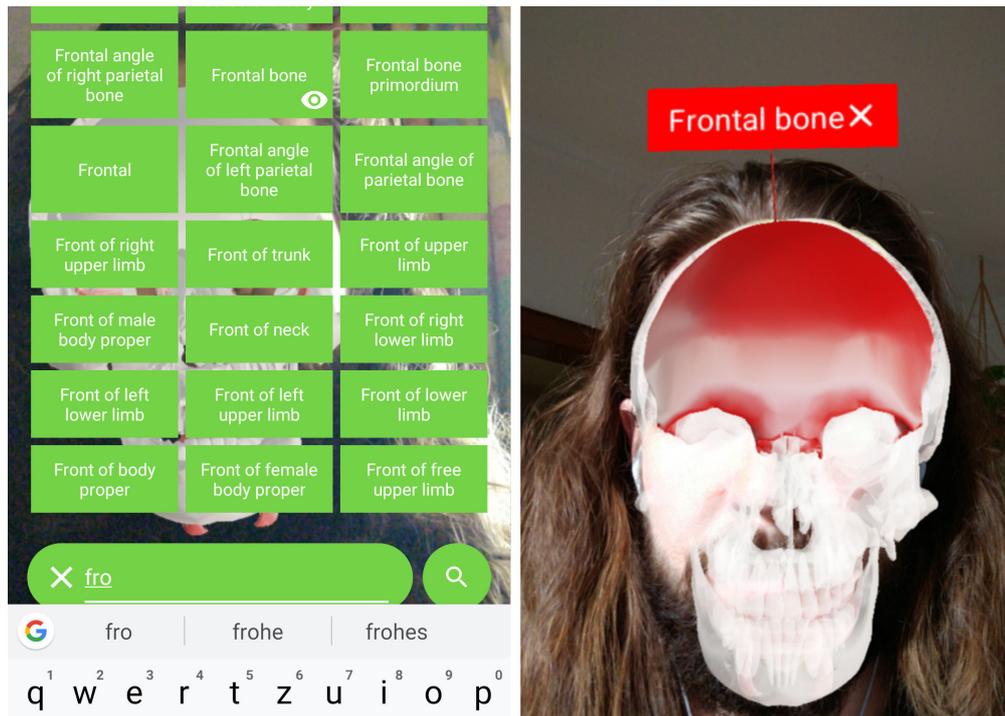


Figure 3.17: Left: Textual search for the first characters of *frontal bone* with its results displayed as grid. Right: The frontal bone with its activated external label synchronized with the face.

3.5.2 Hierarchy Activity

In the HA, the user can view 2D anatomical AR models with all their information and explore the anatomical hierarchy. The HA workflow depicted in Figure 3.7 is a standard *Android* 2D activity displaying the full information about the anatomies in the DB and allows the user to explore the anatomy model hierarchy without AR. Interaction is designed to be intuitive, the user can tap on an anatomy name to show all its information below the name. The menu for each single anatomy located on the right enables the user to view the anatomy in the SA, load the anatomy's parent or anatomy's children. If the anatomy has a model in the DB, an eye icon is displayed on the left side of the menu to indicate visualization capabilities of the current FMA entry. The icons of the anatomy menu in the HA are selected to represent their actions as depicted in Figure 3.18.

If the action of a popup menu entry depicted in Figure 3.18 cannot be performed on a tap, e.g. no more hierarchy available, the entry is removed of the popup menu and an *Android Toast* message is displayed. If a model for the current anatomy object is in the DB, the model is rendered in an 2D *SceneView* indented as preview. The *SceneView* reacts on pinch gestures for zooming gestures, twisting gestures for rotation as well as swiping for camera translations to view the model from any desired perspective. The difference

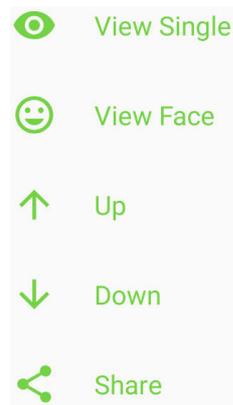


Figure 3.18: The popup menu from a single anatomy entry in the HA. The eye starts the AR anatomy model rendering in the SA. The smiley starts the FA with the corresponding anatomy highlighted by its external label. The up or down arrows try to load further anatomy hierarchy. The share icon starts the *Android Sharesheet* [149] to select a target app as content receiver.

of the 2D `SceneView` in the HA to the whole SA is that the SA is only startable via the popup menu from the HA. In the SA, the user can freely walk around the 3D anatomy rendered as AR with the mobile device and with far more degrees of freedom as well as immersion than in the 2D `SceneView` in the HA. Going one level upward in the hierarchy is equivalent with loading all the information of the parent and the parent's children available in the DB. The loading of all children from a parent is required to display the overall correct anatomy hierarchy. Going one level downward in the hierarchy is equivalent with loading all the information of all children available in the DB.

Adding APs may bring more crosslinkage to other anatomy objects and data as depicted in Figure 3.19. The popup menu depicted in Figure 3.19 allows the user to go to the face or single activities with one click preserving the created mental picture, which is beneficial for learning [13, 15]. Each model has its own popup menu to have fast direct menu interaction. The crosslinked data coming from additional APs should be included in the HA and displayed as a preview with a possibility to go to the activity the crosslinked data belongs to. Fusing anatomy data in the HA should give the user multiple APs on the same anatomy, amend the mental picture but not create a new one [33], a better overview for learning, and the user's next action.

3.5.3 Single Activity

In the SA, the user can view single 3D anatomical AR models with all their information isolated and rendered on a detected plane. The SA workflow depicted in Figure 3.7 uses the back camera of the mobile device to detect horizontal or vertical planes. On the horizontal or vertical planes, single selected anatomy models are rendered in AR depicted in Figure 3.5 to allow the user a single anatomy 3D exploration. If no plane is detected

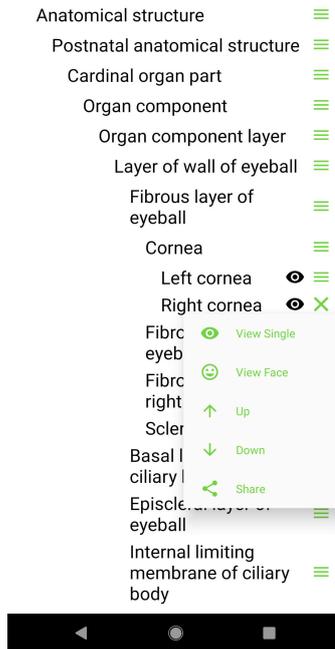


Figure 3.19: The full FMA hierarchy of the left and right cornea displayed in the HA with the popup menu of the right cornea opened.

in the first milliseconds, the standard ARCore PlaneDiscoveryController picture depicted in Figure 3.20 is shown to invite the user to circle his mobile device for a more efficient plane detection than holding the mobile device motionless. The interaction



Figure 3.20: The hand motion picture of ARCore to indicate mobile device movement for better plane detection.

in the SA is consistent with the interaction in FA. The difference from the SA to the FA is the action on tapping an anatomy, which shows an external label with its name, synonyms and definitions rendered as 2D view in AR. The 2D, AR view rendering for all

anatomy information is advised in the SA, since no other space claiming, distracting or interaction capturing elements or anatomies are on screen. Swiping or dragging a finger on the screen repositions the anatomy on its plane with the intention that the user can adapt to environment changes or move the anatomy model. Anatomy movement in the SA is intended to recover from an impeded visualization, e.g. a low contrast between the rendered anatomy and the background. Pinch gestures scale the model to make details visible as the human anatomy encompasses big and small entities. With the AR rendering, the user can walk freely around and even lose the model as PR if, and only if, the surface identified as a plane is still tracked by ARCore. Adding an APs in the SA may only be done via an entry in the previously treated menu FAB. The menu entry should switch to the dedicated activity required for the added AP and show the already loaded anatomy in a manner, which depends on the AP. Extending the currently external label of SA consisting of name, synonyms, and descriptions may overexert the user. The added AP activity should allow the user to go back to the SA with one tap as accidental activity changes should be correctable. On first explorations, a user can feel lost when exploring the single activities one by one.

3.5.4 User Interface Colours and Icons

The link colour in *ARnatomy* is green and was chosen because of its association with openness, peace, calmness and success [150] all features of mother nature, the creation force wherein all life on earth originates. Green has a positive influence on humans considering experience and performance during task execution. Green fits therefore well as interaction colour in an application for anatomy exploration and education. In medical institutions green is often used for its calming attributes and as green is the contrary colour of red, i.e. blood, which enhances the visibility of the colour red. The fact that vessels and muscles are rendered in red gives an additional reason to use green as indication colour. Green and red are never used together in *ARnatomy*. On a progress bar indicating a status, all colours including green have only an insignificant effect on progress perception [151]. To have an appearance fitting with the link colour green, the progress bar showing the extraction status on first application startup is turquoise. Turquoise was chosen because amounts of green are inside its blue. All colours used in the UI are depicted in Table 3.2, wherein the link colour was initially selected and all other colours emerged naturally during the implementation by abiding the guidelines of *Google Material Design* [37].

The rendered anatomy is in similar colours as in the *Anatomography* of BP3D [4] depicted in the previous Table 3.1. Some default Scalable Vector Graphics (SVG) icons of *Android* are suboptimal and during their optimization the idea of making an own icon for *ARnatomy* emerged. The icons drawing path's determining the foreground contain more vertices than their not reducible complexity imposes. All suboptimal icons in *ARnatomy* are redrawn by hand with minimal path vertices and without any representation change or for using them as *Animated Vector Drawables* [152]. Animated vector drawables required from all involved SVGs to have the same number of values in their paths to transforme

3. METHODOLOGY

Name	Colour	HTML-Code	Purpose
Primary		1FBFA9	Information
Primary Dark		187468	Dialog Text Color
Link		74D246	Indicate Interaction
Text Primary		FFFFFF	Standard Text Color
Negative Text		000000	Compensate Bright Background
Scene View		BDBDBD	2D Scene View Background

Table 3.2: UI colours of the mobile application with their assigned purposes.

them from one SVG into another SVG as realtime animation. The advantage along pixel-free scaling of .svg animation is that status changes in systems can be visualized and incorporated in icons. The icon change communicates to the user that an action has now changed to another action and helps the user to understand the application status more easily than without the icon change. The application icon depicted in Figure 3.21 has a not reducible complexity of paths in its .svg file and was inspired by the suggested AR default icon in the talk from Unpingco and Faaborg [153] on Google's *I/O '18* conference.



Figure 3.21: Designed application icon.

The human in Figure 3.21 is intended to represent the human anatomy within the borders of AR.

Implementation

The Section order in this Chapter is similar to Chapter 3. Section 4.1 treats the identified and applied technologies through the search for solutions for fulfilling the set development choices, design choices, and witnessed problems during the implementation. Section 4.2 treats the conflation of the identified Body Parts 3D (BP3D) [4] flat file collection and the Foundational Model of Anatomy (FMA) [5] Database (DB) fitting the macroscopic and regional Anatomy Perspectives (APs) due to their inherent properties. Section 4.3 treats the development process of the Head Pose Estimation (HPE) used in *ARnatomy*. Section 4.4 treats the development and design choices on source code development for refactoring, maintenance, and extension.

4.1 Applied Technologies

The mobile application *ARnatomy* is designed for *Android* 8.1 Oreo Application Programming Interface (API) 27.1.1, ARCore 1.7.0, Structured Query Language Lite (SQLite) 3.19, build with *Android Studio* 3.5.2 with Java 1.8, *Android* Software Development Kit (SDK) tools 28.0.3, *Gradle* 5.4.1, and *Android Studio Gradle Plugin Version* 3.5.2 on an Ubuntu 19.04 operating system for a *Nokia 7 Plus* mobile phone. The generated SQLite DB is a conflation of BP3D 4.0 and FMA 5.0.0. As the programming language Java is used to build an application on the operating system *Android* developed by *Google*, an effort was made to implement the source code in adherence with the *Google Java Style Guide* [35], *Google Developer Policies* [36], and *Google Material Design* [37]. The usage of all technologies mentioned before with details of their implementation is treated in all subsequent sections.

4.2 Selection of Databases

The macroscopic and regional APs had no direct influence on the implementation. Only an indirect influence, because the two APs determined the need for the BP3D flat file collection and the FMA DB with their preprocessing to create an *ARnatomy* SQLite DB. As SQLite was identified to have the best performance on *Android*, BP3D and FMA with their unidirectional relation from the former to the latter had to be preprocessed to receive at the end a conflation of the two DBs.

4.2.1 Body Parts 3D

Most of the BP3D `.obj` flat files are already linked to FMA datasets by their source code comments and exceptions without any linkage to FMA exist. Some anatomical structures in FMA may contain only one logical part in BP3D as name or model. For example, the internal carotid artery exists on BP3D version 4.0, and the external carotid artery is missing partially. Some but not all sub parts of the external carotid artery are included in BP3D and in the FMA DB, all anatomical entries of the external carotid artery exist. Another example is the skin, its model from BP3D is for the whole body without any parts nor different skin layers. Only, if the full body is supported, the skin model can be used as rigid model. Since anatomy below the head moves differently than the head, the single skin model for the whole human body is excluded in the rendering to avoid unrealistic visualizations. Two symmetric halves of an anatomy structure can be contained as one single model, an example is the external ear depicted in Figure 4.1.

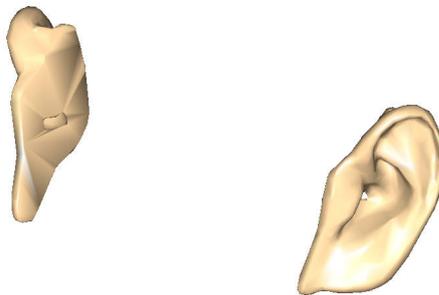


Figure 4.1: The external ear model from BP3D flat file collection version 4.0 [4].

The ears of a user are asymmetric or, if a user has a wider head than the ear distance in the model, the asymmetric ears would not be reflected in the application, decreasing immersion and increasing model discrepancy and accuracy discrepancy. The same applies to other anatomy models, e.g. the mandible position may be correct and its proportions may not align with the detected face proportions. Some Identifications (IDs) are uncontentious through the single BP3D versions, as an example see the *3949 Right internal carotid artery* and *4062 Left internal carotid artery* in BP3D version 4 and BP3D version 4.3. All filterable models from a version of the BP3D should be verified before model usage with their online anatomy model view *Anatomography*. For example in

BP3D version 4.3, all left head veins are missing in the BP3D flat file collection. The lack of corresponding, symmetrical models can affect various BP3D versions and models, e.g. in version 4.0 the *83966 Tentorium cerebelli*, *62004 medulla oblongata*, and the *74877 mammillary body* are missing their symmetrical counterparts. All aforementioned irregularities had to be considered during the implementation phase, as they cause special cases in loading models and rendering models.

4.2.2 Foundational Model of Anatomy

For the investigation of the FMA, its newest version was downloaded as Comma Separated Values (CSV) file. The CSV file was converted into a SQLite DB with a converter from *Github* [154], written in the programming language *Python*. Newer versions of the downloaded FMA CSV file fail [155] the utilized conversion program [154] on *Github* used for conversion to an SQLite DB. The format of the newer FMA CSV file contains syntax errors [156], which need special error handling as quotes and commas in the CSV file were misplaced. To overcome the conversion issue, an own bash script for the conversion from the FMA CSV file to a SQLite DB was written. The written bash script was published on *Github* [157] to help other developers experiencing the same problem.

An extension of the FMA can easily be executed by adding a foreign key column to the FMA or using its Primary Key (PK) in another DB as foreign key. Single datasets in the original CSV FMA file are solely textual and contain a vast number of blank entries, which are sequentially filled over time with version updates. During preprocessing, an optimization to reduce empty datasets was conducted, only the name, synonyms, definitions, and hierarchies of FMA are used. Hierarchies can be traversed from the top to the bottom, known as top down, or from the bottom to the top, known as bottom up. A hierarchy table with the full bottom up paths of a single anatomy entry to its root anatomy is created. Additionally, the created hierarchy table allows the Back-End (BE) to load the anatomy hierarchy faster than without the bottom up hierarchy. The FMA contains similar irregularities as the BP3D, not all FMA datasets, which are linked to BP3D models in the generated *ARnatomy* DB, contain synonyms or definitions. With the knowledge gained from the normalization from BP3D and FMA, their conflation to one *ARnatomy* DB could be executed.

4.2.3 Conflation to Own Database

The BP3D models had to be linked with their name, synonyms, and definitions from FMA to be able to query all anatomical information about an anatomical identity. All models from BP3D are provided as 3.493 `.obj` flat files, which are highly inefficient for fast search and especially in the context of mobile Augmented Reality (AR) model rendering. No management system nor a DB querying language can be applied in the mobile environment to have fast flat file access for high numbers. Fortunately, the standardized comments at the beginning of an `.obj` BP3D flat file followed by the object vertices make a line by line readout trivial. One has to first parse all flat files from BP3D

and insert them into a newly generated DB to query single datasets and to apply further automatised preprocessing. The standard scale in ARCore is in meters, the BP3D scale is in millimeters. An implemented multithreaded bash program generates downscaled copies of all `.obj` files. The preprocessed downscaling prevents the downscaling during runtime on the mobile device, i.e. the demand for downscaling on the mobile device is removed and saves mobile device resources. The downscaled files are then converted into `.sfb` files required by ARCore via an own multithreaded script. Writing a script was necessary, as the Google *Sceneform* [23] team refused a feature request for their converter to scale multiple `.obj` files with one converter invocation [158]. In computer programming, boilerplate code or boilerplate refers to sections of code that have to be included in many places with little or no alteration. It is often used when referring to languages that are considered verbose, i.e. the programmer must write a lot of code to execute small tasks. *Sceneform* [23] is an extension of ARCore, which makes programming faster and easier by preventing to write boilerplate code for simple AR rendering. An Java application realized for this master thesis, named Files 2 SQLite (F2S) [159] and controlled via command line arguments, is freely available on *Github* and generates a SQLite DB of all files in one directory.

F2S is used to generate a DB from all scaled and converted `.sfb` files and allows the programmer via command line arguments to extract data from the single file contents. F2S optimizes the BP3D flat file names so they can serve as integer indices by avoiding their prefix *FJ* and replacing their suffix *M* with *001*. Another SQLite DB is generated with the FMA indices from the comments of the `.obj` flat files. The DB containing the `.sfb` and the DB containing the corresponding FMA indices are joined to have a single DB containing only the name, the `.sfb` model, and the FMA indices of the BP3D flat file collection. Finally with the contained FMA indices, the synonyms, descriptions, and the hierarchy from the original FMA are joined to create the *ARnatomy DB* used in the developed mobile application. Minimization of the *ARnatomy* SQLite DB had the advantage of facilitating the data retrieval and decreased the *ARnatomy* DB size to 80.4 Mega Byte (MB) packed as XZ [160] archive. Unpacked, the *ARnatomy* DB has a size of 290.5 MB on the mobile device. The raw BP3D `.obj` files have a size of 489.6 MB and the raw FMA file 15.8 MB in total 505.4 MB. The overall file size reduction through the optimized conflation of BP3D and FMA is 42.52 %. Faster downloading, installing, querying, less complexity, and less resource demand are the benefits of creating the optimized *ARnatomy* SQLite DB. Without minimization, the BP3D and FMA would have in summary 231 columns defined, which are listed in Appendix A.1. Most of the FMA columns are blank and would not have been needed for *ARnatomy*. In the *ARnatomy* DB, only 12 columns are required to fully contain all model information and text information of BP3D and FMA. The *ARnatomy* DB gets decompressed once at the first application start.

4.2.4 Extensions

The majority of queries executed by the application are *select* queries. The query performance may be increased due to more data added into the DB, e.g. when additional APs are added to *ARnatomy*. The usage of SQLite brings optional extensions, the most important one for *ARnatomy* is the Full Text Search (FTS) 5 [161]. FTS engines search efficiently large texts or documents for one or multiple search terms, which can have a high impact on the text search from the Face Activity (FA). The short synonyms and description entries coming from the FMA, the already fast subtext search with wildcards in SQLite, and the lack of time in the current implementation leave FTS 5 unused.

Another technology with up to six times more performance concerning *create* operations is SQLiteKey Value (KV) [162]. Log Structured Merge (LSM) trees are search trees who have a high performance on indexed big insert volume scenarios. KV pairs are maintained in two or more separate structures optimized for KV's underlying storage medium and data is synchronized between the two or more separate structures in batches. SQLiteKV adopts a Log Structured Merge (LSM) KV DB engine and maintains an SQLite interface, also on mobile devices. A performance gain on small *read* operations below 128 bytes is also given. If more APs would be added, SQLiteKV could be a good choice to track the user actions via logging on a high frequency for more fine-gained debugging, perspective analysis, or trajectory analysis, which require all *create* operations on a high rate.

4.3 Selection of Head Pose Estimation

First, the utilization of Bertók's and Fazeka's [103] work for HPE was tried to be integrated into *ARnatomy*, but this failed due to timing. Second, an own implementation of HPE with the tools and libraries Bertók and Fazeka [103] were using was conducted. Third, a new version of ARCore was released, which contained HPE making the previous two attempts to integrate HPE in *ARnatomy* obsolete. In the following three subsections, the brief course of the three aforementioned events is treated in detail. The purpose of the following three subsections is to report the development approach leading to the results and to inform thereby future developers working with *ARnatomy* what can be done and what should be avoided.

4.3.1 Integrating Dlib

The time schedule of the master thesis, the lack of documentation and interoperability from Bertók and Fazeka [103] with *ARnatomy* caused a detailed source code investigation of the work from Bertók and Fazeka [103]. The usage of the open computer vision library named *OpenCV* [91] and the machine learning library named *dlib* [92] was recognized in the C++ source code of Bertók and Fazeka [103]. *OpenCV* and *dlib* combined offer tools for HPE in C++, especially *dlib* utilizes the work of Kazemi and Sullivan [131] for fast face alignment. An own implementation of mobile face tracking was conducted. The own

implementation transmits image data via Java Native Interface (JNI) from Java over the set up Native Development Kit (NDK) to the C++ API, which feeds *dlib* [92]. The detected facial coordinates from *dlib* are then returned to Java from the C++ for the final anatomy model synchronization with the user's face. The implementation consumed a considerable amount of time and promised to be manufacturable and independent from unfinished third-party dependencies. The programming approach to detect facial landmarks with *dlib* was step by step. First, a functioning face detection example was compiled and fed with the video stream of the webcam from the development laptop to get familiar with the *dlib* source code. Second, source code changes were applied to change the face detection on a laptop webcam to a C++ API for an *Android* mobile application. The C++ API is required for communication between Java and C++ via JNI, all within *ARnatomy*. Compared to the time spent on attempts to integrate the work of Bertók and Fazeka [103], a robust face detection with the HELEN face dataset [163] on the laptop webcam was achieved very fast. The HELEN face dataset consists of 2330 images with 194 face landmark annotations on each image. One example image with its visualized landmarks is depicted in Figure 4.2.



Figure 4.2: Example picture from the HELEN face dataset [163] with 194 face landmarks.

The creation of the C++ API for the mobile device was time consuming, as tasks, which are simple on the high level laptop webcam, were complex through programming language peculiarities and programming language differences on the low level mobile device. As promising the documentation of *OpenCV* [91] with *dlib* [92] reads, as cumbersome is its implementation of HPE on mobile devices. The NDK documentation needed for communication between Java and C++ lacks small, concrete statements, and small examples to begin with. In the NDK documentation, mere abstract, high level instructions without

low level instructions, low level explanations, and simple low level examples are written. For example, *cmake* executed by *Android Studio* via NDK was unable to find the installed library *libX11*. The required paths had to be set by hand, although *cmake* provides an own logic to find libraries and to link libraries. *cmake* fails to find *libX11* on Ubuntu 19.04 and fails to report its inability in the first place [164]. Further problems and workarounds emerged during the implementation of the C++ face detection for *ARnatomy*. A developer requires more time to track down the error sources without any error reporting than with error reporting. In software engineering, a fail-fast programming principle [165] checks the system status frequently and reports immediately any indication for a failure. Own failure reporting *cmake* scripts applying the fail-fast programming principle resolved the malfunction by validating each involved step of finding and linking *libX11*.

An example of the poor NDK documentation of *Android* increasing the development time and development complexity of HPE. To log messages into the *Android Studio* console from the C++ side, the method `__android_log_print` should be used and not `__android_log_write`, since the latter has no effect at all. And for `__android_log_write`, the inclusion of the `android/log.h` file has no effect. Only the linkage of `find_library(ANDROID_LIB android)` in the *cmake* file lets only `__android_log_write` write data to *Android Studio*. The information about `__android_log_write` can only be found on question, error, issue, and help pages and not on the official documentation from NDK. Without the information about `__android_log_write`, one just wonders why no log output arrives and searches first for other error causes. Only exhaustive online search gives an answer or a brute force workaround being no logging at all, which makes debugging rather cumbersome. Additionally, no reference to the required JNI C++ method name mapping in the NDK documents was given. With a small mobile application example a developer could investigate and learn from a working source code required for sending log messages from C++ with JNI and NDK to Java or the *Android Studio*.

After fixing the aforementioned errors, the first bitmap data from Java received the first detected landmarks from C++. The C++ API uses *dlib*'s shape predictor trained with the *HELEN* face dataset [163]. The full transmission process from Java to C++ and back is treated in the remaining paragraph. The application obtains asynchronously in Java the smallest colour image retrievable from the mobile device camera with a resolution of 640 x 350 picture elements (pixels), if the face detection thread is ready. According to *dlib*, small images are sufficient for face detection, so the image is downscaled by a factor of two to a bitmap. Only coloured images can be retrieved by ARCore, simultaneously to the downscaling, a colourspace transformation to greyscale for a lower computational demand in face detection is performed too. Bitmap transmission from Java to C++ via JNI is executed. The bitmap is inserted into *dlib* [92] to receive the face landmark data. A landmark data format conversion to a Java Three Dimensional (3D) array for retransmission from C++ to Java is executed. An objectification from

a 3D array into a Java class for correct positioning in ARCore is performed to have the anatomical positions ready for *arc* node positioning. Only now the face detection thread is ready and asks ARCore to obtain the next image from the mobile device camera.

Three Frames Per Second (FPS) of the mobile face detection running on a *Nokia 7 Plus* were a poor result, which delivered only tampered interaction and weak synchronization with the user's face. Further or other optimization possibilities along with the ones already applied were not known and not found in the *dlib* documentation, on issue pages, and on the Internet. As last resort, a reduction of the used 194 facial landmarks in the HELEN face dataset to 15 should lower the computational demand. The *HELEN* shape predictor file was 187,4 MB big, which contributed to the startup time and high resource demand on the mobile device. After processing and generating the required `.xml` file for *dlib* [92] shape predictor training, containing the needed *HELEN* face landmarks a 233development stop was the case. The reason was that the shape predictor training consumed a high amount of Random Access Memory (RAM) resulting in no evident processing progress over several days on the development laptop. During the search in the *Github* issues of *dlib* to eliminate errors occurring during the aforementioned landmark reduction, an answer from the creator of *dlib* to a developer experiencing similar problems gave further insights. At least 16 Gigabyte (GB) RAM are needed for shape predictor training, otherwise no shape predictor training should be conducted, which is a highly important information excluded in the documentation of *dlib*.

The high demand of RAM for shape predictor training strongly decreased the training progress on a 4 GB RAM laptop. In another found *Github* issue of *dlib*, a *dlib* shape predictor file with only five facial landmarks was mentioned. After some search, the unreferenced repository from the creator of *dlib* was found, which was not mentioned in the *dlib* source file commentaries nor the *dlib* documentation before version 19.0. One week of work and the task of retraining *dlib* with the 2330 pictures of *HELEN* with 15 facial landmarks became obsolete. The already trained five facial landmark shape predictor file with 7485 pictures and with a size of only 5.7 MB, can be used for face detection with far less computational demand. The five facial landmarks have the same landmark accuracy as the landmarks from the *HELEN* dataset. The five facial landmark shape predictor achieves accurate, robust facial landmark detection and identifies the corners of the eyes and the bottom of the nose.

The usage of only five facial landmarks as shape predictor increased the FPS of *ARnatomy* to 23, which was still insufficient for smooth interaction without latency, but sufficient for developing and debugging. The need for tracked facial landmarks in *ARnatomy* arose at the end of September 2018, when the aforementioned decision to implement the C++ API for HPE was made. At the end of September 2018, the published ARCore version was 1.5.0 with no face detection at all. After several months of implementing and improving the C++ API for HPE, ARCore published on 15.02.2019 the version

1.7.0 [23] with integrated realtime face tracking. After short tests, the implemented C++ API became obsolete and the accurate HPE from ARCore was used for anatomy model synchronization.

4.3.2 ARCore

The use of ARCore `AugmentedFaces` to render synchronized facial anatomy solely in the FA brought several advantages being the removal of additional NDK, Application Binary Interface (ABI), `cmake` configuration, third-party dependencies, various translation and conversion malfunctions. As the project build changed with the usage of ARCore, the *Android* Studio errors and abrupt terminations were gone as well as a decrement of application compile time, application start time, and latency was experienced. ARCore caused also a reduction of application file size by 150 MB, the overall source code amount, and the source code complexity. With ARCore, wider face tracking angles are supported, *ARnatomy* has better orientation capabilities for the user, and an overall higher robustness to tracking loss. The aforementioned improvements through using ARCore as HPE software increased in total detection accuracy, usability, User Experience (UE), and immersion.

The use of ARCore `AugmentedFaces` brought additional advantages as the usage of z axes values, realtime computation, complexity reduction, a well documented API, and wider face detection angles. If a small region beside, below, or above the eyes is visible on the screen, a face can already be recognized by ARCore [22] as depicted in Figure 4.3.

In Figure 4.3, screenshots after the immediate facial recognition of ARCore [22] starting the initial z axes translation of anatomy models out of the user's face. The rendered anatomy in Figure 4.3 obscures the face intentionally to illustrate the extreme angles and the detection of partly visible faces. In Figure 4.3, the top right image indicates that ARCore [22] relies heavily on eye location as the face detection was only successful, if the eyes were visible in the camera stream. Normal single AR model rendering with the mobile device's back camera is supported and the back camera usage has no ARCore HPE support. Repercussion is that the ARCore session configuration has to be changed accordingly with impacts on future design decisions. If body pose estimation [133, 134, 135, 136, 137, 138, 139, 140] may be supported, a dedicated body pose estimation algorithm and HPE algorithm is required. The front camera usage comprises proximity to the user's own body and therefore limits the capture of only upper body parts, e.g. the user's own head or throat, and not the whole body due to the arm extension range of the user. The arm extension range limitation of users makes the implementation of native full body pose estimation in the near future from the ARCore development team highly doubtful. Nevertheless, the application case of capturing other individuals with the back camera exists in the future and should be considered in future improvements of *ARnatomy*.



Figure 4.3: Top Left: Left and right lateral partial face occlusion requires one half of an eye for detection. Top Right: Top lateral partial face occlusion requires both eyes to be visible for detection. Bottom Left: Top partial face occlusion through head tilts requires both eyes to be visible for detection. Bottom Right: Bottom lateral partial face occlusion requires both eyes to be visible for detection.

Some source code of ARCore [22] is proprietary, so it can only be assumed what technology or algorithm ARCore is using for plane detection and head detection. Based on the high performance and environmental learning capabilities of ARCore, an adapted version of monocular Oriented Features from Accelerated Segment Test (FAST) and Rotated Binary Robust Independent Elementary Features (BRIEF) Simultaneous Localization and Mapping (ORB-SLAM) 2 [64] with qualified feature point selection may be used. As treated in Section 2.1.2, monocular ORB-SLAM 2 [64] is the state of the art in constructing a map with tracking the camera's position in an unknown environment in terms of accuracy. The map data is provided and updated via the ARCore API, so a developer can use the generated map of horizontal or vertical planes to build a program and to render an AR scene with AR models.

The plane detection with the back camera and HPE with the front camera exclude each other as other algorithms have to be used for practicable outcome, detection accuracy, and detection speed. With the front camera, the HPE of ARCore allows a developer to

use three predefined locations to position models or define a texture for the synchronized 3D face mesh. The usage of face mesh positions may be unwanted by the ARCore team. The reason for the latter assumption is that the documentation concerning the 3D face mesh had no explanations, references nor a picture of the 468 vertices from the face mesh in its first publication [23]. A description of the retrieval from single face mesh vertex coordinates was additionally missing. The coordinates could be retrieved by using the correct, uncommented ARCore methods. As guidance and help for the own development and for other developers, a *Github* repository `augmentedFaceMeshIndices` [132] was created to visualize the ARCore [22] 3D face mesh from different perspectives with its vertex numbers. One generated example picture [132] from a ARCore `.fbx` template file is depicted in Figures 4.4.

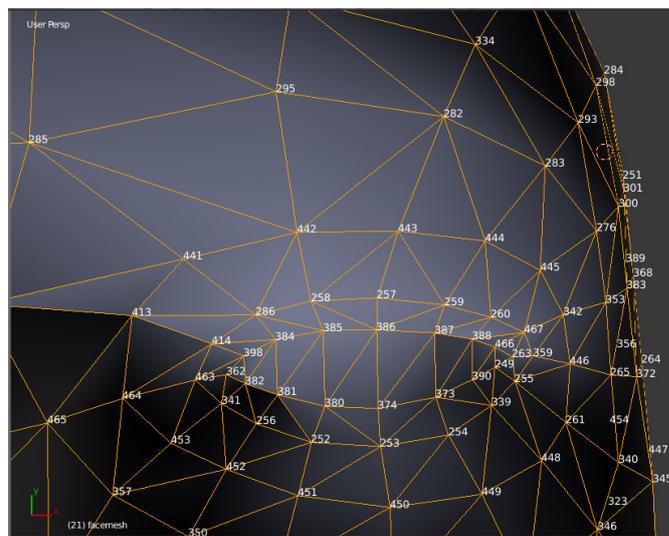


Figure 4.4: Right eye mesh indices of an ARCore `.fbx` file [132].

An own Two Dimensional (2D) texture can be mapped onto the mesh data from the ARCore class `AugmentedFace` delivered every frame by the ARCore facial detection. The `AugmentedFace` mesh data covers the face correctly, gets updated each frame, considers eye lid openings, mouth deformations, and facial expressions. Only assumptions can be made about the algorithm used for HPE. The HPE algorithm may be an adaption of the *Cloud Face API* [166], which is capable to detect all information about faces in an image needed for HPE.

4.4 Development and Design Choices

The general development and design choices for the application are treated in Section 3.4. In this section, specific software development and design choices for coding emerged during the implementation, which refine the general development and design choices.

4.4.1 Software Design Patterns

The simple software design pattern integration is easier [42] than more complex software design patterns [43, 44], because of the predefined code structure by *Android* projects and the own API of ARCore. The complex software design patterns are more specific and not designed for ARCore, which would result in unnecessary, bloated Boilerplate code making the source code structure brittle with loose ends. Simple software design patterns have a wider area of application, can be customized for special requirements, and still, they result in a clear code structure. Structured code with smaller classes than big classes furnishes maintenance, refactoring, advancement, and interchangeability of the BE. Understanding and adapting the relation of simple software design patterns to build a more complex one as the asynchronous DB API was time intense due to frequent source code refactoring. The anatomy is the key object in the *ARnatomy* source code and utilizes the wrapper pattern [42]. The source code object of anatomy, depicted as Unified Modeling Language (UML) class diagram in Figure 4.5, links DB data in the BE, its internal application status, and its visualization in the Front-End (FE), where the methods of each field are excluded for focus and simplicity.

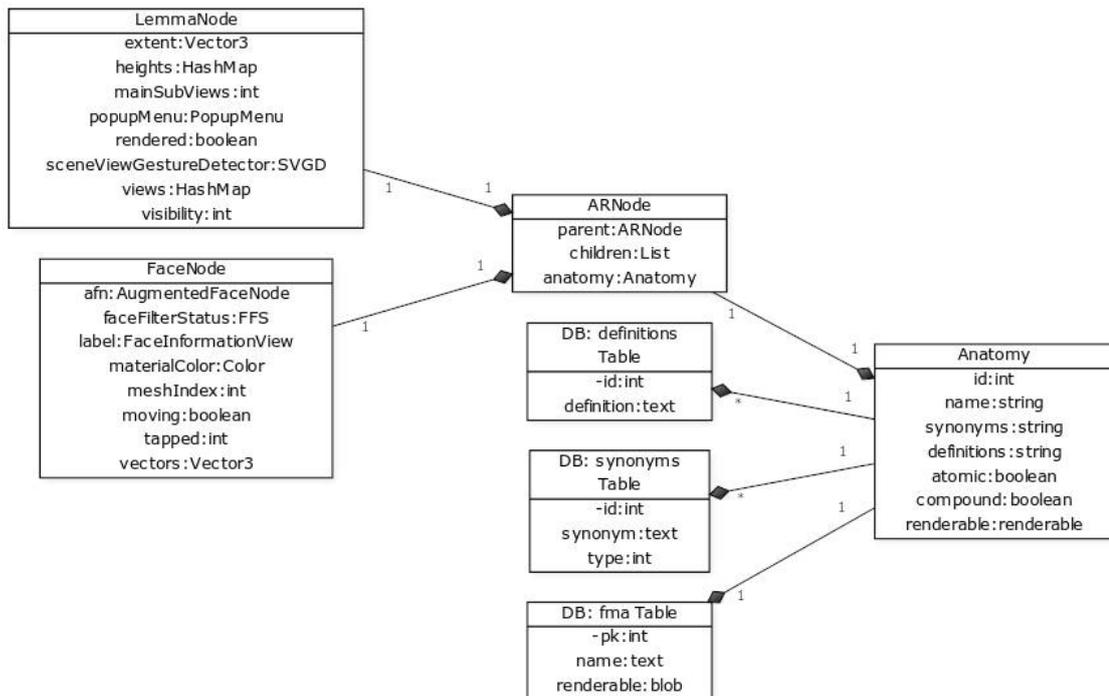


Figure 4.5: UML class diagram without methods of the Anatomy class and its relations.

In Figure 4.5, the anatomy object filled with data during runtime from the three DB tables is filled only by the required DB table entries depending on the current application status. The wrapping [42] node class ARNode, and its two subclasses LemmaNode and FaceNode of the anatomy object, contain render information, e.g. face filtering status,

hierarchy, already loaded or rendered. The `LemmaNode` is used by the Hierarchy Activity (HA) to present the anatomy hierarchy and the `FaceNode` is used by the FA to render the facial AR anatomy models positioned by HPE executed from ARCore. If more APs are added, the `ARNode` or `FaceNode` are the super classes to extend and their `Anatomy` fields should only to be lazy loaded. The lazy loading is highly important for any kind of hierarchy views, which should only load few levels or load the levels one by one for a low DB transaction volume.

Loading one hierarchy level from the DB can require the loading of 20 or more children, which can contain a various number of models. In the following, an example as illustration of the necessity of constraining the anatomy hierarchy loading to one level at a DB query. A DB entry is depicted by its ID and its name, e.g. *58238 left cornea*. All anatomy ancestors of the anatomy leaf entry *58238 left cornea* starting from the left with its topmost parent are listed and each FMA entry with a `*` has arbitrary other of descendants:

```
67135 anatomical structure*, 82472 cardinal organ part*,
14065 organ component*, 82485 organ component layer*,
58101 layer of wall of eyeball, 58102 fibrous layer of
eyeball, 58238 cornea, 58240 left cornea.
```

The above illustrated *58238 left cornea* hierarchy is also depicted in Figure 3.19 with an opened popup menu. Loading the overall *58238 left cornea* hierarchy at once would cause an enormous heavy query and a very long, unnecessary loading time. Any query from the DB loads only one anatomy hierarchy level.

4.4.2 Source Code Documentation and Principles

The documentation consists of *javadoc* comments on every package, class, and method as well as single line comments inside the source code in important or critical sections. The intention of the documentation is to briefly explain why and sometimes how the method or code was written for later, future developers reading the code including the author of this master thesis. The readability, understanding, error search, maintainability, purpose, specifically of complex code, is facilitated through source code documentation [167]. All the previously stated attributes of documentation require less effort and time to improve and extend *ARnatomy*. Quality rises with documentation as less time and effort in the future has to be spent to work with the source code [168]. Documentations about relations and crosslinkage throughout the whole application drew a more precise mental picture in the head of the coding developer and fostered effectiveness and efficiency with clarification. A developer has to read few documentation sentences before he inspects the often uncommented code, knows fast what to expect from a source code. The expectation gives the developer the chance to write more nuanced and more sophisticated source code than without documentation [168]. Misunderstandings are decreased though a first Point of Reference (PR) for understanding, which the documentation gives, and the mental picture about code is less abstract and more precise. The number of multiple code frag-

ments performing the same actions is decreased with the susceptibility to errors and the search for similar code is enhanced. Each source code package, class, field, and method of *ARnatomy* has a *javadoc* for documentation. Despite the importance of documentation, companies, teams, or developers to not write a brief description or expectation about the behaviour and result of their source code. As an example, the retrieval of ARCore’s single face mesh positions was overlooked on the first read, due to no source code documentation.

The API documentation of ARCore is only available online, see ARCore Issue 733 [169], and not in the Integrated Development Environment (IDE) of the source code where it is required the most [167]. Several coding principles were applied during the development. The Keep It Simple, Stupid (KISS) principle for focusing and trying to incorporate simplicity in the development design, to enhance the performance of the single, simple code parts, and to avoid code complexity, was applied to the *ARnatomy* source code. The Don’t Repeat Yourself (DRY) principle [167] aims to avoid redundancy and foster code reuse for an uniqueness of actions in source code fragments and was applied to the *ARnatomy* source code. The fail-fast principle [165] was merely used for debugging and testing. An application using the fail-fast principle reports errors as fast as possible and performs status checks during its runtime to detect malfunctions quickly. The infamous ninety-ninety principle was applied unintentionally too, where “*The 90 % of the code account for the first 90 % of the development time. The remaining 10 % of the code account for the other 90 % of the development time*” [170]. The ninety-ninety principle from Tom Cargill [170] humorously emphasizes that the time effort put into source code development is often non-linear and requires often almost double the time, “double 90 %”. In *ARnatomy*, the majority of the facial AR rendering code required nearly the same time as all other code.

4.5 ARnatomy Rendering

The development and design choices in Section 3.4 were amended or redefined by the act of development. The BE and FE are both separated in three single activities, which are equal to single *Android* activities. All aforementioned three *ARnatomy* activities, FA, HA, and Single Activity (SA) in Chapter 3, constitute and categorize the application in its rendering, understanding, and source code. The preference of the word *activity* before the words *mode* or *status* originates in its close relation to the structure of the *Android* source code. In *Android*, the definition of *activity* is “*An activity is a single, focused thing that the user can do*” [171]. The three activity names are inspired by the word describing their purpose, relation to *ARnatomy*, and the source code the most.

Within the source code of the three activities, software design patterns contributed to the implementation, the interoperability, and reproducibility of *ARnatomy*. Only applying the *Factory* pattern [42] to the *Android* SQLite DB API was insufficient to query anatomy fast and easily. Hooking is a technique used to alter or augment the behaviour of other

software components by intercepting function calls, messages, or events passed between software components. Code executing the described behaviour is simply called a hook [42]. The *builder* pattern [42] along with the *wrapper* pattern applied to the DB API facilitated the creation of customizable, asynchronous queries to the DB API with no FE latencies and no data or model changes through hooking methods [42]. The optional hooking methods are called when a query is finished. Each single invocation of a `build` method from the *build factory* creates a new object queried from the DB API.

The incorporated builder pattern and wrapper pattern inside the factory pattern provide a query customization and query abstraction, which allows the developer to query all table columns fast and with few code lines. With the abstraction, frequent query customizations can be grouped into a single query method according to the DRY principle. To be statically available with one instance and one global point of access, the *singleton* pattern [42] is applied too on the DB API. An adaption for *ARnatomy* is that the singleton is thread save to prevent concurrent DB malfunctions and exceptions caused by concurrency. Especially the hierarchy loading of multiple levels can cause multiple datasets to be queried multiple times. The hierarchy loading is restricted to one level for the most querying performance through SQLite. If somehow high latencies occur nevertheless, the FE remains unaltered as a consequence of the asynchronous fetching on the BE and exclusively updating the visible components when the data is loaded from the DB.

4.5.1 Face Activity

The FA, treated methodologically and previously in Subsection 3.5.1, renders interactive 3D AR anatomy models, which are synchronized through HPE with the user's face. The high level effects of ARCore were treated in Section 4.3.2. This section treats the low level effects and limitations of ARCore [22] and *Sceneform* [23] concerning the implementation of the FA. The ARCore team and the *Sceneform* [23] team prioritise the development on features as can be seen in the Appendix A.2. A disadvantage of ARCore is the need for model conversion to `.sfb` for rendering. The following paragraphs are important further development and the reproducibility, as they are influenced by properties and features of ARCore and *Sceneform*. Direct `.obj` rendering is unsupported by ARCore. The explicit reason why could not be found and remains unknown. The conversion to `.sfb` overwrites the coordinates of all single BP3D `.obj` files. The ARCore converter normalizes all coordinates based on the barycenter of the model, wherein the y coordinate is set to zero. The zeroing of the y coordinate comes from the standard case in ARCore, to position models on a plane. With command line arguments of the *Sceneform* [23] converter 1.8.0 or higher can be instructed to not recenter the model. The *Sceneform* converter 1.8.0 is malfunctioning on BP3D `.obj` files, which was reported as *Sceneform* Issue 613 web page [142]. *Sceneform* Issue 613 [142] limits the *Sceneform* converter to Version 1.7.0, which sets the y coordinate to zero without an option for disabling or enabling. As a consequence, all rendered anatomy AR models had to be positioned manually during

the whole development. The AR face root node determined by ARCore is located in the middle of the nasal bridge and all rendered models are positioned manually and relative to the face root node. The model accuracy is high, as the manually positioned anatomy models are relative to single 3D face mesh coordinates, which change with the detected face. Eye movements and jaw movements of users are supported and move the anatomy models too. Without *Sceneform* Issue 613 [142], the coordinates from the BP3D could be used to position all models automatically and relative to the face root node.

The manual, single AR model positioning due to *Sceneform* Issue 613 [142] was laborious and cumbersome. The correct anatomy model location had to be identified in the *Anatomography* [4] followed by a sequential approximation of model scaling, model rotation, and model positioning in that order for each model. Sometimes, during model scaling, model rotation, and model positioning, already positioned models had to be changed due to the anatomical context getting bigger and bigger, which made false model positioning noticeable. The bones were positioned first as the skeleton gives fortitude to the human body and serve as PR for other anatomy models. Eyes, brain, teeth, and vessels were positioned manually in that order and relative to the already positioned anatomy with guidance of the *Anatomography* [4]. Two major complications contributed considerably to the required time. First, the y axis zeroing of the models lead to a different position after a rotation around the model's barycenter being zero on the y axis but non-zero in the x and z axes as depicted in Figure 4.6. Second, some models needed a rotation around the x axis, so their front faces towards the camera.

Each rotation or position change of a model required an activity or application restart of five to twenty seconds depending on the amount of source code altered. If further APs are added, the *Sceneform* Issue 613 [142] persists, and no coordinates of new models are extracted by preprocessing, model scaling, model rotation, and model positioning have to be executed manually. As a matter of time economy, all alternatives and any form to automatically create predefined AR model positions should be carefully considered before positioning models manually. Anatomy models positioned too close to the ARCore face mesh detect no interaction because the face mesh consumes the intended interaction. Although, the invisible face mesh is located directly on the user's face and the anatomy models are rendered in front of the face mesh. Only the texture, material, and vertices from the utilized ARCore face mesh can be set or read. No other properties can be changed nor set having repercussions on the rendering and on the interaction. First, objects positioned behind the face mesh and not intersecting with the face mesh are not rendered to save computation resources. Second, changing the face mesh texture and its rendering priority causes unrealistic rendering results reported in *Sceneform* Issue 798 [172] and is therefore completely avoided.

Since the face mesh texture and its rendering priority must remain untouched by design in *Sceneform* as previously treated, rendering AR models as illusion inside the head is impossible with ARCore and *Sceneform* 1.7.0. Turning off the face mesh would remove

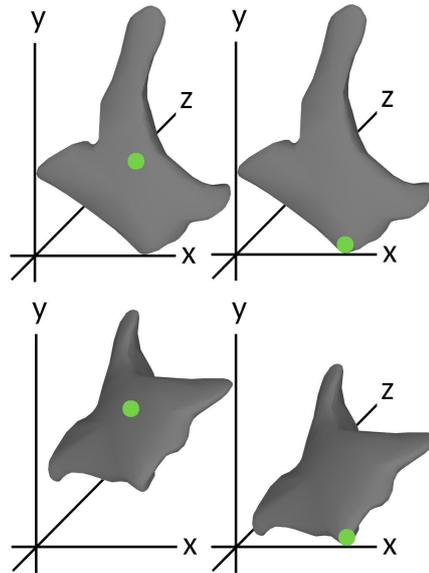


Figure 4.6: Left: Barycenter of the zygomatic bone visualized as green dot positioned in the middle. Right: Barycenter as green dot positioned at the bottom, with the y coordinate set to zero. Different positioned barycenters result in different model positions after a x axis rotation on the lower two images. Further rotations, e.g. on the z axis, would increase the position difference.

all face mesh coordinate tracking. A trade-off had to be made to overcome the face mesh constraints for rendering AR anatomy models with ARCore and *Sceneform*. The trade-off is a pinch gesture, which translates the root face node on the z axis. With the root face node, the residual anatomy translates too in or out of the head, which is similar to before and behind the 3D face mesh. The pinch gesture has a resemblance to zooming and depth translations in mobile device interaction [173]. Anatomy in front of the 3D face mesh is the focus area. Anatomy far in front of the 3D face mesh beyond the focus area is in the context area. In the focus area, the interaction with anatomy works as previously explained. In the context area, the anatomy is rendered transparently and does not react to taps. The overall anatomy can be translated completely in front of the head or behind the face mesh and is always synchronized on the anatomy x and y coordinates from the face mesh as depicted in Figure 4.7.

The control of the z axis is the control of the anatomy depth. So, the user can translate anatomy out of his head, in front of the face mesh, and dissect virtually via swiping the anatomy synchronized with the user's face. In Figure 4.7 in the middle, the anatomy is partially hidden directly behind the face mesh. All the 3D axes are always aligned with the face mesh. The face mesh is the PR for the orientation of all anatomy models. ARCore transparency has four limitations. First, the transparency rendering value going from 0.0 to 1.0 has no effect when set below 0.1. For example 0.05 or 0.01 have the same effect as 0.1. Second, the superimposition of front sides and back sides of single

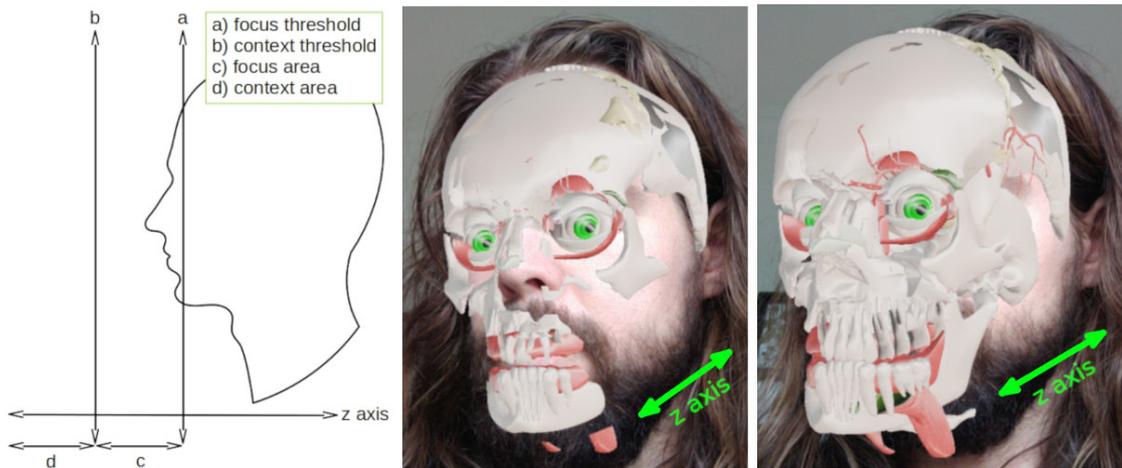


Figure 4.7: The z-axis translation controlled by pinch gestures viewed from a side angle for better representation. Left: Illustration of the thresholds from the focus and context areas. Middle: Back-translated anatomy in focus mode with obscuring face mesh on the cheeks. Right: Front-translated anatomy with anatomy, eyes, teeth, and zygomatic bone, in the context area and therefore transparent.

anatomy models summarizes the single transparency value of one anatomy object to 0.2. Third, with several anatomy models in the line of sight the aforementioned transparency summary is increased. For example two anatomy models sequential on the z axis have already an overall transparency value of 0.4 for all following anatomy models as each of the two anatomy models has a back side and front side. The result is less visibility for models located deeper inside the head. Fourth, the material transparency setting of ARCore suffers from poor design of *Sceneform* [23]. A single loading of materials is unsupported in *Sceneform* and confirmed by Issue 580 [174]. An extra `.obj` file would have to be loaded with special material transparent setting. But in combination with *Sceneform* Issue 613 [142] limiting the *Sceneform* version upgrade and the preprocessing of all models, an object loaded with special texture for all anatomy models is impossible. ARCore transparency cannot be used with its full potential in *ARnatomy*. If in the future higher ARCore versions than 1.7.0 are functional, the material setting has to include the `twoPassesTwoSides` and `doubleSided` properties. Then, the customized material can be used in the conversion of all BP3D `.obj` flat files to the *ARnatomy* DB.

External labels rearrange themselves with an adapted version of the technique of Mühlner and Preim [25]. On label generation, all four corner coordinates of a label emit a raycast to the camera. If no direct line of sight between the four label corners and the camera is given, the new generated label repositions itself downwards or upwards. The repositioning goes on until all raycasts of the four corners from the label hit the camera to guarantee an unimpaired line of sight to the label. An offset from the user's head ensures that all labels are rendered above the head. The label line is centered in the model the label refers

to and in the center of the label. The implemented four corner raycast test for labels needs considerably less computational demand than storing all rendered label positions, label properties, their anchor point computation, and distance transformations [25] as proposed by Mühlner and Preim [25]. The four corner raycast test achieves sufficient results as already depicted in Figure 3.11.

Internal labels [25] may be supported in the future and require a runtime generation as bitmap. The bitmap is then set as a texture of a model in ARCore or all anatomy names are precompiled textures loaded as Binary Large Objects (BLOBs) from the DB. Other approaches for internal labeling with ARCore are unknown. Problems in ARCore with internal labels arise, as the high variance of geometrical proportions in anatomical models makes a standard texture generation complicated. Not the whole anatomy name as internal label may be seen from the front. All the above properties with their mutual effects have to be considered for readable internal labels, i.e. text distortion and 3D shape fitting, anatomy orientation, and bumpy surface approximation via Bezier curves and their scaling effects [25]. Internal labels require further the consideration of their contour crossings with view-impairing anatomy shapes and overall perspective distortions [175] to be readable. An advanced algorithm, as treated by Ropinski et al. [175], is needed and must be implemented to function with the anatomy models loaded from the DB and ARCore. As external labels communicate the anatomy name in a sufficient way and require less implementation time, the topic internal labels remains for Chapter 7, i.e. future work.

The colour error affecting the HA and SA as depicted in Figure 3.5 may be linked to *Sceneform* Issue 613 [142]. Only the rendering of models in the FA with HPE has no colour errors. *Sceneform* versions above 1.7.0 cannot compile the BP3D models and *Sceneform* versions below 1.8.0 may set wrong material values not used in the FA. Howsoever, the *Sceneform* Issue 613 [142] and *Sceneform* Issue 764 [176] restrict *ARnatomy* to use the *Sceneform* model converter 1.7.0 and the *Sceneform* framework 1.7.0 or 1.8.0 but not higher.

4.5.2 Hierarchy Activity

The HA, treated methodologically and previously in Subsection 3.5.2, presents the anatomy hierarchy from FMA in textual form to the user through single hierarchy levels. The textual search of the user in the FA delivers in search results invoking on a tap the HA, which is equal to the previously treated long tap on an external label in the FA. All textual, hierarchical entries of the FMA can be viewed in the HA and the linkage to a BP3D model in the HA is optional. An empty HA, being no anatomy rendered, is forbidden and for safety, an empty HA circles the user back to the activity the user came from. In *Android*, a *shared elements* transition [177] is when User Interface (UI) elements shared between two activities have an animation. The animation is intended to illustrate a visual relation between the two UI elements. The HA and FA have shared elements and utilize the *shared elements* transition [177]. The same UI element in the FA

and HA is the anatomy name, which transits by smooth movement, scaling, and colour adaption to indicate the relation and equality of an element from a previous activity in the current, new activity. Coming from the FA, the initial anatomy has already its name loaded, all other anatomy properties are loaded, if they exist in the DB. The anatomy property loading ensures to show all information about an anatomy. Top down rendering of one anatomy parent or all anatomy children found in the anatomy hierarchy tree is executed by a recursive tree-traversing algorithm. The tree-traversing algorithm iterates asynchronously through the anatomy hierarchy stored on the DB to load one anatomy parent or all anatomy children.

4.5.3 Single Activity

The SA, treated methodologically and previously in Subsection 3.5.3, renders single interactive 3D AR anatomy models on horizontal planes or vertical planes for an isolated anatomy object view. ARCore uses metric data from the mobile device for support of its internal system and features. An example feature is an environmental High Dynamic Range (HDR) mode for light estimation since version 1.10.0 [144]. The environmental HDR mode comprises of three APIs to detect and replicate real world lighting by using the mobile device's back camera. First, the directional light from the environment is determined to cast shadows in the correct direction and on the rendered AR models for higher immersion. Spherical harmonics are special functions defined on the surface of a sphere for a realistic illumination. Second, ambient spherical harmonics for ambient illumination from all directions is available for the rendered AR models. Third, a HDR cubemap for specular highlights and reflections can be applied on the rendered AR models. Through environmental HDR, AR renderings become more realistic than without environmental HDR, as the illumination adapts to the real world, which increases the immersion. Regrettably the *Sceneform* Issue 613 [142] prevents any version upgrade above 1.8.0. Until the rectification of *Sceneform* Issue 613 [142], the disadvantages of lacking further bug fixes and no usage of 60 FPS cameras persist as well as less mobile devices are supported.

The plasticity of augmentations [51, 52] is the adaption of AR to its environment or the AR rendering status. The utilization of the plasticity of augmentations is constrained, e.g. *Sceneform* Issue 387 [178] disallows the application accessing the camera material to set the background colours of an AR model. For example, on close proximity of the camera to the AR rendering a blur or a black and white background could create low illustrative rendering [85, 86] as depicted in Figure 4.8.

To overcome *Sceneform* Issue 387 [178] and to have at least a black and white background, the direct usage of Open Graphics Library (OpenGL) shaders would be necessary. The usage of OpenGL shaders is reserved for future work due to time limitations, as only little documentation and information about the utilization of OpenGL shaders was found. A simple adaption of augmented plasticity [51] was implemented in the SA to communicate anatomy proximity via a red directional light shift. If a user comes close to an AR model

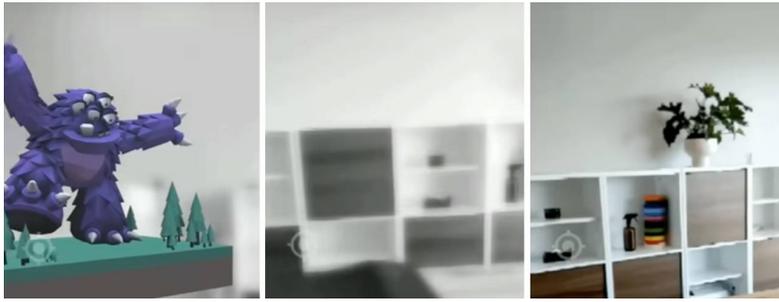


Figure 4.8: A left camera turn from the right over the middle to the left image showing the background colour change and blur due to AR model proximity [179].

rendered in the SA, the directional light turns smoothly to red to indicate closeness to the anatomy or being virtually inside the anatomy. The light shift is turned off by default, but remains in the application as small meaningless feature.



Results

Section 5.1 treats the development results, which is a mobile application with anatomical Augmented Reality (AR) model rendering synchronized with Head Pose Estimation (HPE). Section 5.2 treats the Informal Evaluation (IE) design and its result being the assessment of the implementation divided into two case studies and a subsequent interview to gain insights in usability and functionality. Section 5.3 treats the case study designs specifically created for the IE and the IE results created by the iteration of eight participants. Section 5.4 treats the summary of the single participant interview answers. Section 5.5 treats the limitations of the developed mobile application *ARnatomy*.

5.1 Development

Anatomy AR models paired with anatomical information synchronized with tracked user faces through HPE on a mobile device for the education of the general public was inexistent before. A user can view his face anatomy and information about the anatomy through synchronized, interactive, anatomical AR models rendered with ARCore on a mobile device screen. Each source code package, class, field and method of the 50 Java and 53 eXtensible Markup Language (XML) files has a source code documentation and is formatted for readability, which is reflected in Table 5.1 showing the number of written source code lines and documentation lines.

Language	Source Code	Documentation	Blank	Overall
Java	9134	1584	1652	12370
XML	1355	0	19	1374

Table 5.1: The line number of source code and documentation. Only *javadoc* comments are counted as documentation, whereby many source code comments are written too.

As the *ARnatomy* is written for *Android*, the *Google* guidelines and rules of *Google Play* [38], *Material Design* [37], developer policies [36], and Java [35] are followed. Following the aforementioned guidelines creates a high interoperability for *Android* devices and maintainability. *ARnatomy* can be executed on every mobile device supported by ARCore and should work on *Google Play*. Maybe the upload on *Google Play* as verification of guideline adherence will be conducted when this master thesis is graduated. On the mobile device, the application size is 380 Mega Byte (MB) unpacked and 118 MB packed. Packed refers to when *ARnatomy* is downloaded as well as installed but not started. The starting time of *ARnatomy* is two seconds and the anatomy loading time is three seconds. The relatively long loading time originates in the need to have all facial anatomy in memory for correct anatomy positioning as mutual anatomy dependencies exist. Only the permission to use the camera is required for rendering in AR. If the permission is not granted by the user, only the textual search with the Two Dimensional (2D) anatomy model will work. ARCore requires minimally an eye of the user's face to be visible in the mobile device's video stream for HPE in realtime as previously depicted in Figure 4.3. Sunglasses hiding the eyes seem to have no effect on HPE. Caps or hats prevent HPE, if they hide the eyes. Low illumination and high illumination extremes are supported as depicted in Figure 5.1.

5.2 Informal Evaluation

An assessment of *ARnatomy* with eight participants was executed by the guidance of Lam et al. [45], which surveyed 850 papers. From the identified 350 papers, seven scenarios for evaluation were elaborated via open coding. The seven scenarios were split into *process* and *visualization* classifications for an easier scenario selection. As the process in *ARnatomy* is fixed due to the created Database (DB) containing anatomy for ARCore, which has limitations previously treated in Section 4.5.1, the focus of the evaluation is set to visualization. Within the visualization classification, the User Experience (UE) scenario from Lam et al. [45] was chosen. The UE scenario applied in this master thesis is that *ARnatomy* is evaluated by participants, who received the *Nokia 7 Plus* with *ARnatomy* installed, to start discovery-based learning [27] of anatomy. The selected evaluation method is IE, it lets the user interact naturally with the application and the interaction observations as well as the answers from later asked questions, listed in Appendix A.3, create an assessment. The assessment contains information of intuitiveness, usability, functionality, knowledge transfer of *ARnatomy*, and design flaws. Two case studies were designed for the IE and all eight participants finished the two case studies and answered the asked questions afterwards.

The participants received only a clarification about the application context and about the anonymity of the IE. The participants could play and discover the application without any instructions or comments for about one minute. During the one minute, participants already showed a high degree of familiarization as the participants were already exploring anatomy. Afterwards the two designed case studies had to be executed and notes on



Figure 5.1: Top Left: Sunglasses have no effect on HPE with anatomy translated backwards for illustration. Top Middle: Low illumination right before HPE. Top Right: Low illumination right after HPE. Bottom Left: Cap just not hiding the eyes. Bottom Middle: High illumination right before HPE. Bottom Right: High illumination right after HPE.

the User Performance (UP) were taken. During the case study execution, only brief instructions were given what to do, but not how to do it. An exception with an immediate notation was, if a participant could not perform a case study component within 30 seconds. The intention of the brief instructions was to observe, if a participant can orient oneself and explore *ARnatomy* along with the contained anatomy. The brief instructions aim to create a more intense experience and a stronger familiarization with *ARnatomy* than explaining what to do as well as to receive more qualitative data [180] from the interview. Predefined questions utilized from and inspired by Lam et al. [45] along with specific ones originating in the moment of case study execution were asked and are listed in Appendix A.3. Specific questions about application details emerged

during the IE were carried on to the next participants to address details, to optimize the interview, and to increase the response quality [181]. All recordings of the participant’s answers were transcribed as facilitation for analysis, structuring, and fast access to the verbatim, open-ended answers [181]. The two case studies with the interview afterwards were done sequentially by eight voluntary, anonymous participants on different days but with the same *ARnatomy* version. Contextual participant data concerning the overall IE is listed in Table 5.2.

#	A	G	Profession	M	MR	CSC	CSM
1	23	f	Graduate Psychiatric Nurse	150	Entertainment	04:13	03:15
2	30	m	Production Manager	30	Social Media	04:51	03:04
3	30	m	Warranty Case Handler	180	Entertainment	03:47	02:51
4	25	m	Fitter for Machines	120	Info Search	05:37	03:28
5	30	f	Psychology Student	120	Instant Messaging	04:46	03:37
6	26	f	Office Administrator	180	Social Media	05:02	03:40
7	63	f	Pensioner	2	Calling	08:22	07:26
8	56	f	Home Helper	60	Work	07:48	06:07

Table 5.2: General information about the participants. A = Age, G = Gender, M = Minutes a Day on a mobile device, MR = Main Reason of mobile device Usage, CSC = Case Study Cerebellum Completion Time, CSM = Case Study Mandible Completion Time.

The mean participant age is 35, the mean age of female participants is 39.5, and the mean age of male participants is 28.3. The mean daily mobile device usage is 105:25 minutes, mean time for the case study cerebellum is 05:33 minutes, mean time for the case study mandible is 04:11 minutes, and it was tried to recruit participants from various professions to gain as verbatim results as possible.

5.3 Case Studies

The two case studies are designed for the IE [45], wherein the UE is assessed by the resulting data. The resulting data is expected to give insights and feedback on the user knowledge gain, usability, user satisfaction, knowledge transfer, possible design flaws, or design weaknesses. The two case studies are designed to let the participants go through all sections of *ARnatomy*. By going through the two cases studies, participants familiarize themselves with all application capabilities and get a detailed impression on learning anatomy with AR. As treated in Section 2.1.3, AR affects education positively [26, 75, 27, 76] and the exploration as well as interaction with the anatomy models is expected to construct anatomical knowledge. Similar as the task analysis in instructional design [46], the two designed case studies are subdivided into components to determine the actions needed for the completion. A case study component groups the actions a participant has to perform to finish a case study for a better structure and overview during evaluation. The single components of the two case studies are treated in the two

following Subsection 5.3.1 and Subsection 5.3.2. The similarity to instructional design is that participants get single, brief instructions by the observer describing sequentially one case study component to complete without any further help. The reason of withholding help is to observe, if the participant can perform self-directed learning [46] and discovery-based learning [27] with *ARnatomy*. The brief instructions contained only what to do but not how to perform a case study component. The intention of only saying what to do is to create a more detailed observation on the self orientation and self-directed learning of a participant within *ARnatomy* as well as a more experience-driven feedback about *ARnatomy* instead of explaining how to do a case study component. The brief instructions should also foster the anatomy exploration solely by the participant along with an intense effect on learning anatomy and the recall of the anatomy learning effect and anatomy content in the interview afterwards. All participants completed the two case studies in the same component order to observe the learning effects and to take the different times depicted in the previous Table 5.2. In the following two subsections, the single case study components are treated in detail.

5.3.1 Case Study A: Cerebellum

A fundamental understanding, knowledge, and feeling about the application and its structure should be learned with the case study *cerebellum*. From the standard, first Face Activity (FA) the participant should search for the *cerebellum* with the text search and tap on a search result to go into the Hierarchy Activity (HA). In the HA, the synonyms, description, and model are shown and hopefully read. Then, the parent and siblings of the *cerebellum* should be viewed by going one level up in the anatomy hierarchy. A sibling in the Foundational Model of Anatomy (FMA) hierarchy of the cerebellum is the *pons*. The *pons* should be started as Single Activity (SA) and enlarged. Then, the *cerebellum* or the *pons* should be found, but now on the own face in the FA as the participant should know now, how the *cerebellum* or the *pons* might look and where both are located inside the head, namely the brain. The case study is completed, if the participant was able to show the label of the *cerebellum* or the *pons* in the FA as depicted in Figure 5.2. In comparison to non-AR applications, the experienced immersion, including the positive affected learning, of the user when visiting internal anatomy structures is higher.

Participants needed approximately one minute longer than the *mandible case study*, since the *cerebellum* case study was their first encounter with *ARnatomy* and AR as well as the main case study shows the core functionality of *ARnatomy*. The following paragraphs treat the single case study components, their purpose, and the participant's interaction performance.

Component Text Search: With the textual search of the *cerebellum* the participant should gain knowledge of the text search capabilities of the application and, if the *cerebellum* is unknown, that the *cerebellum* exists in the first place. The same green colour for all User Interface (UI) elements used in this component, Floating Action Button (FAB) search, text field, and search result grid items, should form a mental picture that



Figure 5.2: Left: Labeled *cerebellum* from the side view for a better recognition. Right: Labeled *pons* from the front.

green is an indicator for interaction. The anatomy models react always on interaction as depicted in Figure 5.13.

Younger participants below 40 years of age recognized the textual search through the FAB icon in the FA. Older participants above 40 years of age had to search for the correct FAB via tapping. On a second try after knowing where the search is, all older participants immediately knew the search FAB. Only the oldest participant required more time to realize that the search results above the textual search are shown as a grid and can be tapped. The delay in the realization was caused by a preference of swiping over tapping. The oldest participant could not specify why swiping was preferred over tapping.

Component Hierarchy Activity: The participants should remember the HA as place giving extensive information about anatomy models. The name, the synonyms, the descriptions, and the model of the *cerebellum* are all visible on screen at the same time without any distractions keeping the cognition in one mental picture. In general, perseverance in the same mental picture facilitates the linkage of all observed and learned information, strengthens concentration as well as attention. The strengthening concentration and attention only applies with a lack of distractions, i.e. transitions or animations of disappearing or appearing content [33]. As constructivism states [13, 15], “*Active learning constructs spontaneous knowledge*”, which is hopefully given when translating between the single activities. Not the full potential of constructivism can take affect due

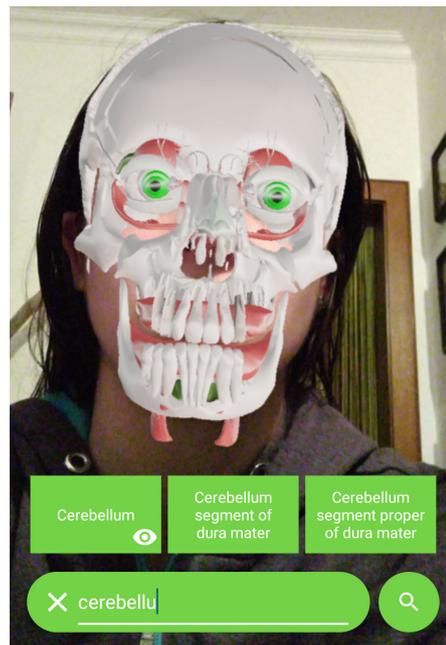


Figure 5.3: Text search with the automatic suggestions for the *cerebellum*.

to the 2D model rendering in the HA, its limited interaction, and no high immersive embodied cognition.

The effect of constructivism and embodied cognition on learning is higher in later case study components with AR renderings due to the higher immersion [26, 27, 75]. The HA with the *pons* as sibling of the *cerebellum* is depicted in Figure 5.4. Similar to the search icon, the menu icon in the FA was unknown by the two oldest participants. The same applies to the menu icon in the HA, but not for the hierarchy up navigation or down navigation as the arrow icons for up or down were known by the two oldest participants. The hierarchy view expansion on each invocation, depicted in Figure 5.4, fostered curiosity in the participants familiar with mobile devices. Questions showing further interest emerged and explorations happened to see “*What anatomy is up or down?*” One participant thought going up in the hierarchy displays the other search results previously viewed as a grid in the FA. On immediate and further investigation, a correct conclusion was made that the viewed content is the anatomical hierarchy.

Component Single Activity: The immersive rendering of the *pons* in the SA should show the full immersion to the participants with all aforementioned benefits of immersion on learning [26, 27, 75]. The participant starts the SA via a popup menu entry in the HA named *Single View* as already depicted in Figure 3.18. As in the previous HA, with a tap all information of an anatomy can be viewed as depicted in Figure 5.5. Via swiping the anatomy can be translated or rotated. After some interaction, knowledge about the

5. RESULTS

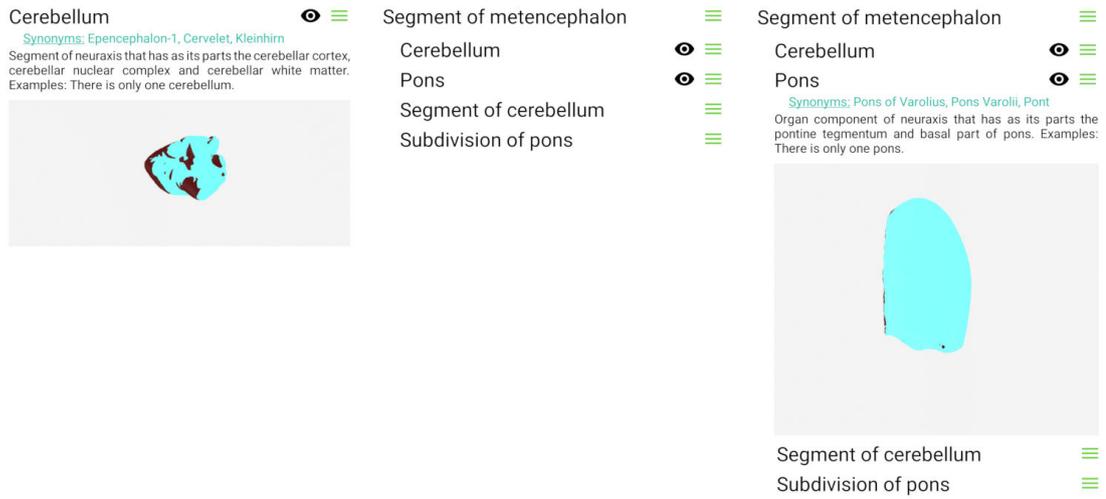


Figure 5.4: Left: Standard anatomy view of *cerebellum*. Middle: Hierarchy expansion of one level. Right: Selection of *pons*.

appearance of the *cerebellum* or the *pons* should be gained and one of both should be found therefore in the FA. With a sub menu entry of the menu FAB, the FA containing the textual search should be restarted.

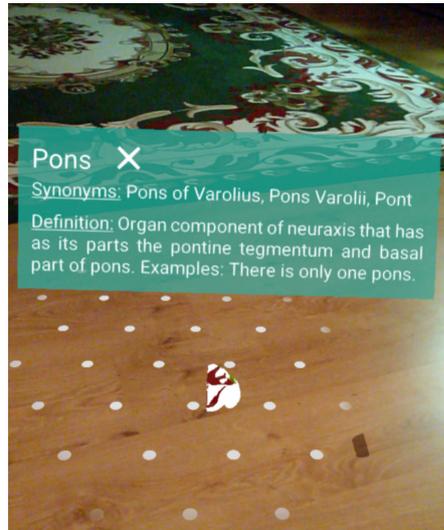


Figure 5.5: *Pons* AR rendering in the SA with its label activated.

Viewing single AR in the SA by coming from the HA was always understood, it took some time to find the icon in the popup menu as the application and its UI was new to the participants. After some seconds, each participant was able to go from the HA to the SA. The hand motion picture of ARCore moving clockwise to indicate mobile device

movement, already depicted in Figure 3.20, is rendered first in the SA, which confused all participants for a moment. Five participants figured out what to do, some with fast success under four seconds and some needing up to 13 seconds until ARCore detected a plane and rendered the anatomy. The required time is mainly caused by ARCore itself, as uniform surface textures with low entropy are poorly detected. As the five participants started to capture wooden surfaces, “*How do I get this to work?*”, the plane detection by ARCore worked significantly faster. The distance to the surface may have additional influence on the ARCore plane detection, most participants succeeded when not capturing a wooden table but a wooden floor. The better detection of ARCore with a wooden floor is a disadvantage, since a wooden floor is not always available. The search for a better detection distracted the participants from learning the longer a search was required.

All participants knew from the initial familiarization of one minute that a tap on the single rendered anatomy shows its label. A positive surprise effect was experienced by all participants in the SA, since swiping relocated the anatomy instead of removing it as in the FA. One participant relocated the anatomy from the wooden floor to dark tiles, “*for a better contrast*” between the foreground colour and background colour. The shader colour error already treated and depicted in Figure 3.5 was perceived by some participants, but the error has been considered as minor. The overall appearance of anatomy was always observable by the participants. An often stated comment by the participants was “*interesting colors*”, which encouraged more movement with the mobile device around the AR rendered model. Another participant stated amused during pinch gesture interaction for zooming “*The zooming is fun, can I zoom into the cells too?*” On the instruction “*Go back to the FA*” two strategies were chosen, tapping the *Android* back button until the activity is reached or pressing the dedicated menu button of the menu FAB as depicted in Figure 5.6.

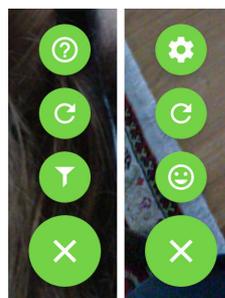


Figure 5.6: Left: The menu FAB invoked in the FA with the options *about*, *restart*, and *filter*. Right: Same FAB menu as left but invoked in the SA with the options *settings*, *restart*, and *face*. The settings allow the user to turn labels or plasticity on or off and go to *about*, showing brief application information.

The back tapping with the *Android* back button occurred only if the menu button was not investigated beforehand. However, the association of the face icon with the FA was

not strong enough. Half of the participants who tapped the face icon stated they have tapped the face icon because they were in the process of exploration or the first position of the icon in the popup menu and not because of an association of the face icon with the FA. The other half of the participants associated the face icon with the FA.

Component Anatomy Search: With tapping to see the external labels [25] showing the anatomy names and swiping to add or remove anatomy to the cache list previously treated in Subsection 3.4.2, anatomy can now be explored superimposed on and synchronized with the participant's own face as AR. The highest effect of constructivism and embodied cognition on learning is adhered through the most immersive UE [33, 75] the application is capable of. The direct interaction with AR models superimposed on the participant's own body lets the participant learn more about his anatomy [13, 15, 29] when searching for the *cerebellum* or *pons*. The exploration and search should be enhanced by a high illustrative rendering [85, 86] technique, namely transparency. Transparency alters the objects appearance allowing the viewer to see more content as less superimposition are observed. When an external label is rendered, all other unlabeled anatomy becomes a transparent context as depicted in Figure 5.7.



Figure 5.7: Highlighted anatomy during search for *cerebellum* or *pons* with its label shown. All other anatomy is transparent.

Back in the FA, the interaction to find now the *cerebellum* or *pons* was hesitant. More confidence came with more interaction on anatomy, sometimes the intended tapping had no or wrong outcomes, but all participants managed to recover and execute their intentions. Half of the participants remained on the outer anatomical regions and needed a hint, “*Can you go deeper?*”, to further try interaction and to discover the pinch gesture

for z axis translation. No labels of bones were viewed, but after going deeper into the head anatomy during their self-directed exploration, participants tapped on anatomy to see the anatomy's labels. The oldest participant was unsure and needed the most time and some encouragement that nothing can be broken or done wrong. After the encouragement of the oldest participant, more confident and faster application exploration was observed. Although the initial restraint on swiping of the oldest participant, the swipe gesture was known and performed after some time "*To enlarge the small anatomy*".

The screen size caused sometimes wrong swipe interaction results during the search for anatomy. When performing the pinch gesture on top of multiple anatomy models to translate the z axis, the gesture may be detected by multiple anatomies as a single swipe. A locking mechanism is implemented, when more than one interaction happens at the same time, no other actions can be invoked. However, the problem is the pinch gesture start, one finger may touch the screen sooner than the other, where the first finger invokes the swipe action and the second finger locks then further actions. Camera location has a big influence on interaction, some participants tend to stretch their middle finger, annular finger, and little finger when they are tapping, swiping or making gestures. The camera occlusion through a stretched finger happened at least once for six of the eight participants. Stretched fingers may occlude the camera automatically deleting the ARCore scene graph by deleting the root face node and all its descendants. An instant face detection is performed afterwards and the anatomy is rendered again from the start with the loss of the application status. Therefore, the distraction from learning is rather low, as a new anatomy exploration can be started in one second.

Component Completion: The *cerebellum* and *pons* are chosen to not have an immediate exploration result and to create a small challenge for the participants. The participants hopefully discover and learn a lot of anatomy due to the challenge of finding internal anatomy. Any kind of anatomy filtering is intentionally excluded in the *cerebellum* case study to make the experience of anatomy exploration more intense without than with anatomy filtering. Any swipe across any anatomy moves the anatomy into the cache list where the participant can scroll through the added anatomy. Cached anatomy can be put back into the face via swipe over a model. The external labels [25] invoked through a tap identify unambiguously an anatomy object. A label invocation of the *cerebellum* or *pons* is equal to the completion of the *cerebellum* case study. All participants discovered a lot of anatomy during the exploration and the interaction with models along with visible interest but not a single participant discovered the *cerebellum* and only one participant discovered the *pons* through model interaction. The positive effect of the *cerebellum* and *pons*, which were difficult to find, was that a lot of other anatomy was explored and found "*Ah that's where this is*", which is also reflected in the interview answers. After about one to two minutes of anatomy exploration, the case study was considered to be completed as the participants performed repeatedly, precise, and confident all interactions required for discovering the *cerebellum* or *pons* as depicted in Figure 5.8.

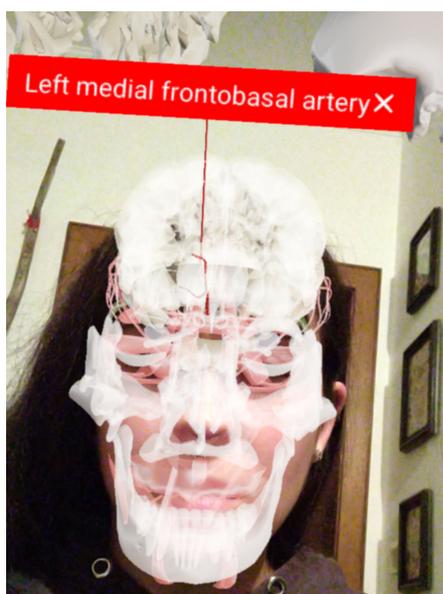


Figure 5.8: Anatomy is located above the user's head during the exploration to find the *cerebellum* or the *pons*. A label of the currently viewed anatomy is shown additionally.

A lot more time would have been required to discover the *cerebellum* or *pons* in most cases. The location of *cerebellum* and *pons* was too far inside the head causing the participants to get lost in the details of head anatomy.

5.3.2 Case Study B: Mandible

The participant's knowledge gained in the former case study should be enriched and amended in the sense of anatomy learning content in the mandible case study through more application details. From the FA the participant should filter all head anatomy but the mandible. To be sure it is the mandible, its label, depicted in Figure 5.9, must be invoked to access through it the HA to observe all information about the mandible.

Then, the parent and siblings of the mandible should be viewed by going one level up in the anatomy hierarchy to know there is an anatomy hierarchy in the first place. A sibling in the FMA hierarchy of the *cerebellum* is the *vomer*. The *vomer* should be started as FA through the popup menu entry *View Face* with the *vomer* label shown, which completes the mandible case study. The intention is behind going from one activity on another one through interacting with anatomy to show the user the different accessible Anatomy Perspective (AP). Participants need approximately one minute less time than for the *cerebellum case study*, since the mandible case study was their second encounter with AR. The interaction with *ARnatomy* and the UI of *ARnatomy* was known by the participants, since some mandible case study components were already visited in the *cerebellum* case study. The second case study shows the additional functionality of *ARnatomy* not visited by the participants in the *cerebellum* case study. The following paragraphs treat in



Figure 5.9: The external label of the mandible with unfiltered face anatomy for comparison with filtered face anatomy in Figure 5.10.

detail the single case study components, their purpose, and the participant's interaction performance.

Component Anatomy Filter: From the standard, first FA the participant should enable the face AR anatomy filter view and filter out everything but the mandible. By unchecking all checkboxes except the anatomy region *oral* or *mental* and anatomy type filter *bones*, the filtering is complete. The participant has observed the change in anatomy rendering according to the set filter selection as depicted in Figure 5.10. Now anatomy filtering should be learned by the participants to learn simultaneously the different anatomical terms through filtering.

Six of the eight participants were able to find the filter function at the first try under eight seconds without any hints, the oldest participant did not know the filter icon, did not try to find the filter function through exploration, and required a hint to proceed. The second oldest participant found the filter function through clicking all menu points. Solely the English anatomical region terms in the filtering depicted in Figure 5.10 caused problems of comprehension. Strictly the anatomical region terms but nothing more was translated by the observer into German for a correct understanding of the participants. The AR checkbox interaction worked only about 50 % of the time, double and tripple tapping had to be done to uncheck all the single filter options. The poor checkbox interaction of 50 % results, that the checkboxes react only when their text is clicked and not the boxes themselves. A technical solution to the checkbox problem could not be implemented in AR as rendering checkboxes in AR results in the text-selection behaviour. In general, the participants performed light and fast taps with later normal taps and

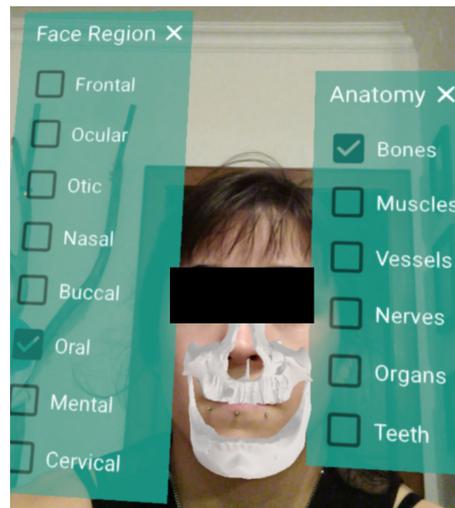


Figure 5.10: Filtered anatomy except the oral region to only show the mandible and maxilla. The filter panels for anatomy region and anatomy type are partially visible beside the face. Eyes are occluded for the participant’s anonymity.

hard taps. A normal tap or a hard tap is easier recognized by ARCore and the gesture detector. An observation focus on light tapping, normal tapping, and hard tapping was set on the residual participants, which showed an adaption of their tapping interaction with a slight annoyance during the first performed hard taps.

Component Label Access: Displaying the anatomy external label [25] of the mandible via a tap is the next step to validate that the viewed anatomy object is the mandible. A label in the FA shows only the name and not the anatomy synonyms or definitions due to the narrow space. Via a long tap on the label the participant should go to the HA to view all the anatomy data at once.

Seven of the eight participants knew how to display the label of the mandible immediately when instructed, the oldest participant required about five seconds to perform the tapping. Some tried a double tap, which has the same result as a single tap.

Component Vomer Search: By tapping the *up* entry of the popup menu in the HA, the parent and all the siblings of the mandible are loaded. One sibling is the vomer, another bone the participant has to find.

Half of the participants discovered or remembered the long tap function after some seconds in the FA to view the labeled anatomy with all anatomical details in the HA. The poor memorization or discovery rate of the long tap means that the interaction text shown once during database extraction is insufficient to communicate all interaction methods. A method more akin to the utilized major learning theory constructivism and embodied cognition [13, 15] would be beneficial to memorize the possible interaction methods better



Figure 5.11: Label of the filtered mandible. Eyes are occluded for the participant’s anonymity.

and was proposed as improvement by three participants during the interview treated in Section 5.4. Searching for the vomer was quickly done by three participants, all others needed about five to ten seconds to tap on the popup menu *up* entry to see the vomer hierarchy entry. The tapping on the *up* entry of the popup menu was faster than the first one in case study *cerebellum*.

Component Completion: To complete the mental picture benefiting from constructivism and embodied cognition [13, 15], the participants should now transit back to the FA via the popup menu entry *View Face* of the vomer. With the click on the popup menu entry *View Face* the case study is considered to be completed as all other three processes, explained after this sentence, are automated. The first process goes from the HA to the FA with its initial animation on the first recognized face. The second process applies the correct filters in the application to make the selected anatomy from HA visible in the FA. The third process renders the external label [25] and all other anatomy objects as transparent to highlight the vomer in its context via high illustrative rendering [85, 86].

Some participants acted so quickly, the next instruction being “*Show the vomer in the face view*” could not be spoken before other actions as loading new hierarchy or going to the SA or even going to the FA were executed. By knowing the HA from the first case study, more confidence and speed was clearly observable from all participants. One participant did not want to put the mobile device away, since he wanted to find the hypothalamus first, which he did with the words “*Is it really that small?*”. In general, the participant’s observation depicted high interest in AR anatomy. The anatomy exploration

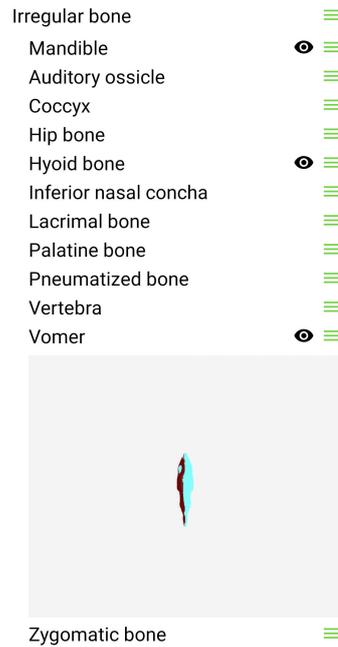


Figure 5.12: The hierarchy view of the mandible, showing the target anatomy, the vomer.

and the anatomy interaction came naturally after a short familiarization time without any instructions during the initial play time. Synchronization of the mandible and the eyes, which moved during speaking, head tilts, or phone tilts enhanced the immersion “*Wow, the anatomy on the screen moves when I speak*”. Some instructions had to be repeated, since some participants had all their attention focused on *ARnatomy*, especially in the FA. As soon as an activity or action was executed, six of the eight participants remembered them and could easily execute them again. Only the two older participants needed more repetitions, two to three, to achieve activity and action memorization.

Some insecurity among the participants was observable, they seem to think what can happen, what to expect, or if they will do something wrong before an interaction. A instructional tutorial about the interaction with *ARnatomy* may align their expectations to the understanding of *ARnatomy* and lower insecurity and concerns to do something wrong. The information provided from the FMA seemed to be too little for the participants. Some participants appealed to be dissatisfied instead of neutral or satisfied to have learned something about anatomy after reading the short FMA descriptions in the HA. The observation focus on the participants satisfaction with anatomical text information was caused by the phrase “*Somewhat sparse information here*” of one participant.



Figure 5.13: The highlighted label of the vomer in the FA coming from the HA. As the colour is orange, the highlighting is hard to detect right below the tip of the nose. Eyes are occluded for the participant’s anonymity.

5.4 Interview

A focus was set on open questions, as it is possible to analyse the open-ended answers with few participants. Open questions can have a higher response rate than closed questions and can generate qualitative data by design as well as give additional insights [180]. Professional blinkers is the blindness for mistakes or defects that occurs in the area in which one works due to long-term affiliation. Open questions allow the interviewee to bring up details, topics or issues hidden behind professional blinkers of the interviewer [180]. The participants can speak their minds on answering open-ended questions and say how they preserve as well as experience the application [181]. The interview answers are recorded for unimpeded, free, easy, natural answering from the participant without the cognitive load of writing and deceleration of writing compared to speaking.

Specific questions important for the application evaluation were close-ended followed by an open-ended question, e.g. “*Did you learn something about anatomy? What did you [not] learn?*” or “*Can the interaction with the application be learned? Why?*”. Evaluation goal is to gain insight about the participant’s perception of *ARnatomy*, their opinion on the usability, the applications capability of anatomical education, and qualitative data to answer alongside other information the two research questions. General demographic data about the participants is already included in the previous Table 5.2 for a brief overview of all participants. The asked questions are listed in Appendix A.3. The answers of the participants from the transcript are in the following paragraphs summarized exemplarily cited in English in the order the questions were asked. All participants were unfamiliar with AR and particularly with synchronized, interactive, anatomical AR model

renderings. Unfortunately, no participant familiar with AR or AR anatomy participated. Only short-term learning effects without medium-term and long-term learning effects were tested in the IE. Six of the eight participants were interested in learning anatomy. The oldest participant who uses the mobile device only for calling people had no interest in anatomy nor the application, since “*It is much easier to search in the Internet and no extra application is required*”. The graduate psychiatric nurse stated that no new knowledge can be gained for herself as she has more detailed anatomical education than the application provides.

However, the nurse’s final opinion along with six other participants on the interaction and the face-synchronized AR anatomy models was consistently positive. Important to note is, it was the first experience for all participants with synchronized AR, which was perceived as interesting, modern, memorable, and immersive. The first impression and experience in minute one was unsure and insecure but curious, because the application interaction and the application capabilities “*Can I move the anatomy?*” had to be discovered. The last impression was amazement that *ARnatomy* exists paired with elaborations for improvement summarized in later sentences to have the participant’s critics structured.

The *ARnatomy* application as a whole was interesting for six of eight participants. The oldest participant stated that the developed application is not interesting at all, but added “*in all fairness*” that most other mobile applications are too not interesting at all as “*They are too complicated and need too much time for understanding*”. Another participant explained that she would not install an application to get information, considering that anatomical information is easily retrievable from the Internet in her mother language, especially due to the medical terms. When the language barrier within the developed application would be inexistent, her first argument would apply with the addition that searching medical Internet sites seem more reliable than the one in *ARnatomy* and seem to contain verified and official medical information. On the question why information on the Internet is considered to be verified the answer was that medical information sites contain information from medical and professional lexica, which are verified and peer-reviewed by professionals. *ARnatomy* was not able to gain the trust of the participants. The assurance of medical correctness is missing inside the application.

Six participants can imagine *ARnatomy* in schools as learning tool for learning anatomy with less abstract pictures and more concrete imagination and understanding about anatomy in the pupils minds. Laymen interested in anatomy or having any anatomical issues may have benefits from using *ARnatomy* too, since the layman can look, which anatomy is propably hurting. By accident, such a laymen was a male participant, but to his pity, his hurting neck muscles were unsupported. All participants could imagine the application used as visual aid in a medical consultation operated by the a member of the medical staff. One participant imagined the application for students in their first lessons on universities and used the phrase “*digital hologram*” for AR. Maybe in the future, real

holograms are utilized for education in every class room. The oldest participant stated, if someone else, a doctor for example, would control the AR application to show her affected anatomy, it would be good for spatial understanding and the anatomical context, which can be expressed on a poster too but not as comprehensible as in *ARnatomy*. One key requirement for the oldest participant was inevitable, a bigger screen due to “*Seeing so much stuff in such a small space hurts in the eyes*”. The participants expressed an understanding, lightness, simplicity, and intuitiveness of the interaction with *ARnatomy*. The participants stated that common interaction methods are used, since there are “*Not that much interaction possibilities*” in the first place, and the interaction is consistent through the whole application. After an interaction method discovery, e.g. the swiping gesture, one knows how to redo the interaction for a desired reaction or invocation of an action. Through the minimal, non-complex UI the an interaction could easily be remembered and learned, especially the transition between activities. The discovery of some interaction methods was not as clear and intuitive to the participants as the discovered interaction itself.

The graduate psychiatric nurse and the home helper stated elderly people are unable to learn all the interactions, which was confirmed by the performance of the oldest participant. All interaction methods were known by the oldest participant, but her unconcern inhibited the interaction execution. The nurse with year-long experience in geriatric medicine explained the unconcern of most elderly people as follows; “*With the age, locomotory and cognitive restrictions adjust in various ways strongly influenced by the lived lives and the environment elderly people have lived in and live today. When people today in their twenies are going to be elderly, their interaction abilities in general will differ a lot than today’s elderly people, not only technical but social too*”. Small children can also not use the application for the same reason as for elderly people the nurse stated.

Three participants communicated, with more exploration and familiarization under maximum 15 minutes, the whole application including the transition between activities or rather their relation and logic can be learned. In the SA, the models are rendered on detected planes so user’s can walk around and inspect the anatomy. Walking around an AR rendering with the mobile device was considered as an interesting, new, and innovative interaction method by all participants, which is considered to be beneficial for the future of AR and *ARnatomy*. Textual information about interaction methods is insufficient to communicate all interaction methods at once. Six of the eight participants portrayed some kind of an initial tutorial teaching once step by step all single interaction methods with texts and explaining the single icons used in *ARnatomy*. Critic on the interaction by four participants was that some taps were not detected, but they recovered by tapping again harder. Swiping may hit accident wrong anatomy, which is annoying and was experienced multiple times by six participants. One participant expressed the desire to turn the whole anatomy in the face so the anatomical back of the head is rendered in

front of the face with a twist gesture. Currently, the anatomy can be viewed from the front and from the side in the FA, adding a view from the back would be highly beneficial for a better understanding. In the HA and the SA, the anatomy can be viewed from all sides. The oldest participant would have needed “*More structure in the application*”. Despite the previously mentioned interaction errors, all participants considered the usage of the common interactions as tapping, swiping, and gestures positively and with the words of one participant, interaction “*Needs some small polishing and it would be perfect*”.

All participants except the graduate psychiatric nurse learned something new about anatomy. The gained anatomical knowledge of all different participants ranges from the location of the pons, vomer, sphenoid, nasal bone, and cerebral artery to general impressions, e.g. the optical nerves end in the middle of the brain and that “*Wow, the brain is far more complex than I thought*”. All participants had the opinion that they will learn more about anatomy by conducting more exploration and that the learning applies for individuals of the general public too. The participants remembered what they have learned after approximately 10 minutes during the interview by explaining what they have learned. The anatomies remembered by the participants are depicted in Table 5.3.

#	A	G	Profession	N	Learned Anatomy
1	23	f	Graduate Psychiatric Nurse	0	-
2	30	m	Production Manager	3	Optical Nerves, Pons, Hypothalamus
3	30	m	Warranty Case Handler	3	Sphenoid, Vomer, Nasal Nerve
4	25	m	Fitter for Machines	2	Frontal Gyrus, Maxilla
5	30	f	Psychology Student	2	Nasal Bone, Zygomatic Bone
6	26	f	Office Administrator	1	Cerebral Artery
7	63	f	Pensioner	1	Frontal Bone
8	56	f	Home Helper	3	Frontal Bone, Vomer, Cerebral Artery

Table 5.3: Juxtaposition of the participants and their learned anatomy. A = Age, G = Gender, N = Number of learned anatomy.

The table 5.3 shows the anatomy memorization during the two case studies with a mean time of 05:33 minutes for the case study cerebellum and a mean time of 04:11 minutes for the case study mandible. The overall mean time is 09:44 minutes and only through interaction, the mean number of remembered anatomy by the participants is 1.87. The participants were only instructed to search for anatomy, but not to learn anatomy. Excluding the graduate psychiatric nurse, who had already detailed knowledge of human head anatomy and learned nothing new, the mean number of learned anatomy by the participants is 2.14. The mean number of remembered anatomy by female participants is 1.75 and the mean number of remembered anatomy by male participants is 2.6. Anatomy entities cannot be expressed as a real numbers with values behind the comma, so the overall mean number of remembered anatomies at 2.14 is therefore rounded down to have the definite result of two anatomies learned within the mean time of 09:44 minutes. Eight of the 15 remembered anatomy entities are bones, which are 53.33 %. Four from eight

participants would reuse the application to locate anatomy they previously have heard of. One participant stated he is unsure and would need more time with the application to receive more anatomical information first. Despite four participants would not use *ARnatomy* for themselves, all participants would recommend the application to another person interested in anatomy to try it. Reasons or the refusal of *ARnatomy* are that the anatomical knowledge is already high, no interest in anatomy at all, no trust in the contained anatomy information, or no interest in mobile applications in general. The appearance of the rendered anatomy is sufficient for five of the eight participants. Three participants would welcome more realism in the rendering, illumination adaptation, maybe photos as texture, more realistic shading, model surfaces, “*Nuances and more granulation as real bones*”.

An outline on each anatomy model or a colourization of all anatomy models, including an off and on switch as visual guidance for differentiation between the single anatomy identities, would be welcomed by five participants. The proposal of coloured anatomy models, anatomy regions or anatomy types [141] for differentiation is aligned to the findings from Garsa et al. [146]. Students who learned with coloured models instead of non-coloured models performed 10 % better in later tests [146]. The colourization could be paired with the single face filtering options too, as now especially the bones and brain gyrruses appear often as one entity, as two participants expressed. The current anatomy rendering was fine for the three other participants, who had the impression “*That this is how the anatomy really looks and one can easily differentiate it from another*”. One participant had the idea to colourize or outline the anatomy only if the camera is relatively close.

Three participants would welcome further anatomy to be tracked and superimposed. The whole body while capturing oneself in the mirror or others, arms, and hands as well as the teeth with information about correct brushing for kids and grownups were stated as beneficial. Two participants had an interesting idea apart from each other. The currently supported filtering options from the FA could be a menu to render not only single anatomies in the SA, but rather multiple anatomical structures. So all anatomy regions or anatomy types [141] would not only be filterable in the FA, but additionally in the SA. In the SA, the whole head anatomy without synchronization “*Maybe with an explosion view*” could be illustrated. Two participants stated more information about sicknesses or illnesses and anatomical relations in general are desirable. Single HA elements could be enlarged to fit the whole screen with a bigger text size one participant stated. Another participant would arrange the menu in the HA at the bottom, so the finger would have not to travel so far upwards to the first entry.

5.5 Limitations

In this section, only circumstances and properties limiting *ARnatomy* are treated. A list of workarounds and issues created during the development of *ARnatomy* is in Appendix A.2.

Only the full head anatomy AR with models from Body Parts 3D (BP3D) is supported, as ARCore only supports HPE. The HPE of ARCore allows only the front camera usage of the mobile device by design. The back camera of the mobile device can render AR too, but without HPE. Users tend to capture only themselves with the frontal camera as the arm extension makes capturing other scenes or objects cumbersome. If the back camera of ARCore would support HPE, the application cases would be extended, since posters, books, other renderings of humans, and humans could be camera-captured with synchronized AR anatomy for the face. For example, *ARnatomy* or AR in general may be used in museums, anatomical art installations or expositions similar to *Körperwelten* [182] to make the anatomy interactive and hopefully also more interesting, engaging, detailed, multi-modal, and innovative [183].

Mobile devices have naturally a smaller screen due to their mobility. Multiple HPEs in the FA are unsupported, because users captured by one mobile device would interfere each other. In the back camera video stream, horizontal and vertical plane detection time can be prolonged, if surfaces with poor entropy are captured. The front camera has a more robust detection than the back camera as already depicted in Figure 5.1. Disparities or clearly visible patterns on the floor as veins of wood, previously depicted in Figure 3.5, or tile joints accelerate the plane detection. During the development over several months, short beards, coffee cups and glasses before the mouth, reading glasses as well as eating had no effect on HPE as long as the user's eyes are visible. People unfamiliar and insecure with mobile applications will not be able to explore all features of *ARnatomy* and will perform self-directed learning [46] rather cumbersome, which reduces the learning effect. No special skills are required to use *ARnatomy*, only being familiar with common mobile application interactions methods as tapping, swiping, and gestures is required.

All other APs listed in Section 3.1 in identified by Preim and Saalfeld [13] require professional training for their understanding and are inappropriate for the education of the general public. Consequentially, other APs are out of scope for this master thesis. The selected macroscopic and regional APs [13] supported by *ARnatomy* limit the target group to the general public, pupils in schools, and patients having interest in macroscopic head anatomy. An example of a task may be that the pupils have to describe the anatomy filtered by the filter, which uses medical terms. Afterwards, a teacher explains the medical terms to the pupils to ensure the correct and complete knowledge transfer. People having professional training are advised to consult specialised literature for more information about anatomy, since *ARnatomy* is medical imprecise and not medically comprehensive regarding the contained textual information and detailed model information. The imprecision originates from the lack of FMA descriptions for all models and the exclusion of all BP3D models smaller than three millimeters, since the latter could not be identified correctly in the face. However, the general public can use *ARnatomy* for education of fundamental macroscopic anatomy and regional anatomy. The medical staff can use *ARnatomy* as visual aid to create a more fig-

urative macroscopic and regional mental picture about head anatomy in a layman's mind. The missing medical information in *ARnatomy* can be explained by the medical staff, who has professional knowledge. The usage of BP3D and FMA as a conflated *ARnatomy* DB imposes several limitations. The utilized flat file collection BP3D [4] contains 3.493 anatomical models, which is not the whole human anatomy in detail. BP3D contains a subset of the macroscopic anatomy and regional anatomy and over time, more anatomy is added by the authors. BP3D was the only free, scientific, and digital Three Dimensional (3D) anatomy model collection found. The linkage of BP3D to the FMA [5] DB is unidirectional as the FMA contains 100.000 anatomical textual entries.

Not the full head anatomy is included in BP3D, in Version 4.0 all veins of the head are missing, and, for example, in Version 4.3 only the right head veins are included. Logically related anatomy may be missing, e.g. the internal carotid artery is contained in BP3D and FMA, but parts of the external carotid artery are missing in BP3D. Symmetric anatomy having a left and a right part may contain only one of the two parts, e.g. the tentorium cerebelli misses its right half. The opposite is additionally included, symmetric anatomy can be unified as one model, e.g. the two ears are given as one model and both halves of the body are one big skin model making its usage for anatomical regions impossible. The best fitting ethnicity for *ARnatomy* is the Japanese ethnicity, since the models from BP3D are created from Japanese people. *ARnatomy* can be used on all human ethnicities, but superimposition discrepancies due to their different facial anatomical properties are possible.

FMA has a detailed hierarchy of the human anatomy and only the Identifications (IDs) and the names are persistent for all entries listed in the Appendix A.1. In the FMA, anatomy synonyms, anatomy definitions or their translation to other languages are optional for each entry. If an entry in the FMA has synonyms or definitions, they are always at least in English, which limits the application language support to English. All the above facts about BP3D and FMA constitute the medical imprecision of *ARnatomy*. Laymen speaking other languages may not know or understand the professional anatomical terminology in English. The future exchange of BP3D or FMA is complicated, since the some indices of BP3D vary in its different versions. New versions of FMA are not included in the current BP3D version or its newer versions. When using another version of BP3D, only the corresponding FMA should be used. Unfortunately, BP3D has a lower release rate of new versions than FMA. Due to *Sceniform* Issue 613 [142], the source code depends on FMA indices from the generated *ARnatomy* DB. A model DB update can therefore only be executed when another DB than BP3D is found. The requirement that all models of the newly found DB can be converted by the *Sceniform* converter must be fulfilled.

Another limitation is that ARCore needs `.obj` files to be converted into `.sfb` files to render them as AR. Because of *Sceniform* Issue 217 [158], no multiple `.obj` files can be converted with the *Android Studio Sceniform* plugin and the conversion of all BP3D files

has to be done via the command line converter of *Sceneform*. The `.obj` models from BP3D cause malfunctions in the *Sceneform* 1.7.0 [23] converter and in versions above reported in *Sceneform* Issue 613 [142]. The *Sceneform* Issue 613 has major effects on application development. The converter of *Sceneform* 1.7.0 normalizes the vertices of an `.obj` file to its barycenter with the y coordinate set to zero. An automatic positioning was impossible as all anatomy models differ highly in their appearance and therefore also in their correct alignment on the 3D axes towards the user. The overriding of the normalization of all BP3D model coordinates caused the need for a manual anatomy model positioning, which at least doubled the development time. Versions above 1.7.0 of the *Sceneform* converter allow the developer to disable the normalization, but have an error in the BP3D models.

A version upgrade of ARCore or *Sceneform* higher than 1.8.0 is impossible, because only ARCore and *Sceneform* 1.8.0 function with models converted by Version 1.7.0. *Sceneform* Issue 764 [176] adds probably independently from *Sceneform* Issue 613 an incompatibility of all *Sceneform* versions above 1.8.0 with the *Nokia 7 Plus* mobile device. *Sceneform* Issue 764 [176] has to be considered for further improvements. So, *ARnatomy* does not profit from error corrections, updates, and features in newer ARCore versions and *Sceneform* versions until *Sceneform* Issue 613 [142] and *Sceneform* Issue 764 [176] are resolved. Realism and immersion in *ARnatomy* are lower as they would have to be, inasmuch as the environmental High Dynamic Range (HDR) light estimation introduced in Version 1.10.0 cannot be used.

An assumption is that the colour error depicted in Figure 3.5 in the HA and SA is related to *Sceneform* Issue 613 [142]. The transparency is lower in models converted with *Sceneform* 1.7.0 as it would be in models converted with higher *Sceneform* versions also treated in Subsection 4.5.1. Newer converter versions support additional arguments controlling the normalization of all coordinates in the converted `.obj` files and support a more fine-gained transparency control. The plasticity of augmentations [51, 52] is limited to a diffuse light change to indicate anatomy proximity in the SA. The reason for the limitation of plasticity is *Sceneform* Issue 387 [178], which inhibits accessing the camera stream for blurring and changing the background colour to black and white on proximity to the mobile device. Blurred, black and white background colours would highlight the coloured AR model for more focus as previously depicted in Figure 4.8. *Sceneform* Issue 798 [172] prevents the rendering of AR behind the 3D face mesh, inside the user's head, from ARCore. Without rendering anatomy behind the face mesh, the illusion of looking inside the own head cannot be created. Instead, a workaround preserving immersion and AR is implemented via a pinch gesture. A user can translate the overall anatomy through the pinch gesture on the z axis in front of the face mesh and therefore also in front of this own head. The translation of anatomy on the z axis creates the impression that the user can see and dissect anatomy models synchronized with his own head. All the macroscopic anatomy and regional anatomy of the head contained in the BP3D with

all its names can be learned. Additional information about the facial anatomy is optional and depends on the entries in the FMA DB. The limited immersion due to *Sceneform* Issue 613 [142] and *Sceneform* Issue 764 [176] is considered to make the learning curve for *ARnatomy* more flat than without the two issues.

The positive effect on spatial knowledge acquisition [75], short-term spatial accuracy and short-term spatial speed in executing anatomical tasks is impaired. As a consequence, the learning effect *ARnatomy* accomplishes is lower as without *Sceneform* Issue 613 [142] and *Sceneform* Issue 764 [176]. The immersion is decreased by the two issues. Hence, it follows [16] that the learning satisfaction with AR is additionally lower than without *Sceneform* Issue 613 [142] and *Sceneform* Issue 764 [176]. Users having low spatial abilities have a significant disadvantage in interaction and learning with 3D visualizations [27] and should learn anatomy with other methods.

Discussion and Conclusion

Section 6.1 treats the development result, the effects on learning, and the evaluation result delivered by the Informal Evaluation (IE), consisting of two case studies and an interview, whereupon the two research questions are answered. Section 6.2 is the conclusion of this master thesis.

6.1 Discussion

Points for improvement and limitations were discovered through the IE and require future changes to enhance the interaction, immersion, and learning effect of *ARnatomy* as well as to create a more sharper mental pictures about anatomy. However, individuals of the general public can educate themselves about macroscopic and regional head anatomy through the developed, Head Pose Estimation (HPE)-synchronized, interactive, immersive, anatomical, mobile, Three Dimensional (3D) Augmented Reality (AR) application. Anatomical models and corresponding texts are utilized by AR to create an engaging, discovery-based [27] learning experience, confirmed by the IE, in the form of a mobile application named *ARnatomy* discussed in the following subsections. Subsection 6.1.1 discusses the insights obtained by assembling all utilized technologies during development into *ARnatomy*. Subsection 6.1.2 discusses the insights created by the IE. Subsection 6.1.3 discusses the effects of *ARnatomy* on user education. Subsection 6.1.4 answers to the two research questions.

6.1.1 Development

The three implemented activities, Face Activity (FA), Hierarchy Activity (HA), and Single Activity (SA), allow the user to learn macroscopic anatomy and regional anatomy [13] from three different perspectives. Macroscopic and regional Anatomy Perspectives (APs) constitute the fundamental knowledge of anatomy, where all other APs build upon.

Additionally, all other APs require professional training making them inappropriate for the education of the general public. With the found Body Parts 3D (BP3D) [4] and its relation to the Foundational Model of Anatomy (FMA), macroscopic and regional anatomy models and textual information about the models are conflated in a Database (DB) for *ARnatomy*.

Having non-replaceable third-party dependencies, i.e. BP3D and ARCore, can create limitations for the application. Especially, if older versions of the third-party dependencies ARCore and BP3D are partially incompatible, unique in their field, and newer versions of both technologies are incompatible as a whole. The uniqueness of ARCore and BP3D forces their usage. The conflation process of BP3D linked with the FMA enabled many optimizations making the *ARnatomy* DB size 42.52 % smaller than without optimization and fast to query on mobile devices. The dependency on BP3D and FMA decreased dramatically the flexibility of models as no easy nor fast DB change is possible due to the overall preprocessing time of one and a half days. At the introduction of new APs two decisions have to be made, i.e. add data to the generated DB or use a second, new DB with indices to the first one. BP3D does not publish full, new versions containing the `.txt` files required for compound BP3D model processing. Furthermore, BP3D does not link their models with newer FMA versions, which are published recurrently. In the future, only newer versions of BP3D containing indices to a FMA version can be used, which are rarely published. BP3D should be exchanged immediately with any other scientific anatomy model DB, which is free of charge, rich in detailed `.obj` files, and available. No pricing and free availability excluded other anatomy DBs during the writing of this master thesis.

Users appreciate up to date anatomical information [13], which cannot be visualized with *ARnatomy* right now. The dependency on ARCore, with its extension framework *Sceneform* [23], released independently from BP3D and FMA newer versions during the development until a version conflict with BP3D, treated in *Sceneform* Issue 613 [142], emerged. Until the rectification of the aforementioned errors, future anatomy models added to *ARnatomy* must be positioned manually and cannot be positioned automatically. A main task of future improvements of *ARnatomy* should be the automatic positioning of all 388 facial anatomy models, when a *Sceneform* converter above Version 1.7.0 does not fail on the BP3D models. As recommendation for future development, the documentation, practice-approved usage, age, and interoperability of a technology play a critical role too in software development. One may focus on the latter mentioned properties of technology during the search for a new technology and not only on the capabilities and features. Good source code documentation and interoperability of a technology can avoid a high time effort and several workarounds during the development. ARCore and BP3D were in their fields unique during the development of *ARnatomy* and had to be used together, despite their partial compatibility previously discussed. *ARnatomy* consists of three different anatomical activities, allowing the user to have three different APs on macroscopic anatomy and regional anatomy. In the FA, the user can interact with

HPE-synchronized AR models by tapping, swiping, and performing gestures for self-directed [46], discovery-based [27], immersive anatomical exploration and learning. In the HA, the user can observe all anatomical details of an anatomy entry from the generated DB along with the anatomy hierarchy to preserve and extend the generated mental anatomical picture [33]. In the SA, the user can investigate single rendered AR models on detected planes without anatomical context and with anatomical information for memorization and a different view. Additional textual information, which is also available in the SA as in the HA, for a better mental concatenation of learned anatomy is available too.

The FA is the main activity invoked when the application is started. The observation and interview answers of the IE suggest, that the synchronized anatomy models had the biggest impact on the eight participants in their first impression. The first interaction with anatomy followed naturally in a few seconds after the participant's observed the HPE-synchronized, rendered anatomy. Each user learned the name of previously unknown anatomies and their location by simply tapping on the observed anatomy and showing thereby the corresponding label. Similar learning results showing the potential of interactive AR anatomy as Bork et al. [184], Blum et al. [29], and Nuanmeseesri et al [185] were achieved. The participants responded positively and understood that the seen anatomy is synchronized with their own one. Some instructions given during the IE had to be repeated two or three times to be heard, since the participants were strongly focused on the content of *ARnatomy*. The concentration level of the participants suggests a high immersion created by *ARnatomy*, which possibly got increased through the realism of synchronized eye movement on head tilts and synchronized jaw movements. One older participant made an amazed comment "*Wow this is my head anatomy, ... hey, the jaw moves with mine!*". Participants familiar with human anatomy recognized the human skull immediately as basis among other rendered anatomy. The skull was considered to be a reference to other, internal, rostral anatomy, especially when participants removed anatomy through swiping across anatomy.

Swiping interactions on anatomy were self-discovered along with the invoked labels, the search, and the pinch gesture. Knowledge that swiping as interaction method exists or swiping by accident elicited the swiping on anatomy models. The participants were confident and seem to expect a swiping interaction method. Generally, older participants tried swiping on *Android* components, which are commonly tapped, e.g. popup menu entries. If *Android* supports swiping on a component and a swipe action has not already an action assigned, tap and swipe actions should be the same, so the user can perform the same actions through different interaction methods. The cache list contains anatomy elements removed from the face above the user's head in parallel and is a container holding the anatomy removed during the digital dissection and exploration. The name cache list, treated in Section 3.5 and depicted in Figure 3.13, was unknown by the participants, fast discovered, and the cache list's principle was immediately understood. The swiping across facial anatomy causes its translation above the user's head into the cache list.

Providing the cache list to observe *cached* anatomy and add anatomy back into the face had a positive impact on the users. Comments in the form of “*Parts are nicely ordered as they were removed*” about the cache list reflected its positive influence. The cache list anatomy has the same interaction as facial anatomy, i.e. a tap shows the anatomy name and a swipe invokes a transition. Various cached anatomy was tapped repeatedly out of curiosity to see what anatomy was removed from the head. The downward swipe to translate anatomy into the head from the cache list was discovered by all participants. Three of the eight participants from the IE discovered that left and right swiping scrolls the cache list left and right. For a clarification on cache list swipe interaction, an AR arrow should be positioned at the left and right screen edges, if the cache list extends beyond the screen due to its overall width. Tapping and swiping the arrows cause a cache list scroll in the corresponding direction to give users several interaction methods to achieve their intention. Depending on the anatomy width, up to five anatomy models fit into the cache list without going out of the screen, which irritated some participants for some seconds.

Only with tapping and swiping, the user is able to explore his own facial topmost rostral anatomy spatially and is able to learn the location, the context, and the name of anatomy. A pinch gesture as third interaction method beside tapping and swiping is required to explore internal head anatomy, i.e. rostral anatomy. As no tutorial, only a text informed the participants about the interaction methods, the pinch gesture was discovered by four of the eight participants. The assumption that the most participants will try or discover the pinch gesture as interaction method to translate anatomy rostral was wrong. The IE confirmed that a not-closable tutorial, discussed later in Section 6.1.2, undergone by the users on the first application start is required to teach the interaction methods. The other half needed a hint “*Can you go deeper?*” to try other interaction methods and people unfamiliar with mobile devices may not discover z axis translation through the pinch gesture at all. Furthermore, people unfamiliar with common icons used in digital content and on mobile devices may have troubles finding actions as the textual search, whose symbol is an outline of a magnifying glass. The two oldest participants needed several seconds to discover the textual search Floating Action Button (FAB), learning the three most important icons beforehand may lower the user’s unconcern through unknowing, which inhibits interaction.

6.1.2 Informal Evaluation Insights

One insight of the informal evaluation was the requirement of an instructional tutorial. The instructional tutorial should learn the user the interaction possibilities by requiring the user to do the supported interactions once at the first startup of *ARnatomy*. If the user gained interaction confidence through the instructional tutorial, the users’s cognition focuses more on anatomy learning and anatomy exploration instead of interaction exploration. The icon learning should be included in the instructional tutorial for an increased orientation of the user in *ARnatomy*. The information dialog failed to inform seven of

the eight participants about the interaction and the dialog lacked any explanation of the used icons. The lack of interaction knowledge was observed in the two case studies and highlighted by the answers of the interview. Important to note is, six of eight participants suggested some kind of interaction introduction without any specific questions concerning an interaction introduction or a tutorial.

At application start during the DB extraction in the background thread and instead of the information dialog, an interactive, not skippable tutorial could be run utilizing the previously described benefits of constructivism and embodied cognition [26, 75, 27]. Instead of reading interaction methods, users must execute the interactions to achieve a better memorization. The tutorial instructions should to be given twofold, i.e. few words as AR 2d view at the bottom screen edge show the user what to do with clearly visible graphical cues as depicted in Figure 6.1.

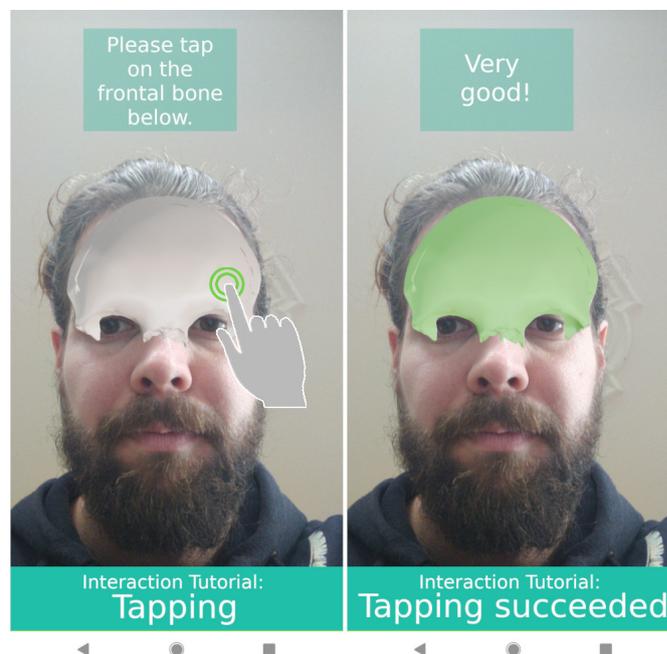


Figure 6.1: Created with image manipulation, to illustrate the initial interaction tutorial for the tapping interaction. Long tap, swipe, pinch, and icon tutorials should be highly similar.

Tapping, long tapping, swiping, and the pinch gesture would be visualized, as depicted in Figure 3.20, for consistency in the same style as the hand motion picture of ARCore, previously depicted in Figure 3.20. The user is forced to execute all interaction tutorials sequentially once after the installation of *ARnatomy*. All other interaction methods than the required one by the tutorial are turned off, so no false interactions can be made. The anatomical model used in the instructional tutorial is the frontal bone. The frontal bone

is a relatively large model and seen always on application start. When the instructional tutorial is completed, the user has at least one anatomical Point of Reference (PR), i.e. the frontal bone. The pinch gesture should be mediated at last in the tutorial to achieve the best memorization after the tutorial. The importance of the pinch gesture originates in its poor discovery by only four of the eight participants. The four other participants, which did not discover the pinch gesture, received a hint after 30 seconds. Users familiar with mobile device interaction would finish the tutorial in a few seconds, and unfamiliar users can take the time they need.

As “*active learning constructs spontaneous knowledge*” [13, 15], the memorization of the gained knowledge is more persistent through interaction than non-interactive learning or non-interactive knowledge transfer. Hence, the commonly used, most important icons in the application, i.e. the search icon, menu icon, eye icon, and face icon, should be learned too. The icon memorization can be enhanced by letting the user write the icon name written above or beside the icon in a text field. Familiar users would go fast through the tutorial and unfamiliar can take their time to observe and learn what the icons are representing. After tutorial completion, the DB extraction may be finished so the user would experience no or at least a decreased waiting time. The learning curve [186] is currently flatter as the learning curve would be with the tutorial. A juxtaposition of the expected learning curve and the current one as illustration is depicted in Figure 6.2.

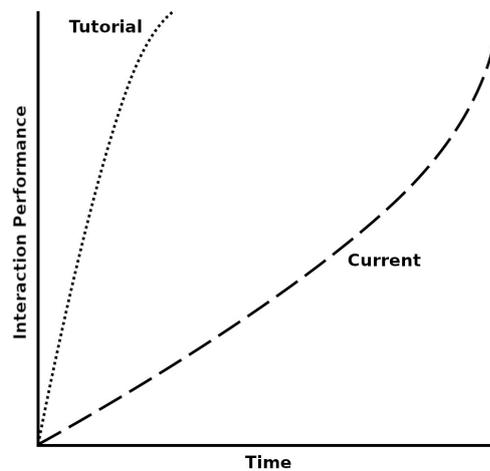


Figure 6.2: An assumption of the current learning curve based on the performance of the eight participants compared to the tutorial one. The tutorial learning curve is expected to be far more ascending due to the required interaction in the tutorial. The upper boundary of both curves marks the capability of performing tapping, long tapping, swiping, and the pinch gesture.

The learning curves depicted in Figure 6.2 are both an assumption, since different learning types exist and users may experience the instruction tutorial along with the preservation of the gained knowledge differently [186, 27]. Some of the interaction methods may have

to be repeated or rediscovered until they are remembered, but they would have been shown once at least and be known at best. By implementing the tutorial, the user is expected to learn faster the application interaction and has afterwards less cognition load from interaction as Pusic et al. [186] and Diegmann et al. [27] show. Küçük et al. [187] showed that less cognition load yields to more learning.

With a tutorial, more than one participant may have found the *cerebellum* and *pons*. The exploration direction of the *cerebellum* and *pons* was through the forehead slightly tilted downwards, highly sagittal and slightly transversal. Far too much anatomy was between the participants and the *cerebellum*, especially small anatomy contained in the diencephalon, all cerebral arteries, and sagittal nerves near the brain. The case study *cerebellum* was considered to be completed after one to two minutes as the interaction for target anatomy discovery was repeatedly performed on other anatomies. The participants learned through their exploration, although the case study completing anatomy, i.e. the *cerebellum* and *pons*, remained unexplored. Various other anatomy explored by the participants located in the sagittal direction not far inside the head, as depicted in Figure 6.3, would have been a better choice for a fast anatomy exploration.

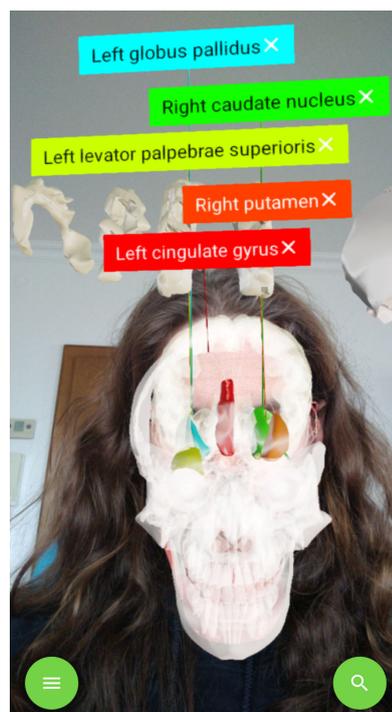


Figure 6.3: The coloured anatomy, with external labels shown, is best choice for search targets in the first exploration of a case study due to their close location to the forehead.

In future case studies, the target anatomy depth should be proportional to the number of anatomy already learned by the participant. Interesting would be another IE con-

ducted the same way with the same two case studies, but with the proposed tutorial implemented and the target anatomy set more rostral. All participants understood that the textual search results in the FA redirected them into the HA. During the activity translation, an animation of the anatomy name translating the name from the previous to the new location is displayed to preserve the created mental picture and extend but not replace the mental picture through more information for a better learning effect [33]. Title and optional synonyms, definitions, and Two Dimensional (2D) model rendering of an anatomy are displayed, if in the DB, to amend the mental picture of the user generated in the FA. The 2D model rendering in the HA should to be an additional visual verification for the user that the anatomy according to the users will is shown.

One participant suggested to put the entries in the HA at the bottom instead of at the top to have the anatomy in the reach of the fingers and to avoid stretching. The suggestion could be extended by evolving the hierarchical anatomy bottom up instead of currently top down as depicted in Figure 5.4. Avoiding a the stretching of fingers may have an effect on the User Experience (UE) too, since camera occlusions may be prevented more often. Mobile device camera occlusions in the face or SA are severe, since the scene graph is deleted automatically by ARCore and the application status is lost. The frequency of camera occlusions depend on the finger stretch of the user during the interaction. The immediate re-detection by ARCore of the user's face makes no difference for the application status loss. A solution would be to utilize the generated anatomy node structure as cache and to recreate the previous ARCore scene at the next face detection. But ARCore scene graph recreation along with application state preservation is complex and time intensive, and reserved for future work.

The upward and downward loading of the anatomy hierarchy in the HA is intended to add a hierarchical context and grouping to the created mental picture. All participants understood immediately the created hierarchy and the interaction possibilities through the popup menu. The items in the popup menu of an anatomy entry with the prefix *View* fostered curiosity and are intentionally ordered at the top of the popup menu. The immersive activities and views with their positive effects on learning [13, 15] should be accessible first.

Coming from the HA, the SA renders single, isolated 3D anatomical AR models on detected planes. On SA start, the hand motion picture of ARCore confused all participants as wrong surfaces prevented the plane detection. The reason for the plane detection delay was the camera-captured surface, uniform texture aggravates plane detection, i.e. wood, stone, or tile surfaces are well suited for detection. An information text or an animation, which suggest to capture textured files or distance changes to the surface, may accelerate the detection. As no other contextual anatomy is rendered in the SA, enough space is available to render the name, synonyms, description, and the model of an anatomy DB entry. The intention is to extend the generated mental picture of the anatomy model from the FA or HA with information the user does not know or wants to read again. In the

SA, the hierarchical loading as in the HA is unavailable due to the isolated view the SA is designed for. A conflation of the HA, depicted in Figure 5.4, and the SA, depicted in Figure 5.5, rendered as AR could increase the overall immersion and learning effect. The question is how to render FMA entries located in a complex hierarchy, e.g. 15 anatomy ancestor entries without any models hierarchical as AR. Scaling the hierarchy entries to a smaller size may be an option, i.e. the user would simply have to go closer to the rendered AR objects with his mobile device. Another amendment of the SA rendering may be the introduction of the currently supported filtering options from the FA. The filters of the FA could be in the SA menu to render not only single anatomies, but the anatomy regions and anatomy types as published by Rohen et al. [141]. The hierarchical relation could be visualized with the same lines from the external labels as depicted in Figure 6.3.

With the interaction knowledge from the FA, all participants engaged with the anatomy in the SA almost instantly. Swiping relocates the anatomy similar as in the FA, but in the SA the user has the full control of the anatomy location on the detected plane. A participant made an interesting comment on zooming into cells. Indeed, zooming may be a transition to enter another AP, e.g. a microscopic anatomy. The activity transition would have to be approved by the user before its execution to prevent unwanted perspective changes with a high computational demand. With the addition of microscopic anatomy, *ARnatomy* becomes more interesting to the medical staff. Laymen lack the knowledge to comprehend microscopic images or visualizations, but the medical staff may use *ARnatomy* as visual aid in the microscopic perspective of anatomy too.

All participants gave positive feedback about the application and its possible usage in the medical field as consultation aid operated by medical staff. Participants seem to sense that their interaction, or an interaction during a consultation, would require too much cognition and draw the attention, focus, and understanding away from the information given. Patients want all their attention available and unimpaired to understand their illness, its origin, and treatment during a consultation. Stationary screens in a medical facilities are bigger than mobile devices, draw less cognition to interaction, and would allow users with impaired sight to recognize more content. General anatomy knowledge can improve the information exchange between doctor and the patient, particularly after a diagnosis [1]. Successful exchange and communication between patients and the medical staff effects indirectly treatment success and compliance.

AR in general facilitates the presentation of anatomy affected by a treatment or by a surgery and the consultation quality rises the more synchronized the anatomical visualization is with the patient's body. The medical staff can give a more figurative explanation with the superimposed anatomy drawing a clearer, representative image in the patient's mind [1]. Rehabilitative measures after surgery have the most benefits of patient knowledge and awareness [1]. Indirectly surgeries may benefit too from using AR

during consultations, since knowledge and awareness form the basis of patient approval for the surgery. Unfortunately, in the last years the consultation duration decreases, but *ARnatomy* has the potential for a faster and more precise knowledge transfer [1] than verbal consolation to due to its AR anatomy illustrations in the same time than without AR. Familiarization with interaction techniques are required to use *ARnatomy*. The User Performance (UP) in the two case studies was in strong relation to the minutes daily spent on the mobile device. Derived from the data of the eight participants depicted in Table 5.2, the users require rudimentary mobile device interaction abilities and understanding about tapping, swiping, and pinch gestures to be able to educate themselves with *ARnatomy*. Derived from the observations taken in the two case studies, it seems that insecurity and concerns about interaction creates too much cognitive load inhibiting the anatomical learning. A high time investment and better familiarization may counterbalance the high cognitive load, but this assumption needs further confirmation.

All participants stated that after at most 10 - 15 minutes of exploration, the whole application interaction can be learned, which is encouraging for a longer and frequent usage of future users. The participants learned quickly from their mistakes and managed to recover from them. All participants completed the second case study faster than the first one, since several actions and components were remembered from the first case study. Nearly all participants knew immediately what to do when a case study component was accidentally already invoked or visited. The successful self orientation of all participants suggests no false mental pictures about the User Interface (UI) nor the interaction were generated throughout the whole application usage. The intention to keep the UI simple and only implement common interaction methods to let the user focus more on learning was fruitful. Because the last case study component instructions could not be spoken fast enough for some users, they thought the study was over and performed other actions in the meantime.

The IE is considered to be successful due to the insights obtained from two case studies and the answers from the interview with open-ended questions [180, 181]. Seven participants welcomed the anatomical visualizations with interest as well as excitement and could imagine that there is a positive effect on learning anatomy when the participants were asked about that. The positive feedback from the participants may come from no previous experience with AR [76] and the general interest in anatomy of seven participants. The participant opinions on AR were that AR is interesting, modern, memorable, and immersive. Only one participant had no interest in anatomy, rather in the AR technology and how he could view the stationary anatomy in the SA by dynamic camera movements. The positive feedback reflects the often reported higher user satisfaction of 3D visualizations compared to 2D teaching methods [75]. Only the oldest participant was slightly upset after the case study completion, since the exploration time was longer and more cumbersome compared to the other participants. The application was still considered to be beneficial for learning by the oldest participant, notwithstanding “*Rather for people keen for new technology*”. Many insights were gained through the IE. Each

participant had at least one behaviour, perception, view, critic, or statement, which obtained an insight having the potential to improve the mobile application. The two case studies made design flaws and interaction flaws visible as well as the interview depicted how the users perceived the application outlining detailed issues or future ideas. The IE results show that simplicity and minimalism in digital technology can be a double-edged sword. A non-complex UI and low application depth makes the interaction, orientation, and understanding simple. Too few PRs, known content or common content can cause concerns on how intentions can be accomplished. An insight obtained by the IE concerning user studies is that the latter help to find the transgression of necessary reduction as well as unnecessary extension.

The interaction of *ARnatomy* was not always clear and can be improved. Front camera occlusions created an unexpected restart. Most users did not know what they had done wrong as they focused what their thumb or index finger was doing and not were their other fingers were. A message for clarification may be displayed to at least give the user the possibility of knowing what happened. The exploration of interaction methods is not as immersive, nor exciting as the exploration of anatomy and information texts about interaction help little. If possible, users have to perform an interaction at least once, in an instructional tutorial for example, to be aware that certain interaction methods exist, especially, if multiple interactions should be learned. Similar anatomy renderings have the potential to be perceived as one anatomy entity and should be clearly differentiable. A mechanism for differentiation could lead to a better memorization of the learned content [79].

This paragraph treats general insights and ideas obtained by the IE. The interest in new technology, in this case AR, can persuade users unfamiliar with that particular technology to use an application, albeit no interest in the application's content is given. Interest in the content can then arise through the created curiosity on the first impression and through the mere application usage. Any kind of content verification, best from an institution, is beneficial for the user's trust in the application. One application may have several application cases, which are found best by asking multiple people and multiple users. In the future of AR, probably the fusion or even its relay with or through mobile, interactive, markerless holograms may be realized. The requirements of younger and older people on mobile applications should be considered in the software development. Similar to the results of Bork et al. [184], questions about the appearance of seen anatomy had an instant response from the participants. Seven of the eight participants could explain verbally what anatomy they have learned and memorized two anatomy structures within 09:44 minutes through interaction, without any instruction to learn anatomy. The graduate psychiatric nurse did not learn something new. During interaction, anatomy visited before had a high recognition rate, e.g. participants often said the name of an anatomy when they were swiping across. Letting the participants sketch the learned anatomy would have been interesting as an additional feedback and visual illustration of

the learning effect.

6.1.3 Effects on Learning

Macroscopic education is the basis of some medical studies and key for an advanced training of a surgical subject. Learning on and with the own body has significant benefits for spatial abilities [1]. Institutions for basic education as high schools can establish a more student focused learning with multi-modal AR applications including *ARnatomy* for head anatomy education. The anatomical spatial comprehension and learning are facilitated through the 3D AR renderings and the immersion [33] of the latter. Learning with AR extended with textual information is superior to 2D teaching methods but not to all traditional learning approaches [75]. In studies about education [75, 27] with 2D visualization learning groups and 3D visualization learning groups, the latter achieved the best results.

In the IE, participants less familiar with head anatomy learned that they are viewing the human skull and became, through their self-directed, ongoing exploration [46], more familiar with the head anatomy. The participants learned something about anatomy in the aspects of position, location, orientation, name, or complexity as the participant's answers to the asked questions show. The participants learned two anatomy structures within 09:44 minutes through interaction, without any instruction to learn anatomy, which suggests that the memorization of anatomy is created through active learning [13, 15]. The number of learned anatomy under ten minutes may improve when the application is used frequently over a period of time for a specific learning goal, e.g. learning for an anatomy test. A future study on learning a specific anatomy type or anatomy region with *ARnatomy* can be that participants learn the content 15 minutes a day for one week. To test the gained short-term knowledge, textual questions and hand-drawn sketches could be demanded from the participants on the last weekday. The textual questions and hand-drawn sketches could be repeated as test after one week and one month to test the medium-term learning effects and long-term learning effects too. However, the learned anatomy during the IE encompasses bones, parts of the brain, arteries, nerves, and the complexity of the brain. The high memorization rate of bone anatomy with 53.33 % compared to residual anatomy may come from the fact that bones are one of the topmost rostral anatomy types and contain some of the biggest anatomy entities. In the mean, female participants memorized 1.75 and male participants 2.6 anatomy structures. If the currently researched differences of the spatial abilities between the sexes [188, 189] had an influence on the learned anatomy remains unknown. Studies with hundreds or thousands female and male participants could maybe answer the question, if spatial AR learning is preferable for females or males.

A question is how long the participants remember the spatial knowledge. Most studies conducted and confirmed that AR benefits for short-term memorization [13, 27, 16, 77, 78]. Only learning mathematics with AR has significant long-term effects shown by Som-

merauer and Müller [78], but not in general nor for anatomy. Desirable would be a confirmed medium-term learning effect and long-term learning effect by learning spatial subjects with AR, e.g. macroscopic and regional human head anatomy. A future study with *ARnatomy* as mobile learning technology may bring new insights on short-term memorization, medium-term memorization, and long-term memorization. The researchers could focus on the study, its quality, its structure, and its execution, since the technology is already developed in this master thesis. As proposal, a Randomized Controlled Trial (RCT) [79, 190] study could be conducted with two groups for short-term, medium-term, and long-term, memorization. One participant group learns with *ARnatomy* the other group not with AR, but with traditional learning. A parameter in the proposed study could be the separation of subject matter into non-spatial and spatial to see the relation of AR to non-spatial and spatial learning in short-term, medium-term, and long-term memorization more clearly. Other confirmed and researched attributes than spatial AR may contribute differently to short-term, medium-term, and long-term memorization types. These attributes are augmented emotional engagement, increased user satisfaction, perceived own effectiveness, motivation, attention, concentration, creativity, interactivity, and a steeper learning curve [13, 16, 17, 26, 27, 28, 75, 76, 77, 78, 191, 192].

Learning with AR can have some undesired effects too encompassing justified criticism. Users having low spatial abilities have a significant disadvantage in interaction and learning with 3D visualizations [27] due to the increased spatiality of learning. But, if the users learn over a long period of time, approximately one year, students with lower spatial abilities can have significant higher improvements in understanding of spatial anatomy than students with a high spatial abilities [184]. For example, only the oldest participant struggled during the case study, but still managed to complete both case studies. If a learning inefficiency is experienced, one should not force oneself to learn with AR education tools, but rather switch to other learning tools or techniques. Facilitated spatial knowledge gain linked with information is the main advantage of AR, but many other non-spatial knowledge about anatomy exists too. In comparison to traditional learning, learning on a mobile device has the problem of lacking details [193], which cannot be counterbalanced by a longer usage. Longer mobile device usage would introduce even more problems, since reading hard copies is more comfortable for the eyes and causes less eye fatigue than reading on a mobile device [194]. In general, long and intense usage of any kind of computer with a screen causes eye fatigue [194], social exclusion, and social isolation [195], myopia in pre-school children. The total number of myopia patients increases steadily in the last decade [196]. In general, a message after one hour should remind the user to take a break from the continuous mobile device usage. Micro lessons [191] are an option too to prevent myopia, i.e. learning on mobile devices is implemented as small tasks and short readings in a multi-modal learning context, which contains the benefits of AR learning. The interaction should be as intuitive as possible, otherwise the cognition is drawn away from learning to interaction decreasing the positive effects for learning. Users may focus more on the novelty of the technology or new devices as instead on the displayed content [75, 79].

Other activities on the user's own mobile device may distract the attention fully or as short learning disruption, e.g. writing messages, social media consumption, sending pictures, searching information on the Internet, taking pictures or capturing videos. Non of the previous mentioned distractions was experienced during the IE. Learning independence is defined as “*The behavior of student in manifesting their wants or desires in a real way by not relying on others* [191].” The digital distractions have the potential to transform independent learning into doing something else independent from learning. Learning with mobile applications may require dedicated devices without installed distractions, time logging implemented to ensure that a certain time is spent on the learning application, if specific content is viewed or invoked. Students of the 21th century are keen on learning with the newest technology, which certainly can foster independent learning, if it appears intuitive and modern to the learners [191]. As some participants were highly focused on the *ARnatomy* content and instructions had to be repeated two to three times, it is assumed that *ARnatomy* has a high immersion. The high immersion of AR and the related concentration increase showed by Diegmann et al. [27] can lower digital distractions of users.

6.1.4 Answering the Research Questions

With the gained knowledge, data from the literature, and from the IE of *ARnatomy*, the two following research questions are answered in the next paragraphs.

1. *How can interactive AR be used in anatomical education for the general public?*
2. *How much and what anatomy can be learned in which time with the developed application compared to state of the art works?*

The answer to research question one is through any publicly available, mobile AR anatomy application, which synchronizes anatomical models with additional textual information in realtime on the user's own body parts. *ARnatomy* achieves a subset of the previous answer, since the aforementioned usability problems of *ARnatomy* limit its availability to science. Future improvements can make *ARnatomy* publicly available. The mobile device usage of the general public increases [8, 9, 10] and with publicly available mobile applications, the potential of public education increases too. Several recommendations were identified for beneficial learning properties of an education application of anatomy with AR. Beneficial for interactive AR is the usage of pose estimation to synchronize the supported anatomy region and its models in realtime. Anatomy synchronized with the user's own body or another camera-captured body enhances the positive learning effects of AR and increases immersion. The immersion is proportional to interest, motivation, eagerness, attention, concentration, and user satisfaction [26, 27, 75, 76, 78, 79, 183, 191, 193], which are important factors of learning.

Bork et al. [184] conducted an one year study with medical students including the relation of macroscopic and regional anatomy with radiographic pictures of anatomy. In questions concerning identifying macroscopic and regional anatomy in radiographic pictures, the medical students had deficiencies [184]. When medical students with an anatomical education have deficiencies in learning the relation of topographic anatomy of radiographic pictures, how can laymen of the general public learn radiographic AP? Further and other learning modalities are required to teach the linkage between macroscopic and regional anatomy and radiographic AP more efficiently. *ARnatomy* can only teach macroscopic and regional anatomy [13]. The difficult understanding of macroscopic and regional anatomy in radiographic pictures cannot be achieved by *ARnatomy*. However, the macroscopic and regional APs of head anatomy can be taught by *ARnatomy*, whereupon other APs build. Laymen can comprehend macroscopic and regional anatomy, i.e. anatomy observed with the unaided eye, without any professional training, which does not apply for other APs. When the user has learned something about macroscopic and regional anatomy with *ARnatomy*, the user should know at least the anatomy name, the anatomy location, and the anatomy appearance. Since then the user has a “*minimal working knowledge*” of anatomy as Turney [197] states, and can rudimentary communicate with others, especially the medical staff, about macroscopic anatomy.

All contained anatomical information in an education application should be scientific and medically correct, not verified information should be avoided to prevent the distrust of users. The interactive, rendered, synchronized, anatomical AR models should show at least their name during the most simple interaction in an application and provide opportunities to receive more information on one click. The requirement of showing at least the name originates in the importance of terminology in communication as Lampert [198] states. A user should be able to call learned anatomy entities by their name, otherwise the communication “*falls out of form*” [198]. The anatomy models should be mapped realistically for high immersion. The interaction should allow the user to perform a digital dissection [26] and assemblage for self-directed [46] discovery-based learning [27] and self-directed [46] exploration, since “*Active learning constructs spontaneous knowledge*” [13, 15].

Human anatomy is highly complex, filtering of anatomy regions and anatomy types [141] should be included to allow the user a more focused exploration than without anatomy filtering. A textual anatomy search for all supported models should be included too, if the user wants to search or discover a particular anatomy. In general, an application should preserve the first generated mental picture about the content as long as possible and should try to avoid the replacement and the recreation of multiple mental pictures of the same content for effective memorization [75]. Individuals of the general public should require as less technology as possible for their mobile or electrical education to make the access to the latter easy, convenient, and everywhere applicable. Currently, the best choice today to accomplish access to education easily is an application for a mobile

device or a website [199] as the growth of the former is ten times higher than computers in the eighties and two times higher than the internet in the nineties.

The answer to research question two is that the participants learned in the mean two new rostral anatomy structures within the mean time of 09:44 minutes, which is similar to state of the art works. The number of learned anatomies, depicted in the previous Table 5.3, was retrieved through the participant’s answers of the questions, listed in Appendix A.3, which were asked after the two case studies. The overall mean time of 09:44 is the sum from the two case study mean times of the IE. When a human head is viewed from the front, the rostral anatomy is the first and topmost anatomy, which explains why the most anatomy learned by the participants is located rostrally. The identified state of the art works depicted in Table 6.1 do not declare what anatomy is learned in which time, rather the education time of the participants and if there was an improvement on previously conducted tests or not.

Work	Year	AR	UBS	NP	ST	MT	ET
<i>ARnatomy</i>	2020	ML	HPE	8	x		09:44 minutes
Bork et al. [200]	2020	ML	none	16			2 hours
Kugelmann et al. [201]	2018	ML	Kinect	880			5 months
Noll et al. [79]	2017	MB	none	44	x	x	45:00 minutes
Küçük et al. [187]	2016	MB	none	70	x		-

Table 6.1: Juxtaposition of *ARnatomy* with state of the art works educating users about anatomy with AR. AR = Augmented Reality, UBS = User Body Synchronization MB = Marker-Based, ML = MarkerLess, NP = Number of Participants, ST = Short-Term, MT = Medium-Term, LT = Long-Term, ET = Education Time

Criteria of the selected works in Table 5.3 was that the education topic is anatomy and the content is mediated through AR technology. Works, which are more similar to *ARnatomy* as a mobile application with interactive, face-synchronized 3D, markerless AR anatomy models, were not found. The comparison with the works in Table 5.3 involves different properties, as the data situation for all identified state of the art works is unequal.

Noll et al. [79] and Küçük et al. [187] had different results on short-term memorization. Noll et al. [79] state that learning with AR resulted in no significant advantage compared to traditional learning, Küçük et al. [187] state there is a significant short-term advantage. The discrepancy maybe originates in the different implementations, since Noll et al. [79] use marker-based AR on humans and Küçük et al. [187] use pictures for marker-based AR in a book. The results of *ARnatomy* seem to be aligned with Küçük et al. [187], because two anatomies were learned in the mean of 09:44 minutes without the goal of learning them. The anatomy memorized in detail with *ARnatomy* was learned through interaction, i.e. active learning [13, 15]. From all works in Table 5.3, only Kugelmann et al. [201] and *ARnatomy* synchronize AR models with the the user’s body. Kugelmann et

al. [201] have a stationary screen using a Red Green Blue Depth (RGBD) camera on the top, which serves as the *magic mirror* to synchronize 3D anatomy models and Magnetic Resonance Imaging (MRI) with the user's body. *ARnatomy* uses in the FA only HPE without any depth information to synchronize the 3D anatomy models with the user's face. Bork et al. [200] use markerless AR on a flat surface, e.g. a table, which is highly similar to viewing a single anatomy in the SA of *ARnatomy*. Kugelmann et al. [201] and Bork et al. [200] contain both APs available in *ARnatomy*. The participants in *ARnatomy* memorized in the short-term in the mean two new anatomy structures under ten minutes just by interaction, without a set goal or set intention for learning. Noll et al. [79] were the only authors reporting on medium-term learning effects, wherein the participants who learned anatomy with AR performed significantly better in tests than other participants. In Table 5.3, long-term effects were excluded as no referenced work declared any results about long-term effects. After one month of the IE, the participants of *ARnatomy* were contacted in a short telephone interview and asked, if any learned anatomy name, location, or appearance is remembered. No specific anatomy was remembered by the *ARnatomy* participants, only the general rendering and interaction of and with *ARnatomy*.

Kugelmann et al. [201] and Bork et al. [200] did not declare, if their participants memorized new anatomical knowledge, rather a high user satisfaction, appreciation, and that the users perceived the learning applications as useful and practical. The *ARnatomy* participants of had the same view on AR as the participants of Kugelmann et al. [201] and Bork et al. [200]. Kugelmann et al. [201] and Bork et al. [200] had by far the longest education time of their participants, reoccurring over 5 months, which would have given different learning results, especially for medium-term or long-term memorization. Küçük et al. [187] did not specify how long the participants learned with the developed application. In *ARnatomy* the participants learned actively through interaction and probably with the shortest time, which makes a comparison insufficient for the works with short or no times. Similar as the user study times, the number of participants differ highly between all identified works, i.e. minimal eight and maximal 872. Only Bork et al. [200] supported simultaneous learning for multiple participants on the same rendered and viewed anatomy models.

In all identified works the models used, their rendering, and the AR technology differ too. Bork et al. [200] use 3D macroscopic anatomy models synchronized on the full body on a stationary screen. Kugelmann et al. [201] render pictures of different skin diseases on the skin detected by markers. Noll et al. [79] render 3D macroscopic anatomy models and two graphs, which are all not synchronized with the user's body. Küçük et al. [187] use 3D anatomy models, videos, and graphs rendered on top of pictures depicted in a book. *ARnatomy* synchronizes the head anatomy models from BP3D with the user's head on a mobile phone. The attributes *markerless*, *mobile*, and *user-synchronized* make together *ARnatomy* unique in the field of anatomy AR education. *ARnatomy* had as only work in Table 5.3 individuals of the general public as target group. As another difference

to all other works, *ARnatomy* had not one medical student as participant, whereby all other works in Table 5.3 had only medical students as participants. The testing with a homogeneous participants, i.e. only medical students, introduces a bias in the results [202] and in the comparison with *ARnatomy*, wherein more inhomogeneous participants were tested. The bias is considered to be small, as the number of participants in *ARnatomy* is the smallest compared to the other works in Table 5.3 and big numbers of homogeneous participants counterbalance the introduced bias. *ARnatomy* shares some properties with all selected state of the art works in Table 5.3, but the small subsets of properties among the identified works inhibit further comparison. However, the participants appreciate learning with AR from all works in Table 5.3 including *ARnatomy*, which is in alignment with previous conducted studies [13, 16, 17, 26, 27, 28, 75, 76, 77, 78, 191, 192]. The works in Table 5.3 which asked for obtained anatomical knowledge or tested the obtained anatomical knowledge resulted in advantages for at least short-term memorization.

The major difference of *ARnatomy* to the works in Table 5.3 is that the participants learned two new anatomy in the mean time of 09:44 minutes without instructions nor the goal to do so. The goal in all other studies with longer usage times was to learn anatomy and in the IE of *ARnatomy*, with shorter usage times, the goal was that the users learn anatomy through interaction during self-directed [46] and discovery-based [27] exploration. The learned anatomy with *ARnatomy* in a relative short time compared to the state of the art works in Table 5.3 suggests promising anatomy learning properties. The results obtained with *ARnatomy* and in the works in Table 5.3 confirm other results [16, 75] as well as the potential of AR as useful addition in macroscopic and regional anatomy education. Despite the short tested learning properties of *ARnatomy*, the positive outcome of the IE aligns the potential of this master thesis with the state of the art in anatomical AR education.

6.2 Conclusion

The mobile application *ARnatomy* was developed to educate individuals of the general public with anatomical, interactive, Three Dimensional (3D), Augmented Reality (AR) models [4] linked with textual information [5]. The models in *ARnatomy* are synchronized with the user's face and show additional information about anatomy during interaction. *ARnatomy* creates one mental picture [75] per anatomy entity and tries to extend the created mental picture with further information through the three developed perspectives on anatomy a user can occupy. In *ARnatomy*, the name, the synonyms, the definition, the appearance, and the location of macroscopic and regional [13] human head anatomy can be explored self-directed [46] and discovery-based [27]. Tapping, swiping, and pinch gestures are the interaction methods, which were all learned naturally by the eight participants after a short familiarization time during the conducted Informal Evaluation (IE) [45] of *ARnatomy*. The IE comprised of two case studies with an interview afterwards and tested short-term memorization. Seven of the eight participants have gained anatomical

knowledge in geometrical, spatial, and textual form. One participant, which did not learn something new, had already detailed knowledge about head anatomy. The other seven participants learned two anatomy structures within the mean time of 09:44 minutes through interaction, without any instruction to learn. The participants knowledge gain does not represent the general public and the results are therefore not applicable to the general public, but significantly convincing as other works [29, 79, 184, 185, 187, 200, 201] for the potential of mobile anatomy education with AR. Anatomical, spatial abilities along with short-term memorization [13, 16, 77, 78], the user satisfaction, as well as the user concentration are increased when using AR renderings for learning [27, 28, 75]. The aforementioned and positive effects were observed on the participants during the two case studies of the IE.

Usability problems were identified through the IE in the application giving valuable insights for improvements as well as attributes and properties to be mindful about in the future. Camera occlusions causing an automatic application restart have to be communicated and an initial interaction tutorial should mediate through accomplishment the interaction possibilities once. Each rendered anatomy should be clearly differentiable from each other anatomies via a colour-coded filtering, which can be turned on and off. Besides individuals of the general public, the medical staff can use *ARnatomy* as visual aid during consultations to create a more figurative mental picture about macroscopic and regional head anatomy in the user's mind. The mental picture in the user's mind can then be refined by the professional information explained by the medical staff. Albeit the contained scientific anatomical information [4, 5] in *ARnatomy*, its private usage or its usage in the medical field can only educate and support users [1]. *ARnatomy* cannot replace the consultation of a physician. Publications concerning education with AR grow [21]. *ARnatomy* and the obtained insights hopefully participate positively in the future of education with AR.

Future Work

Many extensions and improvements can be made and are reserved for future work as they impact many areas in the source code and application interaction, which were too time consuming to design and to implement within in the time limitations. Many ideas originated during research, implementation, the Informal Evaluation (IE) or as ideas. Shareable hyperlinks to application views are supported by *Android*. Links to show specific anatomy could be included in *ARnatomy* for easier and faster user communication about anatomy. Sending messages to the mobile device of a patient becomes possible too, through sharable hyperlinks of anatomy. Additional information or references concerning anatomy as region highlighting, small comments or notes could be included in the link parameters. In education facilities, a teacher may send to the students a link for their next task, homework or as simple visualization of anatomy during a class. When experiencing pain, patients may communicate the affected region or anatomical structure to doctors via hyperlink. Doctors could share hyperlinked anatomy too, e.g. for remote consultation and remote treatment. Users would be able to send other users messages including a link to an anatomy. The medical staff may benefit from shareable anatomy visualizations too, since educational levels about anatomy are different among nursing personnel, therapists, doctors, and patients causing sometimes misconceptions. Museums, art installations, or expositions concerning anatomy can benefit also, as they could send Augmented Reality (AR) preview messages to persuade the recipients for a visit.

The application may be extended to add notes or multiple choice questions [28] to anatomy objects or Anatomy Perspectives (APs) for learning. Even short videos, sounds, or images for *ARnatomy* customization may be added by teachers or the medical staff for an adapted guidance of discovery-based learning [27]. The customization should be possible without any programming skills [28] to not limit the interaction to information technology teachers. An edit mode activated by a password in the preferences of *ARnatomy* may show an additional icon on each label. A tap in the activated edit mode

redirects to an activity allowing the user to add multimedia content or textual notes for other users to view. Juanes et al. [203] state that learning with mobile devices can have an ubiquitous character, immediate student-teacher interaction, and a lower price. Further, learning with mobile devices has a greater accessibility and portability through an Internet connection. Important to note is, the benefits are only inherent when mobile learning and thoughtful usage is executed, otherwise mobile device usage results in an Intelligence Quotient (IQ) decline [8, 9, 10]. Mobile learning can comprise collaborative learning, i.e. experiencing digital enhanced learning material together were teachers can provide information, answers, or reminders [203]. Users may assess their effectiveness of learning anatomy [13]. In combination with the user's self assessment, a participation in the application development to not only consume but produce mobile AR for learning [192] becomes possible.

Images as markers for AR are supported by ARCore [22]. Simultaneously up to 20 static or moving images on a flat object are supported for realtime tracking. User mobile device movements capturing the images only temporally, when images are out of the camera view, are supported too by ARCore causing an improved tracking robustness. Right now, one device can store 1.000 detectable images on its local Database (DB), which is sufficient for the most macroscopic and regional anatomical structures. Single or multiple images in anatomical books could be linked via *AugmentedImages* from ARCore [22] and serve as marker. When AR anatomy models are rendered above the detected images, the user has an efficient and practical method for fast translation of learning environment from textual books into the digital AR content. A small icon printed near the image identifying the image as *augmented* is recommended, so users do not capture unnecessarily unsupported images.

Wang et al. [204] created an application capable of recognizing multiple images, even from long distances and successfully displaying corresponding Three Dimensional (3D) models as AR superimposition above the images. Wang et al. [204] had problems in crowded places due to occlusion, hence only usage with pictures of a book is recommended by the authors. During AR anatomy rendering started by image recognition, additional information can be viewed, and the 3D orientation of anatomy structures can be explored by interaction.

A digital dissection comes with more cognitive involvement, cognitive engagement, financial benefits, and wider accessibility as treated in Subsection 2.1.3. With Computed Tomography (CT) images and Magnetic Resonance Imaging (MRI), a digital dissection on 3D-generated models may become possible. A cut represents the plane or area where Volume Clipping (VC) is applied on the models created by AR volume segmentation of CT images or MRI similar to the work of Macedo and Apolinário Jr. [89]. The different black, white, and greyscale values of CT images or MRI can be used to differentiate between the background, bones, and soft tissue, and thus, their realistic colouring as

previously depicted in Figure 3.8. The VC could not only be applied to dissection, but additionally to the camera stream and clip the users camera-captured anatomy by rendering CT images or MRI as model clipping planes. The captured background behind clipped anatomy can be cached beforehand and rendered only when needed [89]. Combined with cinematic rendering of CT images or MRI [143] depicted in Figure 3.8, a new level of immersion would be created. ARCore’s 3D mesh face mask with 468 vertices may use few carefully selected points to synchronize the segmentation volume with the tracked face with rather low computational demand. The *Android* Neural Networks API (NNAPI) executes machine learning operations on mobile devices supporting *Android* 8.1 or higher and is implemented in the programming language *C*. NNAPI is designed to be a base for high level machine learning networks, which build and train neuronal networks [205]. *ARnatomy* is developed to support *Android* 8 or higher, so the Neural Networks API (NNAPI) introduced with *Android* 8.1 can be utilized for future releases. The NNAPI can be used as an overall improvement in Head Pose Estimation (HPE) by using state of the art works [104, 105, 108]. Probably body pose estimation state of the art detection approaches [133, 134, 135, 136, 137, 138, 139, 140], which ought to be implemented directly in the *Android Realm* and not as third-party software, could be added too.

Own written code could have better performance than third-party software due to the lower overhead in the application with the operating system *Android*. The requirements of all involved body pose estimation approaches must be verified with the capabilities of NNAPI before their implementation can start. With NNAPI, the supported anatomical region of the human body may be extended from the head to the full body. A free multiple human landmark detection framework originating in conflating multiple scientific papers [133, 134, 135] into one is *OpenPose* [133]. Body pose estimation, food pose estimation, hand pose estimation, and head pose estimation of multiple people in one image or frame are fully supported including frontal, up, down, and side views as depicted in Figure 7.1.

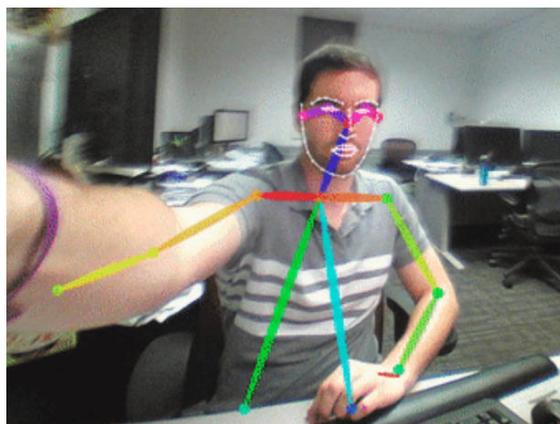


Figure 7.1: Detected body and face landmarks in realtime with *OpenPose* [133].

Important to note is the invariancy of computational demand, i.e. up to 30 people are

tracked simultaneously with the same computational demand as tracking only one person. The programming in C++ can make *OpenPose* available on mobile devices supporting Native Development Kit (NDK). Careful research and requirement checking is key to prevent lagging as experienced with the usage of *dlib* [92]. The benefit of *OpenPose* compared to other recent works [136, 137, 138, 139, 140] is its availability. However, including *OpenPose* as third-party C++ extension may be slower than using other body pose estimation approaches [136, 137, 138, 139, 140] utilizing Convolutional Neuronal Networks (CNN)s due to the overhead of communication between Java and C++. The CNNs from all the different works [136, 137, 138, 139, 140] may be implemented with the dedicated high efficient native *Android* NNAPI counterbalancing the overhead. A lot of subtasks in recognition, e.g. *lifting* Two Dimensional (2D) joint locations into 3D with approximately 30 % better performance, can still be improved by deep feed-forward networks for receiving state of the art performance results [137].

Apart from C++ and NNAPI, the “*Powerful, efficient, lightweight, embeddable scripting language Lua supporting procedural programming, object-oriented programming, functional programming, data-driven programming, and data description*” [206] may be considered as programming language for more performance. The *Android* operating system can run Lua scripts with the *LuaJ Application Programming Interface (API)* [207]. Free works on human pose estimation with feature pyramids in deep CNNs [208] using *Lua*’s features are published and achieve the same accuracy with less computational demand as similar estimation approaches written in other programming languages. As *Lua*’s binary files do not exceed the one Mega Byte (MB) mark, it may be used as application size decrement too. A comparison of benchmarks on the same tasks in *Lua* as well as C++ on *Android* are currently inexistent and hopefully yet to come.

The current implementation scales anatomy models to fit into the corresponding meshes superimposing the tracked faces. Facial user anatomy proportions may differ from the standardized Japanese anatomy models used in the Body Parts 3D (BP3D) flat file collection [4]. Model deformation results in a better model fit for the face mesh, more realistic rendering, better superimposition, and higher immersion. Many strategies for adaptive physical model deformation exist. Manteaux et al. [129] give a survey of proposed deformation approaches so far in computer graphics categorized in time-stepping, freezing, geometric adaptivity of grids, meshes or particles from fluids, deformable objects, and articulated solids. Several different techniques may be used as the human anatomy consists of various forms and shapes consisting of different organic tissues as well as textures.

RenderScript focuses primarily on parallel and secondarily on serial computation of computational demanding tasks executed with high performance on *Android* mobile devices. The execution is distributed on all the available Central Processing Units (CPUs) and Graphical Processing Units (GPUs) on a device [209]. RenderScript may be

an excellent choice to outsource the resource management of the complex deformation processes to multiple CPUs and GPU. GPUs allow a developer to create efficient parallel computation. With the landmarks of the aforementioned full body estimation, the deformation boundary [210] of anatomy could be determined by the silhouette of the body as in the work by Ali-Hamadi et al. [210]. Ali-Hamadi et al. [210] achieve fast deformation generation for multiple bodies. Deformation can be combined with medical data or certain animations to create highly realistic and immersive visualizations of human anatomy, which may be dynamic as a beating heart and breathing lungs [26]. Model animations according to the heart rate give a more astonishing impression and realism to the visualization resulting presumably in a longer memorization by the learners.

Users with low and high spatial abilities profit from supportive handles facilitating repositioning and orientation changes of rendered models [13]. Low spatial ability users gain more profit as their handicap is counterbalanced. Cognitive load needed for spatial changes is decreased and available for learning. Users with a high spatial ability completed their tasks faster than users with a high spatial ability but without supportive handles. Supportive handles should be rendered only on selected models to facilitate the positioning and to highlight additionally the current selection.

The support of different colour modes for people whose colour vision is impaired may be considered in the future. Eight percent of the male and 0.5 % of the female world population suffer from colour illnesses [211]. Tanuwidjaja et al. [211] propose four colour modes as support for a better colour distinction targeted at colour blind people. The anatomy model colouring can be implemented accordingly to the major colour deficiencies, whose spectrums are depicted in Figure 7.2.



Figure 7.2: Normal and impaired colour spectrum [211].

The four modes of *Chroma* [211] are highlight, contrast, Daltonization, and outline. Highlighting is simple area selection by the user and the area's highlighting. Contrast allows the user to select two colours and the second colour then filtered by the first colour. Daltonization is a colour spectrum shift for the best colour distinction of a user

having colour deficiencies. Outlining outlines all colour regions influenced by the user's colourblindness for easier distinction. The IE consisting of the two case studies with the interview gave interesting insights and promising ideas from the participants on future application implementations. In the Hierarchy Activity (HA), rendering the head anatomy transparent in the scene view and rendering only the currently viewed anatomy with a red material to see the anatomical context would result in a better illustration of the anatomy location. An option to turn the whole head rendering off would have to be included to render only one anatomy. If one, some, or all ideas from Chapter 7 are implemented, another user study may be conducted with the AR analytics framework [212] to get more insight into the idea's effects on the usage of *ARnatomy*.

Appendix

A.1 Body Parts 3D and Foundational Model of Anatomy Column Names

- Body Parts 3D (BP3D) file [4]: Compatibility version, File ID, Representation ID, Build-up logic, Concept ID, English name, Bounds (mm), and Volume (cm³)
- Foundational Model of Anatomy (FMA) Comma Separated Values (CSV) [157]: Class ID, Preferred Label, Synonyms, Definitions, Obsolete, CUI, Semantic Types, Parents, AAL, AAL ID, abbreviation, action type, adheres to, adjacent, adjacent to, afferent to, anatomical entity observed, anatomical landmark for, anatomical plane, anterior to, anteroinferior to, anterolateral to, anteromedial to, anterosuperior to, AP position, arterial supply, arterial supply of, articulates with, attaches to, attributed constitutional part, attributed development, attributed part, attributed regional part, author, authority, bounded by, bounds, branch, branch continuity, branch of, cell appendage type, CMA label, connected to, connecting part, connection type, constitutional part, constitutional part of, contained in, contains, continuous distally with, continuous inferiorly with, continuous proximally with, continuous superiorly with, continuous with, contributing author, corresponds to, creator, date entered modified, days post-fertilization, definition, derives, derives from, development type, developmental fusion, developmental stage of, develops from, develops into, dimension, direct left of, direct right of, distal to, DK Freesurfer, drains into, efferent to, English equivalent, Eponym, external to, FMAID, forms, Freesurfer ID, full grown phenotype, fuses with, fusion of, germ origin, gestation age weeks, has anatomical landmark, has boundary, has developmental stage, has dimension, has direct cell layer, has direct cell shape, has direct number of pairs per nucleus, has direct ploidy, has inherent 3-D shape, has insertion, has location, has mass, has origin, has projection, homonym for, homonym of, <http://data.bioontology.org/metadata/prefixIRI>,

<http://purl.org/dc/elements/1.1/contributor>, <http://purl.org/dc/elements/1.1/description>, http://purl.org/sig/ont/fma/anatomical_coordinate, <http://purl.org/sig/ont/fma/laterality>, inferior to, inferolateral to, inferomedial to, inherent 3-D shape, input from, insertion of, is boundary of, JHU DTI-81, JHU White-Matter Tractography Atlas, language, lateral to, Latin name TA, left lateral to, left medial to, located in, location of, lymphatic drainage, lymphatic drainage of, matures from, matures into, medial to, member, member of, merges with, merging of, modification, modified by, muscle attachment, name, nerve supply, nerve supply of, nerve supply type, Neurolex, Neurolex ID, NeuroQuant, NN abbreviations, non-English equivalent, nucleus of origin of, nucleus of termination of, orientation, origin of, other Latin equivalents, outdated meaning, output to, part, part of, part type, partition, percentage, physical state, PI-RADS v1 16 ID, PI-RADS v1 27 ID, PI-RADS v2 ID, polarity, posterior to, posteroinferior to, posterolateral to, posteromedial to, posterosuperior to, preferred name, primary author and curator, primary segmental supply, primary segmental supply of, projects from, projects to, proximal to, RadLex ID, rank of tissue, receives attachment from, receives drainage from, receives input from, receives projection, reference, regional part, regional part of, related developmental entity, related object, related part, right lateral to, right medial to, RO ID, secondary segmental supply, secondary segmental supply of, segmental composition, segmental composition of, segmental contribution from, segmental contribution to, segmental innervation, segmental supply, segmental supply of, sends output to, sensory nerve supply, slot synonym, species, state, state of determination, superior to, superior-inferior position, superolateral to, superomedial to, surrounded by, surrounds, synonym, TA ID, Talairach, Talairach ID, technical support, term status, transforms from, transforms into, tributary, tributary continuity, tributary of, UMLS ID, union, unites with, venous drainage, venous drainage of, view

A.2 Projects and Issues on Github

The following projects were created during this master thesis as aid:

1. `augmentedFaceMeshIndices`, Google ARCore *Sceneform* 1.10.0 augmented face mesh indices from their `.fbx` file as pictures [132].
2. `fma2sqlite`, a bash script converts the Foundational Model of Anatomy (FMA) `.csv` file into a minimal SQLite database [157].
3. `f2s`, Files 2 SQLite (F2S) converts all files in one directory into a SQLite database [159].

The following issues were created due to found malfunctions during this master thesis:

1. *Sceneform* Issue 217: Convert multiple `.obj` to `.sfa` and `.sfb`.

2. *Sceneform* Issue 250: Render PreferenceScreen as ViewRenderable.
3. *Sceneform* Issue 281: Let Renderable implement Serializable or Parcelable.
4. *Sceneform* Issue 334: [Question][Improvement] ARCore Node implementation [213].
5. *Sceneform* Issue 387: Video stream changes, background black and white in relation to distance.
6. *Sceneform* Issue 613: The .obj Converter fails since version 1.7.0 [142].
7. *Sceneform* Issue 733: Is it possible to get documentation in *Android* Studio ? [169].
8. *Sceneform* Issue 798: Render models behind the Augmented Face Mesh inside the head [172].
9. Dlib Issue 1621: xlib11-dev, libopenblas-dev, and liblapack-dev installed but not found [164].

The following issues affected this master thesis:

1. ARCore Issue 89: ARCore Device Support Requests
2. *Sceneform* Issue 251: Can not make a white background on SceneView.
3. *Sceneform* Issue 369: The Child Node inside a Room Node is not getting the callback for the OnTouchListener.
4. *Sceneform* Issue 563: ArCore - 1.7 Face detection Api for All classification and landmarks.
5. *Sceneform* Issue 580: Custom material [174]
6. *Sceneform* Issue 764: [device support] Nokia 7 Plus (*Android* 9) is not supported [176].
7. Dlib Issue 181: Face landmark retrain takes very long time.
8. Dlib Issue 305: Parameters used for training shape predictor.
9. Dlib-*Android* Issue 49: Question: Is there any way that JNI take bitmap or byte array of the bitmap and convert to dlib::array2d without openCV?.
10. FMA-Structured Query Language Lite (SQLite) Issue 2: List Index out of Range error.
11. E-Mail to Biportal: Warning about invalid entries [156].

One may observe the generation date and the closing date of all issues about ARCore [22] and *Sceneform* [23].

A.3 Instructions and Questions

All participants had German as their mother language. Naturally the below instructions and questions in English were given and asked in German to eliminate translation errors or confusion. The questions are inherited and inspired by Lam et al. [45] and Shipley et al. [46].

The instructions for the case study *cerebellum* are:

1. Search for “cerebellum”
2. Go one up, find the “pons”
3. Look at the single “pons” in augmented reality
4. Go back to the face view
5. Find now the “cerebellum” or the “pons”

The instructions for the case study *mandible* are:

1. Filter everything except the mandible
2. Show its label
3. Go from the label in the detail view
4. Find the “vomer”
5. Show the “vomer” in the face view

The asked questions after the completion of the two case studies:

1. What is your age?
2. What is your gender?
3. What is your profession?
4. How much time do you spent on your mobile device?
5. Primary reason for your mobile device usage?
6. Are you interested in learning anatomy?
7. Is an anatomy learn app interesting for you?

8. What was your first impression?
9. What was your last impression?
10. Is the app usage understandable? Why?
11. Can the usage of the app be learned? Why?
12. What have you learned about anatomy?
13. To you think the general public could learn anatomy with this app?
14. Would you reuse this app to learn something about anatomy or where anatomy is located?
15. Would you recommend this app? How?
16. What do you think about the presented anatomy? How do you like it?
17. What would you change with the presented anatomy?
18. What would you change with the interaction?
19. What do you like about the app?
20. What do you dislike about the app?
21. Do you want to say something additionally?

List of Figures

2.1	Notions between virtuality and reality [48, 49].	10
2.2	Example of a One Dimensional (1D) marker [58], which can be used for Augmented Reality (AR) orienting its renderings relative to the 1D marker position in the captured scene by the device camera.	11
2.3	Left: Decoding procedure and direction of the data matrix inside a Two Dimensional (2D) marker. Middle: Decoding procedure and direction of the data matrix inside a Quick Response (QR) code. Right: Decoding procedure and direction of the data matrix inside a colour code [57].	12
2.4	Creation process of a stereogram [59].	14
2.5	Outdoor AR: a) Polygon side red center adjustment. b) AR content shown due to correct building identification. c) No content due to incorrect Field of View (FoV) not hitting the center of the buildings facade. d) False positive match as a) due to perspective distortion [60].	15
2.6	Left: The object in reality. Right: The generated model ground truth used for AR registration [61].	16
2.7	Green lines indicate matched key points identified by ORB on a view point change. Red dots indicate unmatched points [69].	19
2.8	Visualization of the generated Oriented Features from Accelerated Segment Test (FAST) and Rotated Binary Robust Independent Elementary Features (BRIEF) Simultaneous Localization and Mapping (ORB-SLAM) 2 map from a moving vehicle [71].	21
2.9	Each single line end is a matched feature point. The blue line clusters are used for camera pose calculation and the orange lines are excluded [64].	22
2.10	The final build depth map at the top with the corresponding real scene at the bottom [65].	23
2.11	The yellow line represents the resection line and for novices the tumor is highlighted [83].	30
2.12	A sketched blood vessel with its metaballs and the handheld [84].	31
2.13	a) Skeletal model used b) Knee flexion muscle activity c,d) Skeleton superimposition e) Multiple organ superimposition f) Single selected organ superimposition [74].	33

2.14	Left: Ankle with the drill entering as small yellow dot is visualized. Right: Focus and context rendering with shading as an additional depth cue of the ankle [87].	33
2.15	The green points show future snout positions and the cyan points past snout positions. The magenta coloured lines show the body orientation determined by four vertices [88].	35
2.16	Left: Focus and Context Visualization Paradigm (FCVP) with smooth contours on the user. Middle: Visible background with a superimposed Computed Tomography (CT) image. Right: Volume clipping of the user by Magnetic Resonance Imaging (MRI) data [89]	36
2.17	The processing pipeline of <i>WithTeeth</i> [30]	37
2.18	Final result of <i>Withteeth</i> , zoomed in on teeth superimposition by virtual denture [30].	38
2.19	Image a) Operator perspective seeing the AR with an incorrect C-Arm (CA) alignment. Image b) Side X-ray image of alignment in a). Image c) Top X-ray image of alignment in a). Image d) Operator perspective seeing the AR with a correct CA alignment [90]	39
2.20	Triangular fuzzification with the lowest (x_l), mean (x_c), and highest (x_r) greylevel intensity values [102].	42
2.21	The Three Dimensional (3D) axes with their rotations shown on a sketch of a face.	42
2.22	Multiple detected faces with their 3D rotational axes: x red, y green, and z blue [108].	44
3.1	The methodological pipeline of this master thesis as Unified Modeling Language (UML) flow diagram.	49
3.2	Macroscopic and regional perspective of vessels in the cerebral region with iris and choroid from the eyes as Point of Reference (PR) from the BP3D flat file collection [4].	52
3.3	Entity Relationship (ER) model of the created <i>ARnatomy</i> Database (DB) from the conflation of BP3D and FMA.	55
3.4	Left: AR bone and organ models synchronized with the detected face. Right: Generated image of the ARCore .fbx frontal face mesh indices in wireframe view [132].	60
3.5	View of the frontal bone 30 centimeters above the horizontal floor plane with ARCore points for orientation.	60
3.6	Anatomy region and anatomy type filtering through the AR filter options beside the detected face.	64
3.7	UML activity diagram showing the workflow of the Face Activity (FA), Hierarchy Activity (HA), and Single Activity (SA). The view results are determined by the user through the search and allows users's to go from the FA to the HA.	65

3.8	Left: Skull interior with the midline as clipping plane. Right: Bony surface rendering. Left and right: Renderings of a non-contrast-enhanced CT [143].	66
3.9	Left: The introduction dialog for basic AR interaction clues. Right: The <i>Android Toast</i> message shown on search Floating Action Button (FAB) tap to indicate that the database extraction is ongoing.	67
3.10	Left: Standard Phong lighting with shadows of a metallic model in ARCore. Right: The same as left, but with environmental High Dynamic Range (HDR) light estimation, note the specular light and model shadow adjustment to the natural lighting in the captured scene [145].	68
3.11	Coloured anatomy model labels in the FA.	70
3.12	The Hue Saturation Value (HSV) colour cylinder.	71
3.13	The cache list above the head from the FA with the left frontal gyrus, right frontal gyrus, and frontal bone added.	71
3.14	Annotated arrows on anatomy in the cache list to illustrate the possible swipe directions. A downward swipe would locate the anatomy back into the face. A left swipe or right swipe would scroll through all anatomies in the cache list to the left or right.	72
3.15	Visualization of transparent context revealing anatomy located deeper inside the head.	74
3.16	All checked FA filtering options rendered in AR beside the detected face.	75
3.17	Left: Textual search for the first characters of <i>frontal bone</i> with its results displayed as grid. Right: The frontal bone with its activated external label synchronized with the face.	76
3.18	The popup menu from a single anatomy entry in the HA. The eye starts the AR anatomy model rendering in the SA. The smiley starts the FA with the corresponding anatomy highlighted by its external label. The up or down arrows try to load further anatomy hierarchy. The share icon starts the <i>Android Sharesheet</i> [149] to select a target app as content receiver.	77
3.19	The full FMA hierarchy of the left and right cornea displayed in the HA with the popup menu of the right cornea opened.	78
3.20	The hand motion picture of ARCore to indicate mobile device movement for better plane detection.	78
3.21	Designed application icon.	80
4.1	The external ear model from BP3D flat file collection version 4.0 [4].	82
4.2	Example picture from the HELEN face dataset [163] with 194 face landmarks.	86
4.3	Top Left: Left and right lateral partial face occlusion requires one half of an eye for detection. Top Right: Top lateral partial face occlusion requires both eyes to be visible for detection. Bottom Left: Top partial face occlusion through head tilts requires both eyes to be visible for detection. Bottom Right: Bottom lateral partial face occlusion requires both eyes to be visible for detection.	90
4.4	Right eye mesh indices of an ARCore <code>.fbx</code> file [132].	91

4.5	UML class diagram without methods of the <i>Anatomy</i> class and its relations.	92
4.6	Left: Barycenter of the zygomatic bone visualized as green dot positioned in the middle. Right: Barycenter as green dot positioned at the bottom, with the <i>y</i> coordinate set to zero. Different positioned barycenters result in different model positions after a <i>x</i> axis rotation on the lower two images. Further rotations, e.g. on the <i>z</i> axis, would increase the position difference.	97
4.7	The <i>z</i> -axis translation controlled by pinch gestures viewed from a side angle for better representation. Left: Illustration of the thresholds from the focus and context areas. Middle: Back-translated anatomy in focus mode with obscuring face mesh on the cheeks. Right: Front-translated anatomy with anatomy, eyes, teeth, and zygomatic bone, in the context area and therefore transparent.	98
4.8	A left camera turn from the right over the middle to the left image showing the background colour change and blur due to AR model proximity [179]. .	101
5.1	Top Left: Sunglasses have no effect on Head Pose Estimation (HPE) with anatomy translated backwards for illustration. Top Middle: Low illumination right before HPE. Top Right: Low illumination right after HPE. Bottom Left: Cap just not hiding the eyes. Bottom Middle: High illumination right before HPE. Bottom Right: High illumination right after HPE.	105
5.2	Left: Labeled <i>cerebellum</i> from the side view for a better recognition. Right: Labeled <i>pons</i> from the front.	108
5.3	Text search with the automatic suggestions for the <i>cerebellum</i>	109
5.4	Left: Standard anatomy view of <i>cerebellum</i> . Middle: Hierarchy expansion of one level. Right: Selection of <i>pons</i>	110
5.5	<i>Pons</i> AR rendering in the SA with its label activated.	110
5.6	Left: The menu FAB invoked in the FA with the options <i>about</i> , <i>restart</i> , and <i>filter</i> . Right: Same FAB menu as left but invoked in the SA with the options <i>settings</i> , <i>restart</i> , and <i>face</i> . The settings allow the user to turn labels or plasticity on or off and go to <i>about</i> , showing brief application information.	111
5.7	Highlighted anatomy during search for <i>cerebellum</i> or <i>pons</i> with its label shown. All other anatomy is transparent.	112
5.8	Anatomy is located above the user's head during the exploration to find the <i>cerebellum</i> or the <i>pons</i> . A label of the currently viewed anatomy is shown additionally.	114
5.9	The external label of the mandible with unfiltered face anatomy for comparison with filtered face anatomy in Figure 5.10.	115
5.10	Filtered anatomy except the oral region to only show the mandible and maxilla. The filter panels for anatomy region and anatomy type are partially visible beside the face. Eyes are occluded for the participant's anonymity.	116
5.11	Label of the filtered mandible. Eyes are occluded for the participant's anonymity.	117
5.12	The hierarchy view of the mandible, showing the target anatomy, the vomer.	118

- 5.13 The highlighted label of the vomer in the FA coming from the HA. As the colour is orange, the highlighting is hard to detect right below the tip of the nose. Eyes are occluded for the participant’s anonymity. 119
- 6.1 Created with image manipulation, to illustrate the initial interaction tutorial for the tapping interaction. Long tap, swipe, pinch, and icon tutorials should be highly similar. 133
- 6.2 An assumption of the current learning curve based on the performance of the eight participants compared to the tutorial one. The tutorial learning curve is expected to be far more ascending due to the required interaction in the tutorial. The upper boundary of both curves marks the capability of performing tapping, long tapping, swiping, and the pinch gesture. 134
- 6.3 The coloured anatomy, with external labels shown, is best choice for search targets in the first exploration of a case study due to their close location to the forehead. 135
- 7.1 Detected body and face landmarks in realtime with *OpenPose* [133]. 151
- 7.2 Normal and impaired colour spectrum [211]. 153

List of Tables

2.1	Overview of AR frameworks with their most important features [120, 121, 122, 123, 124, 125].	47
3.1	Rendering source code colours of the mobile application similarly selected to the <i>Anatomography</i> of BP3D [4] colours to differentiate anatomy easier. .	69
3.2	User Interface (UI) colours of the mobile application with their assigned purposes.	80
5.1	The line number of source code and documentation. Only <i>javadoc</i> comments are counted as documentation, whereby many source code comments are written too.	103
5.2	General information about the participants. A = Age, G = Gender, M = Minutes a Day on a mobile device, MR = Main Reason of mobile device Usage, CSC = Case Study Cerebellum Completion Time, CSM = Case Study Mandible Completion Time.	106
5.3	Juxtaposition of the participants and their learned anatomy. A = Age, G = Gender, N = Number of learned anatomy.	122
6.1	Juxtaposition of <i>ARnatomy</i> with state of the art works educating users about anatomy with AR. AR = Augmented Reality, UBS = User Body Synchronization MB = Marker-Based, ML = MarkerLess, NP = Number of Participants, ST = Short-Term, MT = Medium-Term, LT = Long-Term, ET = Education Time	144

Glossary

- ABI** An Application Binary Interface (ABI) connects two binary program modules via an interface defining how data structures or computational routines are accessed in a low-level, machine code. The opposite is an Application Programming Interface (API). 58, 89
- API** An Application Programming Interface (API) is a particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another particular software program that implements that API. 40, 51, 81, 152, 169
- ARCore** ARCore is Google's platform for building AR experiences. It enables the mobile device to sense its environment, understand the world, interact with information, and integrate virtual content into the real world [22]. xi, xiii, 3, 5, 46–49, 54, 55, 57, 59–63, 65–68, 70, 73, 78, 79, 81, 84, 85, 87–100, 103, 104, 110, 111, 113, 116, 124–126, 130, 133, 136, 150, 151, 157, 162, 163
- BA** Bundle Adjustment (BA) is the optimization of line of sight bundles in a 3D scene or 3D map recorded by multiple cameras, where each camera can have a different perspective. 20
- BE** Back-End (BE) refers in software engineering to the data access layer of a piece of software, or the physical infrastructure or hardware. The Back-End (BE) counterpart is the Front-End (FE). 5, 20, 49, 61, 83, 169
- boilerplate code** In computer programming, boilerplate code or boilerplate refers to sections of code that have to be included in many places with little or no alteration. It is often used when referring to languages that are considered verbose, i.e. the programmer must write a lot of code to execute small tasks. 84, 92
- BP3D** Body Parts 3D (BP3D) is a free, scientific, digital 3D, anatomical model flat file collection with multiple versions including an online view of its models named *Anatomography* [4]. xi, xiii, 1, 40, 51, 81, 124, 130, 152, 155
- BRIEF** Binary Robust Independent Elementary Features (BRIEF) is an efficient real-time feature point descriptor using binary strings. BRIEF is fast to build, compare,

and achieves a high accuracy despite having high speed, which is gained via moving from the Euclidean to the Hamming distance for matching purposes [70]. 17, 67, 90, 161

CA A C-Arm (CA) intensifies scanned images, mostly fluoroscopic ones, primarily during surgical or orthopedic procedures. Figuratively depicted, the patient or object to be scanned lies inside the recess of the letter *C*, which is the name origin of the CA. 38, 162

CCN Cascaded Convolutional Networks (CCNs) are a consecutive interconnection of single Convolutional Neural Networks (CNNs) without feedback. Thus, the adjective *cascaded* in the name of CCN. 42, 57

CNN Convolutional Neural Networks (CNNs) are a class of deep neural networks, wherein neuron connectivity patterns resemble the organization of an animal visual cortex. CNNs are designed to require minimal preprocessing. 42, 57, 152, 170

CT Computed Tomography (CT) scans, formerly known as a Computerized Axial Tomography (CAT) scans, are computer-processed X-ray measurement combinations taken from different angles to produce cross-sectional (tomographic) images or virtual slices of specific areas of a scanned object, allowing observers to see into an object without invasive interference. 29, 50, 150, 162

DVR Direct Volume Rendering (DVR) gets a 3D representation of the volume data directly. The data is considered to represent a semi-transparent light-emitting medium, which is used slice by slice for 3D volume generation. 35

EMF An Electromagnetic Field (EMF) is a physical field produced by electrically charged objects. 39

FAB A Floating Action Button (FAB) performs the primary, or most common, action on a screen. It appears in front of all screen content, typically as a circular shape with an icon in its center. In this master thesis, the actions of a FAB are overloaded depending on the current status. 64, 107, 132, 163

FAST Features from Accelerated Segment Test (FAST) is a very fast high quality corner detector who uses machine learning and has high repeatability levels “*under last aspect changes for different kinds of features*” [68]. 17, 67, 90, 161

FCVP Focus and Context Visualization Paradigm (FCVP) shows the medical data on the patient’s body as focus region, provides a better perception and immersion of medical data for the viewer as superimposition than without FCVP. 35, 36, 162

FE Front-End (FE) refers in software engineering to the data presentation layer of a piece of software, or the physical infrastructure or hardware. The Front-End (FE) counterpart is the Back-End (BE). 5, 20, 49, 61, 92, 170

- flat file** Flat-Files are ordinary, unstructured files on the operating system's file system. 1, 6, 40, 51–54, 61, 66, 81–84, 98, 125, 152, 162, 163, 169
- Flynn effect** The Flynn effect is the steadily rise within fluid and crystallized intelligence test scores retrieved in multiple countries throughout the twentieth century. Research of Bernt Bratsberg and Ole Rogeberg suggests that there is an ongoing reversed Flynn effect, meaning a fall in Intelligence Quotient (IQ) scores, approximately starting in the final decade of the last millennium [6, 7]. 1
- FMA** The Foundational Model of Anatomy (FMA) is a mature ontology of anatomy created by the Structural Informatics Group at the University of Washington [5]. xi, xiii, 1, 52, 81, 107, 130, 155
- FPS** Frames Per Second (FPS) is the frequency of consecutive images, named frames, which appear on a display. 24, 88
- hooking** Hooking is a technique used to alter or augment the behaviour of other software components by intercepting function calls, messages, or events passed between software components. Code executing the described behaviour is simply called a hook [42]. 94, 95
- HSV** Hue Saturation Value (HSV) is an alternative representation of the Red Green Blue (RGB) colour model. 70, 163
- ICP** Iterative Closest Point (ICP) is an algorithm employed to minimize the difference between two or more point clouds. 20
- IDE** Integrated Development Environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools, and a debugger. 48, 59, 94
- IMU** An Inertial Measurement Unit (IMU) is an electronic device that measures and reports a body's specific force, angular rate, and sometimes the magnetic field surrounding the body, using a combination of accelerometers and gyroscopes, sometimes also magnetometers. 24
- JNI** The Java Native Interface (JNI) is a foreign function interface programming framework that enables Java code running in a Java virtual machine (JVM) to call and to be called by native applications and libraries written in other languages such as C, C++, and *assembly*. 47, 58, 86
- k-means** The K-means clustering is used to minimize the feature point cluster to a prototype of itself resulting in less computation time [64]. 22, 171

LBA The idea of the Living Book of Anatomy (LBA) is to capture a user action (limb movement, speech sound or orofacial movement) with Microsoft Kinect (MK) and to visualize information about the muscles superimposed onto the user image [74].
32

long-term Long-Term is defined in the Oxford dictionary as *occurring over or relating to a long period of time* [20], and is in this master thesis considered to be more than a month. 3, 21, 27, 28, 69, 120, 140, 141, 145

LSD-SLAM Large Scale Direct - Simultaneous Localization and Mapping (LSD-SLAM) is a direct monocular SLAM technique: Instead of using landmarks or feature extraction, it directly operates on image intensities both for tracking and mapping [73]. 17, 23

LSM Log Structured Merge (LSM) trees are search trees who have a high performance on indexed big insert volume scenarios. Key Value (KV) pairs are maintained in two or more separate structures optimized for KV's underlying storage medium and data is synchronized between the two or more separate structures in batches.
85

medium-term Medium-Term is defined in the Oxford dictionary as *occurring over or relating to a period of time of moderate length* [19], and is in this master thesis considered to be less than a month but longer as short-term. 3, 27, 28, 69, 120, 140, 141, 145

metaballs In computer graphics, metaballs are organic-looking 3D-dimensional objects.
31, 161

MRI Magnetic Resonance Imaging (MRI) scans are computer-processed Nuclear Magnetic Resonance (NMR) measurement combinations taken from different angles to produce cross-sectional (tomographic) images or virtual slices of specific areas of a scanned object, allowing the viewer to see into an object without invasive interference. In comparison to CT, no X-rays are involved, but rather an in-vivo atom core resonance induced by strong magnet fields eliciting an observable electric signal used to accumulate a MRI picture. 29, 50, 145, 150, 162

NDK A Native Development Kit (NDK) is used to generate native code bibliographies from another programming language. 47, 58, 86, 152

NMS Non-Maximum Suppression (NMS) is an edge thinning technique applied to find *the largest edge*. 43

NNAPI The *Android* Neural Networks API (NNAPI) executes machine learning operations on mobile devices supporting *Android* 8.1 or higher and is implemented in the programming language *C*. NNAPI is designed to be a base for high level machine learning networks, which build and train neuronal networks [205]. 151

- OAH** The Online Anatomical Human (OAH) is a web-based viewer for studying anatomy. It is based on real human anatomy and incorporates medical image data in linked 2D and 3D views that students can freely interact with [31]. 29
- OpenGL** Open Graphics Library (OpenGL) is a cross-language, cross-platform API for rendering 2D and 3D vector graphics. 51, 100
- ORB-SLAM** Oriented FAST and Rotated BRIEF Simultaneous Localization and Mapping (ORB-SLAM) is an approach of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. The *ORB* highlights the oriented extension of the FAST key point detector and BRIEF descriptor who have high performance at low cost [69]. 17, 67, 90, 161
- ORM** Object Relational Mapping (ORM) is a programming technique for converting data between incompatible system types via a virtual object database for object data conversion and transmission. 45, 53, 54
- Phong** Phong lighting is an empirical model of the local illumination of points on a surface. It describes the way a surface reflects light as a combination of the diffuse reflection on rough surfaces with the specular reflection on shiny surfaces. 67, 68, 163
- PLK** Pyramidal Lucas-Kanale (PLK) finds the location of a certain picture element (pixel) area from one image in a sequential image by the area's intensity. PLK computes a velocity vector for the area known as optical flow. Window size is proportional to accuracy and robustness, bigger windows smooth the area, smaller windows capture more motion. The pyramidal version blurs and shrinks the image recursively. From the pyramids peak down to the biggest image the standard Lucas-Kanale is applied and each result is the initial value for the next pyramid level [66]. 24
- RenderScript** RenderScript focuses primarily on parallel and secondarily on serial computation of computational demanding tasks executed with high performance on *Android* mobile devices. The execution is distributed on all the available Central Processing Units (CPUs) and Graphical Processing Units (GPUs) on a device [209]. 152
- RNN** A Residual Neural Network (RNN) builds on pyramidal cells similar to the ones in the cerebral cortex. RNN utilize skip connections or shortcuts to optionally jump over one or more pyramidal cells. 34
- Scale Drift** Scale Drift (SD) is an interpretation change over time legitimately applied to entries inside a certain scale [72], e.g. a map, model, measure or criterion. Through scale drift, gathered data differs over time more and more from its ideal or reality. 21

- SDK** Software Development Kit (SDK) is a set of development tools for creating applications with a certain software, software package, software framework, or hardware, for and on a certain platform. An SDK can be some files, an API, with or without documentation or include sophisticated hardware to communicate with a certain embedded systems. 28, 81
- short-term** Short-Term is defined in the Oxford dictionary as *occurring over or relating to a short period of time* [18], and is in this master thesis considered to be shorter than a day. 2, 3, 27, 28, 62, 68, 120, 127, 140, 141, 144–147, 172
- SQLite** Structured Query Language Lite (SQLite) is a public domain library that contains a relational DB system. It supports most of the SQL language commands specified in the SQL-92 standard [127]. 45, 53, 81, 157
- SRC** The Sparse Representation Classification (SRC) is a face recognition classifier using pixels without feature extraction and the l1 optimization for less influence of illumination and occlusions [101]. 41
- surfel** Surface elements (surfels) is in 3D computer graphics an alternative to polygonal modeling. An object is represented by a dense set of points or viewer-facing discs holding lighting information. 17
- ToF** Time of Flight (ToF) cameras measure distances with the time of flight method. Light impulses are sent and the camera measures the time of an impulse to come back to the camera from an object. By knowing the speed of light and the flight time of a light impulse to come back, the distance from the camera to the object can be calculated. 16
- ULOC** Uncommented Lines of Code (ULOC) is a software metric used to measure the programming effort put into a computer program by its uncommented source code line count. 45
- V-SLAM** Visual Simultaneous Localization and Mapping (V-SLAM) is a computational approach of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. The *V* highlights the usage of primarily visual (camera) sensors, because of the increasing ubiquity of cameras such as those in mobile devices. 19
- VC** Volume Clipping (VC) selectively disables or enables rendering operations inside a Region of Interest (RI), which is in practice predominantly the intersection of two or more overlapping objects. 35, 65, 150
- VHP** The Visible Human Project (VHP) is an effort to create a detailed data set of cross-sectional photographs of the human body, in order to facilitate anatomy visualization applications. 40, 52

WLAN A Wireless Local Area Network (WLAN) uses wireless communication to create a wireless computer network, linking two or more devices within an area determined by the WLAN signal strength. 39

Acronyms

- 1D** One Dimensional. 11, 12, 14, 161
- 2D** Two Dimensional. xi, xiii, 2, 11–14, 27–30, 35–39, 41, 42, 47, 51, 59, 63, 65, 76–78, 80, 91, 104, 109, 136, 138, 140, 152, 161, 173
- 3D** Three Dimensional. xi, xiii, 2–4, 6, 17–22, 24–30, 32, 35–38, 40–42, 44, 51, 52, 59, 65–67, 69, 72–74, 77, 87, 88, 91, 95–97, 99, 100, 125–127, 129, 136, 138, 140, 141, 144–146, 150–152, 162, 169, 170, 173, 174
- ABI** Application Binary Interface. 58, 89, *Glossary: ABI*
- AP** Anatomy Perspective. xv, 2–4, 49–56, 59, 61, 62, 65, 77, 79, 81, 82, 85, 93, 96, 114, 124, 129, 130, 137, 143, 145, 149
- API** Application Programming Interface. 40, 44, 45, 51, 53, 56, 58, 61, 81, 86–92, 94, 95, 100, 152, 169, 173, 174, *Glossary: API*
- AR** Augmented Reality. xi, xiii, 1–5, 7, 9–21, 23, 25–29, 31–41, 46, 47, 49, 51, 54, 57, 59–69, 73–80, 83, 84, 89, 90, 93–97, 100, 101, 103, 104, 106, 107, 109–112, 114, 115, 117, 119–121, 124–127, 129, 131–133, 136–147, 149, 150, 154, 161–164, 167, 169
- AV** Augmented Virtuality. 9, 10
- AVX** Advanced Vector eXtensions. *Glossary: AVX*
- BA** Bundle Adjustment. 20, *Glossary: BA*
- BB** Bounding Box. 43, 44, 58
- BE** Back-End. 5, 6, 20, 49, 61, 62, 65, 83, 92, 94, 95, 169, *Glossary: BE*
- BLOB** Binary Large Object. 62, 99
- BP3D** Body Parts 3D. xi, xiii, 1, 5–7, 40, 51–56, 61–63, 66, 68, 69, 79, 81–84, 95, 96, 98, 99, 124–126, 130, 145, 152, 155, 162, 163, 167, *Glossary: BP3D*
- BRIEF** Binary Robust Independent Elementary Features. 17, 19, 67, 90, 161, 173, 180, *Glossary: BRIEF*

CA C-Arm. 38, 39, 162, 170, *Glossary: CA*

CAD Computer Aided Design. 17

CCN Cascaded Convolutional Networks. 42, 43, 57, *Glossary: CCN*

CNN Convolutional Neuronal Networks. 42–44, 57, 152, *Glossary: CNN*

CPU Central Processing Unit. 20, 23, 45, 66, 152, 153, 173

CRUD Create, Read, Update and Delete. 9, 45, 46, 53, 54

CSV Comma Separated Values. 83, 155

CT Computed Tomography. 29, 32, 35, 36, 40, 50, 65, 66, 150, 151, 162, 163, 172, *Glossary: CT*

DB Database. 1, 3–7, 9, 13, 15, 27, 40, 44–46, 49, 51–57, 61–63, 65, 66, 74–77, 81–85, 92–95, 98–100, 104, 125, 127, 130, 131, 133, 134, 136, 150, 162, 174

DRY Don't Repeat Yourself. 94, 95

DVR Direct Volume Rendering. 35, *Glossary: DVR*

EAN European Article Numbering. 11

EMF Electromagnetic Field. 39, *Glossary: EMF*

ER Entity Relationship. 55, 162

F2S Files 2 SQLite. 84

FA Face Activity. 62, 63, 65–68, 70, 71, 73, 75, 77, 78, 85, 89, 93–95, 99, 100, 107–112, 114–119, 122–124, 129–131, 136, 137, 145, 162–165

FAB Floating Action Button. 64, 65, 67, 74, 75, 79, 107, 108, 110, 111, 132, 163, 164, *Glossary: FAB*

FAST Features from Accelerated Segment Test. 17, 19, 67, 90, 161, 173, 180, *Glossary: FAST*

FCVP Focus and Context Visualization Paradigm. 36, 162, *Glossary: FCVP*

FE Front-End. 5, 6, 20, 38, 49, 50, 61, 62, 65, 92, 94, 95, 170, *Glossary: FE*

FMA Foundational Model of Anatomy. xi, xiii, 1, 5–7, 52, 54–56, 62–64, 76, 78, 81–85, 93, 99, 107, 114, 118, 124, 125, 127, 130, 137, 155, 157, 162, 163, *Glossary: FMA*

FoV Field of View. 14–16, 161

FPS Frames Per Second. 24, 88, 100, *Glossary: FPS*

FTS Full Text Search. 85

GB Gigabyte. 45, 88

GPS Global Positioning System. 14–16

GPU Graphical Processing Unit. 23, 31, 36, 152, 153, 173

HA Hierarchy Activity. 62–67, 70, 74–78, 93, 94, 99, 100, 107–110, 114, 116–119, 122, 123, 126, 129, 131, 136, 137, 154, 162, 163, 165

HDR High Dynamic Range. 68, 100, 126, 163

HMD Head-Mounted Display. 38

HPE Head Pose Estimation. xi, xiii, 3, 4, 7, 9, 34, 36, 37, 40–44, 49, 51, 57–60, 63, 66, 73, 81, 85–91, 93, 95, 99, 103–105, 124, 129, 131, 144, 145, 151, 164

HSV Hue Saturation Value. 70, 71, 163, *Glossary: HSV*

ICP Iterative Closest Point. 20, *Glossary: ICP*

ID Identification. 48, 55, 82, 93, 125

IDE Integrated Development Environment. 48, 59, 94, *Glossary: IDE*

IE Informal Evaluation. 5–7, 103, 104, 106, 120, 129, 131, 132, 135, 138–140, 142, 144–147, 149, 154

IMU Inertial Measurement Unit. 24, *Glossary: IMU*

IQ Intelligence Quotient. 1, 150, 171

ISO International Organization for Standardization. 56

JNI Java Native Interface. 47, 58, 59, 86, 87, *Glossary: JNI*

KISS Keep It Simple, Stupid. 94

KLT Kanade-Lucas-Tomasi. 13

KV Key Value. 85, 172

LBA Living Book of Anatomy. 32, *Glossary: LBA*

LSD-SLAM Large Scale Direct - Simultaneous Localization and Mapping (LSD-SLAM). 17, 23, *Glossary: LSD-SLAM*

LSM Log Structured Merge. 85, *Glossary: LSM*

MB Mega Byte. 84, 88, 89, 104, 152

MK Microsoft Kinect. 32, 36, 172

MRI Magnetic Resonance Imaging. 29, 35, 36, 40, 50, 65, 66, 145, 150, 151, 162, 172, *Glossary: MRI*

NDK Native Development Kit. 47, 58, 59, 86, 87, 89, 152, *Glossary: NDK*

NMS Non-Maximum Suppression. 43, 44, *Glossary: NMS*

NNAPI Neural Networks API. 151, 152, *Glossary: NNAPI*

NNC Nearest Neighbour Classification. 41

OAB Open Anatomy Browser. 29

OAH Online Anatomical Human. 29, *Glossary: OAH*

OpenGL Open Graphics Library. 51, 100, *Glossary: OpenGL*

ORB-SLAM Oriented FAST and Rotated BRIEF Simultaneous Localization and Mapping. 17, 19–24, 39, 67, 90, 161, *Glossary: V-SLAM*

ORM Object Relational Mapping. 45, 46, 53, 54, *Glossary: FMA*

PEP Performance, Energy consumption, and Programming effort. 46

pixel picture element. 19, 23, 24, 34, 35, 41, 43, 44, 58, 80, 87, 173, 174

PK Primary Key. 55, 56, 83

PLK Pyramidal Lucas-Kanale. 24, *Glossary: PLK*

POJO Plain Old Java Object. *Glossary: POJO*

PR Point of Reference. 11, 52, 79, 93, 96, 97, 134, 139, 162

QR Quick Response. 11–13, 57, 161

RAM Random Access Memory. 45, 66, 88

RCT Randomized Controlled Trial. 141

RGB Red Green Blue. 35, 44, 70, 171

RGBD Red Green Blue Depth. 20, 23, 39, 145

RI Region of Interest. 33, 174

RNN Residual Neural Network. 34, *Glossary: RNN*

ROM Read Only Memory. 45

SA Single Activity. 62, 63, 65–67, 75–79, 94, 99–101, 107, 109–111, 117, 121–123, 126, 129, 131, 136–138, 145, 162–164

SD Scale Drift (SD). 21, 22, *Glossary: Scale Drift*

SDK Software Development Kit. 28, 46, 48, 81, *Glossary: SDK*

SE(3) Special Euclidean Group. *Glossary: SE(3)*

SFB SceneForm Binary. 61

SQLite Structured Query Language. 45, 46, 53–57, 61, 81–85, 94, 95, 157, *Glossary: SQLite*

SRC Sparse Representation Classification. 41, *Glossary: SRC*

surfel surface element. 17, *Glossary: surfel*

SVG Scalable Vector Graphics. 79, 80

ToF Time of Flight. 16, *Glossary: ToF*

UAI Universal Anatomy Index. 56, 57

UE User Experience. 2, 4, 6, 10, 12, 18, 61–63, 73, 89, 104, 106, 112, 136

UI User Interface. 4, 6, 28, 50, 58, 61, 64, 75, 79, 80, 99, 107, 110, 114, 121, 138, 139, 167

ULOC Uncommented Lines of Codes. 45, 46, *Glossary: ULOC*

UML Unified Modeling Language. 49, 65, 92, 162, 164

UP User Performance. 105, 138

UPC Universal Product Code. 11

UPS United Parcel Service. 11

V-SLAM Visual Simultaneous Localization and Mapping. 19–21, 23, 24, 38, 46, 47, *Glossary: V-SLAM*

VC Volume Clipping. 35, 36, 65, 150, 151, *Glossary: VC*

VHP Visible Human Project. 40, 52, *Glossary*: VHP

VR Virtual Reality. 1, 2, 9, 27, 30

WLAN Wireless Local Area Network. 39, *Glossary*: WLAN

XML eXtensible Markup Language. 103

Bibliography

- [1] Felix Bork. Interactive Augmented Reality Systems. *Der Unfallchirurg*, 121(4):286 – 292. Springer, April 2018.
- [2] Daewon Kim and Doosung Hwang. Non-Marker based Mobile Augmented Reality and its Applications using Object Recognition. *Journal of Universal Computer Science (J. UCS)*, 18(20):2832 – 2850. Verlag der Technischen Universität Graz, December 2012.
- [3] Ronald T. Azuma. A Survey of Augmented Reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355 – 385. MIT Press, August 1997.
- [4] Nobutaka Mitsuhashi, Kaori Fujieda, Takuro Tamura, Shoko Kawamoto, Toshihisa Takagi, and Kousaku Okubo. BodyParts3D: 3D Structure Database for Anatomical Concepts. *Nucleic Acids Research*, 37(1):782 – 785. Oxford University Press, October 2009.
- [5] Cornelius Rosse and José L. V. Mejino. The Foundational Model of Anatomy Ontology. *Anatomy Ontologies for Bioinformatics: Principles and Practice*, pages 59 – 117. Springer, December 2007.
- [6] Bernt Bratsberg and Ole Rogeberg. Flynn Effect and its Reversal are Both Environmentally Caused. *Proceedings of the National Academy of Sciences (PNAS)*, 115(26):6674 – 6678. National Academy of Sciences, June 2018.
- [7] James R. Flynn and Michael Shayer. IQ decline and Piaget: Does the Rot Start at the Top? *Intelligence*, 66(1):112 – 121. Elsevier, February 2018.
- [8] Adrian F. Ward, Kristen Duke, Ayelet Gneezy, and Maarten W. Bos. Brain Drain: The Mere Presence of One’s Own Smartphone Reduces Available Cognitive Capacity. *Journal of the Association for Consumer Research (JACR)*, 2(2):140 – 154. University of Chicago Press, April 2017.
- [9] Henry H. Wilmer, Lauren E. Sherman, and Jason M. Chein. Smartphones and Cognition: A Review of Research Exploring the Links between Mobile Technology Habits and Cognitive Functioning. *Frontiers in Psychology*, 8(3):605 – 621. Frontiers, October 2018.

- [10] Chaelin K. Ra, Junhan Cho, Matthew D. Stone, Julianne De La Cerda, Nicholas I. Goldenson, Elizabeth Moroney, Irene Tung, Steve S. Lee, and Adam M. Leventhal. Association of digital media use with subsequent symptoms of attention-deficit/hyperactivity disorder among adolescents. *Journal of the American Medical Association (JAMA)*, 320(3):255 – 263. American Medical Association, October 2018.
- [11] Shaghayegh Vahdat, Leila Hamzehgardeshi, Somayeh Hessam, and Zeinab Hamzehgardeshi. Patient Involvement in Health Care Decision Making: A Review. *Iranian Red Crescent Medical Journal*, 16(1):1 – 6. Iranian Hospital in Dubai, January 2014.
- [12] Dee Mangin, Gaibrie Stephen, Verdah Bismah, and Cathy Risdon. Making Patient Values Visible in Healthcare: A Systematic Review of Tools to Assess Patient Treatment Priorities and Preferences in the Context of Multimorbidity. *British Medical Journal (BMJ)*, 6(6). BMJ Publishing Group Ltd, June 2016.
- [13] Bernhard Preim and Patrick Saalfeld. A Survey of Virtual Human Anatomy Education Systems. *Computers & Graphics*, 71(1):132 – 153. Elsevier, April 2018.
- [14] John R. Fredieu, Jennifer Kerbo, Mark Herron, Ryan Klatte, and Malcolm Cooke. Anatomical Models: a Digital Revolution. *International Association of Medical Science Educators (IAMSE)*, 25(2):183 – 194. JulNet Solutions LLC, June 2015.
- [15] Robi Kelc. Zygote Body: A new Interactive 3-Dimensional Didactical Tool for Teaching Anatomy. *WebmedCentral ANATOMY 2012*, 3(1):1 – 14. Webmed Limited, January 2012.
- [16] Samy A. Azer and Sarah Azer. 3D Anatomy Models and Impact on Learning: A Review of the Quality of the Literature. *Health Professions Education*, 2(2):80 – 98. Elsevier, December 2016.
- [17] Susan Jang, Jonathan M. Vitale, Robert W. Jyung, and John B. Black. Direct Manipulation is Better than Passive Viewing for Learning Anatomy in a Three-Dimensional Virtual Reality Environment. *Computers & Education*, 106(1):150 – 165. Elsevier, March 2017.
- [18] Oxford Dictionary. Definition of short-term in English:. <https://www.lexico.com/en/definition/short-term>. Accessed Online: 06.12.2019.
- [19] Oxford Dictionary. Definition of medium-term in English:. <https://www.lexico.com/en/definition/medium-term>. Accessed Online: 06.12.2019.
- [20] Oxford Dictionary. Definition of long-term in English:. <https://www.lexico.com/en/definition/long-term>. Accessed Online: 06.12.2019.

- [21] Long Chen and Thomas Day and Wen Tang and Nigel W. John. Recent Developments and Future Challenges in Medical Mixed Reality. *16th Institute of Electrical and Electronics Engineers (IEEE) International Symposium on Mixed and Augmented Reality (ISMAR)*. Institute of Electrical and Electronics Engineers (IEEE), September 2017.
- [22] Google Inc. ARCore. <https://developers.google.com/ar>. Accessed Online: 14.05.2018.
- [23] Google Augmented Reality. Sceneform SDK for Android v1.7.0. <https://github.com/google-ar/sceneform-android-sdk/releases/tag/v1.7.0>. Accessed Online: 08.05.2019.
- [24] Bernhard Preim and Charl Botha. *Visual Computing for Medicine, Chapter e21 - Computer-Assisted Medical Education*. Elsevier, 2. edition, November 2014.
- [25] Steffen Oeltze-Jafra and Bernhard Preim. Survey of Labeling Techniques in Medical Visualizations. *Proceedings of the 4th Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM)*, pages 199 – 208. Eurographics, September 2014.
- [26] Carolien Kamphuis, Esther Barsom, Marlies Schijven, and Noor Christoph. Augmented Reality in Medical Education? *Perspectives on Medical Education*, 3(4):300 – 311. Springer, September 2014.
- [27] Phil Diegmann, Manuel Schmidt-Kraepelin, Sven van den Eynden, and Dirk Basten. Benefits of Augmented Reality in Educational Environments – A Systematic Literature Review. *International Conference on Wirtschaftsinformatik*, 3(6):1542 – 1556. Association for Information Systems, March 2015.
- [28] Joaquín Cubillo, Sergio Martín, Manuel Castro, Gabriel Diaz, Antonio Colmenar, and Ivica Botički. A learning environment for augmented reality mobile learning. *Frontiers in Education Conference (FIE) Proceedings*, pages 1 – 8. Wiley-IEEE Press, October 2014.
- [29] Tobias Blum, Valerie Kleeberger, Christoph Bichlmeier, and Nassir Navab. Miracle: An Augmented Reality Magic Mirror System for Anatomy Education. *Institute of Electrical and Electronics Engineers (IEEE) Virtual Reality Workshops (VRW)*, pages 115 – 116. Institute of Electrical and Electronics Engineers (IEEE), March 2012.
- [30] Aleksandr Amirkhanov, Artem Amirkhanov, Matthias Bernhard, Zsolt Toth, Sabine Stiller, Andreas Geier, and Meister Eduard Gröller. WithTeeth: Denture Preview in Augmented Reality. *Vision, Modeling and Visualization (VMV)*, pages 29 – 38. Eurographics, October 2018.

- [31] Noeska Smit, Cees-Willem Hofstede, Anne Kraima, Daniel Jansma, Marco Deruiter, Elmar Eisemann, and Anna Vilanova. The Online Anatomical Human: Web-based Anatomy Education. *Eurographics*, pages 37 – 40. Eurographics, May 2016.
- [32] Michael Halle, Valentin Demeusy, and Ron Kikinis. The Open Anatomy Browser: A Collaborative Web-Based Viewer for Interoperable Anatomy Atlases. *Frontiers in Neuroinformatics*, 11(1):1 – 22. Frontiers, March 2017.
- [33] Pere-Pau Vázquez, Timo Götzelmann, Knut Hartmann, and Andreas Nürnberger. An Interactive 3D Framework for Anatomical Education. *International Journal of Computer Assisted Radiology and Surgery (IJCARs)*, 3(6):511 – 524. Springer, December 2008.
- [34] Gilbert Cockton. *Encyclopedia of Human-Computer Interaction, Chapter 15 - Usability Evaluation*. Interaction Design Foundation, 2. edition, January 2013.
- [35] Google Inc. Google Java Style Guide. <https://google.github.io/styleguide/javaguide.html>. Accessed Online: 26.10.2019.
- [36] Google Inc. Let’s build the world’s most trusted source for apps and games. <https://play.google.com/about/developer-content-policy/>. Accessed Online: 26.10.2019.
- [37] Google Inc. Material Design. <https://material.io/guidelines>. Accessed Online: 07.05.2018.
- [38] Google Inc. Google Play. <https://play.google.com/store/apps?hl=en>. Accessed Online: 09.10.2018.
- [39] Threema GmbH. Threema. <https://play.google.com/store/apps/details?id=ch.threema.app>. Accessed Online: 15.01.2020.
- [40] Airbnb, Inc. Airbnb. <https://play.google.com/store/apps/details?id=com.airbnb.android>. Accessed Online: 15.01.2020.
- [41] Asana, Inc. Asana. <https://play.google.com/store/apps/details?id=com.asana.app>. Accessed Online: 15.01.2020.
- [42] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, and Grady Booch. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1. edition, November 1994.
- [43] Asa Macwilliams, Thomas Reicher, Gudrun Klinker, and Bernd Bruegge. Design Patterns for Augmented Reality Systems. *International Workshop Exploring the Design and Engineering of Mixed Reality Systems - MIXER*, pages 8 – 14. Technische Universität München, January 2004.

- [44] Zaixian Xie, Zhenyu Guo, Matthew O. Ward, and Elke A. Rundensteiner. Operator-Centric Design Patterns for Information Visualization Software. *Visualization and Data Analysis*, 7530(1):75300 – 75312. ResearchGate GmbH, January 2010.
- [45] Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. Empirical Studies in Information Visualization: Seven Scenarios. *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Visualization and Computer Graphics*, 18(9):1520 – 1536. Institute of Electrical and Electronics Engineers (IEEE), September 2012.
- [46] Stephanie L. Shipley, Jacqueline S. Stephen, and Andrew A. Tawfik. Revisiting the Historical Roots of Task Analysis in Instructional Design. *TechTrends: Organized by the Culture, Learning, and Technology*, 62(4):319 – 320. Association for Educational Communications & Technology (AECT), July 2018.
- [47] Hoshang Kolivand, Abdennour El Rhalibi, Sarmad Abdulazeez, and Pisit Praiwattana. *Cultural Heritage in Marker-Less Augmented Reality: A Survey*. Intech Open, 1. edition, November 2018.
- [48] Pixabay. Fractals Environment Chess. <https://pixabay.com/en/fractals-environment-chess-1728595/>. Accessed Online: 02.02.2019.
- [49] Pixabay. Woman Blonde Model Fashion Casual Jeans Denim. <https://pixabay.com/en/woman-blonde-model-fashion-casual-1031533>. Accessed Online: 02.02.2019.
- [50] Shao-Ning Chang and Wei-Lun Chen. Does Visualize Industries Matter? A Technology Foresight of Global Virtual Reality and Augmented Reality Industry. *International Conference on Applied System Innovation (ICASI)*, pages 382 – 385. Institute of Electrical and Electronics Engineers (IEEE), May 2017.
- [51] Nehla Ghouaiel, Jean-Marc Cieutat, and Jean-Pierre Jessel. Adaptive Augmented Reality: Plasticity of Augmentations. *Virtual Reality International Conference (VRIC)*, pages 10 – 14. International Association of Computer Science and Information Technology (IACSIT), April 2014.
- [52] Jens Grubert, Tobias Langlotz, Stefanie Zollmann, and Holger Regenbrecht. Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality. *Transactions on Visualization and Computer Graphics (TVCG)*, 23(6):1706 – 1724. Institute of Electrical and Electronics Engineers (IEEE), June 2017.
- [53] Oxford Dictionary. Definition of fiducial in English:. <https://www.lexico.com/en/definition/fiducial>. Accessed Online: 06.12.2019.
- [54] Pedro Q. Brito and Jasmina Stoyanova. Marker versus Markerless Augmented Reality. Which Has More Impact on Users? *International Journal of Human-Computer Interaction*, 34(9):819 – 833. Elsevier, November 2017.

- [55] Minhua Ma, Lakhmi Jain, and Paul Anderson. *Virtual, Augmented Reality and Serious Games for Healthcare*. Springer, 1. edition, August 2014.
- [56] Sanni Siltanen. *Theory and Applications of Marker-Based Augmented Reality*. Julkaisija, 3. edition, January 2012.
- [57] Seungho Chae, Jonghoon Seo, Yoonsik Yang, and Tack-Don Han. ColorCodeAR: Large Identifiable ColorCode-Based Augmented Reality System. *International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2598 – 2602. Institute of Electrical and Electronics Engineers (IEEE), October 2016.
- [58] Jo McCulloch. Free Barcode Stock Photo. <https://www.freeimages.com/photo/barcode-1517677>. Accessed Online: 28.01.2020.
- [59] Minh Nguyen and Albert Yeap. StereoTag: A novel stereogram-marker-based approach for Augmented Reality. *International Conference on Pattern Recognition (ICPR)*. Institute of Electrical and Electronics Engineers (IEEE), December 2016.
- [60] Vlasios Kasapakis, Damianos Gavalas, and Dzardanova Elena. Robust Outdoors Marker-Based Augmented Reality Applications: Mitigating the Effect of Lighting Sensitivity. *Augmented Reality, Virtual Reality, and Computer Graphics*, pages 423 – 431. Springer, July 2018.
- [61] Michiel Vlaminck, Hiep Luong, and Wilfried Philips. A Markerless 3D Tracking Approach for Augmented Reality Applications. *International Conference on 3D Imaging (IC3D)*, pages 1 – 7. Institute of Electrical and Electronics Engineers (IEEE), December 2017.
- [62] Raúl Mur-Artal, Jose M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Robotics*, 31(5):1147 – 1163. Institute of Electrical and Electronics Engineers (IEEE), October 2015.
- [63] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Robotics*, 33(5):1255 – 1262. Institute of Electrical and Electronics Engineers (IEEE), October 2017.
- [64] Yifan Wang and Huiliang Shang. Improved Monocular ORB-SLAM2 Inspired By The Optical Flow With Better Accuracy. *WRC Symposium on Advanced Robotics and Automation (SARA)*, pages 153 – 158. Institute of Electrical and Electronics Engineers (IEEE), August 2018.
- [65] Pengfei Wang, Chao Gao, Wei Guo, Mantian Li, Fusheng Zha, and Yanjie Cong. Robust and Fast Mapping Based on Direct Methods. *International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 592 – 595. Institute of Electrical and Electronics Engineers (IEEE), August 2017.

- [66] Xin Yang, Tangli Xue, Hongcheng Luo, and Jiabin Guo. Fast and Accurate Visual Odometry from a Monocular Camera. *Frontiers of Computer Science*, 13(6):1326 – 1336. Springer, December 2019.
- [67] Qing H. Gao, Tao R. Wan, Wen Tang, and Long Chen. A Stable and Accurate Marker-Less Augmented Reality Registration Method. *2017 International Conference on Cyberworlds (CW)*, pages 41 – 47. Eurographics, September 2017.
- [68] Edward Rosten and Tom Drummond. Machine Learning for High-Speed Corner Detection. *European Conference on Computer Vision (ECCV)*, pages 430 – 443. Springer, May 2006.
- [69] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. *International Conference on Computer Vision (ICCV)*, pages 2564 – 2571. Institute of Electrical and Electronics Engineers (IEEE), November 2011.
- [70] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. *European Conference on Computer Vision (ECCV)*, pages 778 – 792. Springer, September 2010.
- [71] Raúl Mur Artal. ORB-SLAM in the KITTI dataset (Sequence 00). <https://youtu.be/8DISRmsO2YQ?&list=PL11-11>. Accessed Online: 21.01.2019.
- [72] Shelby J. Haberman and Neil J. Dorans. Scale Consistency, Drift, Stability: Definitions, Distinctions and Principles. *Annual Meeting*, pages 1 – 8. American Educational Research Association (AERA), April 2009.
- [73] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. *European Conference on Computer Vision (ECCV)*, pages 834 – 849. Springer, September 2014.
- [74] Armelle Bauer, Ali-Hamadi Dicko, Olivier Palombi, François Faure, and Jocelyne Troccaz. Living Book of Anatomy (LBA) Project: See Your Insides in Motion! *Special Interest Group on Computer GRAPHics and Interactive Techniques (SIG-GRAPH)*, pages 16 – 19. Association for Computing Machinery (ACM), November 2015.
- [75] Kaissar Yammine and Claudio Violato. A Meta-Analysis of the Educational Effectiveness of Three-Dimensional Visualization Technologies in Teaching Anatomy. *Anatomical Sciences Education*, 8(1):525 – 538. American Association for Anatomy, December 2014.
- [76] Jorge Camba, Manuel Contero, and Gustavo Salvador-Herranz. Desktop vs. Mobile: A Comparative Study of Augmented Reality Systems for Engineering Visualizations in Education. *Frontiers in Education Conference (FIE) Proceedings*, pages 1 – 8. Wiley-IEEE Press, October 2014.

- [77] Francisco Munoz-Montoya, Juan M. Carmen, Magdalena Mendez-Lopez, and Camino Fidalgo. Augmented Reality Based on SLAM to Assess Spatial Short-Term Memory. *Institute of Electrical and Electronics Engineers (IEEE) Access*, 7(1):2453 – 2466. Institute of Electrical and Electronics Engineers (IEEE), December 2019.
- [78] Peter Sommerauer and Oliver Müller. Augmented Reality in Informal Learning Environments: Investigating Short-term and Long-term Effects. *Proceedings of the 51st Hawaii International Conference on System Sciences*, pages 1423 – 1430. Curran Associates, Inc., January 2018.
- [79] Christoph Noll, Ute von Jan, Ulrike Raap, and Urs-Vito Albrecht. Mobile Augmented Reality as a Feature for Self-Oriented, Blended Learning in Medicine: Randomized Controlled Trial. *Journal of Medical Internet Research (JMIR)*, 5(9):139 – 153. JMIR Publications, September 2017.
- [80] Visible Body. We Create Apps that Advance the Way Students Learn and Healthcare Professionals Practice. <https://www.visiblebody.com>. Accessed Online: 07.05.2018.
- [81] 3D 4 Medical. Transforming Medical Learning; Experience the World’s Most Advanced Medical Education Platform. <https://3d4medical.com>. Accessed Online: 07.05.2018.
- [82] eMedia Interactive Ltd. Pocket Anatomy, The Interactive Human Body. <https://www.pocketanatomy.com>. Accessed Online: 07.05.2018.
- [83] Jeanette Mönch, Konrad Mühler, Christian Hansen, Karl-Jürgen Oldhafer, Gregor Stavrou, Christian Hillert, Christoph Logge, and Bernhard Preim. The Liver-SurgeryTrainer: Training of Computer-Based Planning in Liver Resection Surgery. *International Journal of Computer Assisted Radiology and Surgery (IJCARS)*, 8(5):809 – 818. Springer, January 2013.
- [84] Patrick Saalfeld, Steffen Oeltze-Jafra, Sylvia Saalfeld, Uta Preim, Oliver Beuing, and Bernhard Preim. Sketching and Annotating Vascular Structures to Support Medical Teaching, Treatment Planning and Patient Education. *2017 Dirk Bartz Prize for Visual Computing in Medicine*, pages 5 – 8. Eurographics, March 2017.
- [85] Bernhard Preim and Charl Botha. *Visual Computing for Medicine, Chapter 12 - Illustrative Medical Visualization*. O’Reilly Media, Inc., 2. edition, November 2014.
- [86] Stefan Bruckner. *Interactive Illustrative Volume Visualization*. PhD thesis, March 2008.
- [87] Bernhard Preim and Charl Botha. *Visual Computing for Medicine, Chapter 18 - Image-Guided Surgery and Augmented Reality*. pages 625 – 663. Elsevier, November 2014.

- [88] Alexander Mathis, Pranav Mamidanna, Kevin Cury, Taiga Abe, Venkatesh Murthy, Mackenzie Mathis, and Matthias Bethge. DeepLabCut: Markerless Tracking of User-Defined Features With Deep Learning. *Nature Neuroscience*, 21(9):1281 – 1289. Nature Publishing Group, September 2018.
- [89] Márcio Cerqueira de Farias Macedo and Antônio Lopes Apolinário Junior. Improving On-Patient Medical Data Visualization in a Markerless Augmented Reality Environment by Volume Clipping. *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 149 – 156. Elsevier, August 2014.
- [90] Jonas Hajek, Mathias Unberath, Javad Fotouhi, Bastian Bier, Sing Chun Lee, Greg Osgood, Andreas Maier, Mehran Armand, and Nassir Navab. Closing the Calibration Loop: An Inside-Out-Tracking Paradigm for Augmented Reality in Orthopedic Surgery. *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 299 – 306. Springer, September 2018.
- [91] Gary Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, pages 1 – 6. UBM Technology Group, November 2000.
- [92] Davis E. King. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research (JMLR)*, 10(1):1755 – 1758. Microtome Publishing, 2009.
- [93] Epic Games, Inc. Unreal Engine. <https://www.unrealengine.com/en-US/>. Accessed Online: 20.01.2020.
- [94] Microsoft, Inc. HoloLens 2 - A new vision for computing. <https://www.microsoft.com/en-us/hololens>. Accessed Online: 20.01.2020.
- [95] Nicolas Loy Rodas, Fernando Barrera, and Nicolas Padoy. See It With Your Own Eyes: Markerless Mobile Augmented Reality for Radiation Awareness in the Hybrid Room. *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Biomedical Engineering*, 64(2):429 – 440. Institute of Electrical and Electronics Engineers (IEEE), February 2017.
- [96] Hans Geiger and Walther Müller. Technische Bemerkungen zum Elektronenzählrohr. *Physikalische Zeitschrift*, 30(1):489 – 493. S. Hirzel Verlag, August 1929.
- [97] Digimation. The Archive. <https://digimation.com/the-archive>. Accessed Online: 07.05.2018.
- [98] Michael J. Ackerman, Victor M. Spitzer, Ann L. Scherzinger, and David G. Whitlock. The Visible Human Data Set: An Image Resource for Anatomical Visualization. *Medinfo*, 8(2):1195 – 1198. International Medical Informatics Association (IMIA), January 1995.
- [99] Trubosquid. Anatomy 3D Models. <https://www.turbosquid.com/3d-model/anatomy>. Accessed Online: 07.05.2018.

- [100] John Qualter, Frank Sculli, Aaron Olikier, Zachary Napier, Sabrina Lee, Julio Garcia, Sally Frenkel, Victoria Harnik, and Marc Triola. The Biodigital Human: A Web-Based 3D Platform for Medical Visualization and Education. *Studies in Health Technology and Informatics*, 173(1):359 – 361. IOS Press, January 2012.
- [101] Yiran Shen, Mingrui Yang, Bo Wei, Chun T. Chou, and Wen Hu. Learn to Recognise: Exploring Priors of Sparse Face Recognition on Smartphones. *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Mobile Computing*, 16(6):1705 – 1717. Institute of Electrical and Electronics Engineers (IEEE), June 2017.
- [102] Muhammad Adi Nugroho and Benyamin Kusumoputro. Fuzzy-Appearance Manifold and Fuzzy Nearest Distance for Face Recognition on Various Poses and Degraded Images. *15th International Conference on Quality in Research (QiR)*, pages 342 – 346. Institute of Electrical and Electronics Engineers (IEEE) Indonesia Section, July 2017.
- [103] Kornél Bertók and Attila Fazekas. Face Recognition on Mobile Platforms. *7th Institute of Electrical and Electronics Engineers (IEEE) International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 37 – 42. Institute of Electrical and Electronics Engineers (IEEE), October 2016.
- [104] Lulian-Ionut Felea, Laura Florea, Corneliu Florea, and Constantin Vert-An. Head Pose Estimation Using Deep Architectures. *International Conference on Communications (COMM)*, pages 505 – 508. Institute of Electrical and Electronics Engineers (IEEE), June 2018.
- [105] Zhenni Cai and Qingshan Liu and Shanmin Wang and Bruce Yang. Joint Head Pose Estimation with Multi-task Cascaded Convolutional Networks for Face Alignment. *International Conference on Pattern Recognition (ICPR)*, pages 495 – 500. Institute of Electrical and Electronics Engineers (IEEE), August 2018.
- [106] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICLR)*, pages 1 – 14. DBLP Computer Science Bibliography, September 2014.
- [107] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770 – 778. Institute of Electrical and Electronics Engineers (IEEE), June 2016.
- [108] Hao Wu and Ke Zhang and Guohui Tian. Simultaneous Face Detection and Pose Estimation Using Convolutional Neural Network Cascade. *Asian Conference on Computer Vision (ACCV)*, pages 187 – 202. Springer, December 2018.
- [109] Chris J. Date. *Database in Depth*. O’Reilly, 1. edition, June 2009.

- [110] Zheng Jason Song, Jing Pu, Junjie Cheng, and Eli Tilevich. Performance and Programming Effort Trade-offs of Android Persistence Frameworks. *Journal of Systems and Software*, 146(1):99 – 111. Elsevier, June 2018.
- [111] Stephen M. Blackburn, Robin Garner, Chris Hoffmann, Asjad M. Khan, Kathryn S. McKinley, Rotem Bentzur, Amer Diwan, Daniel Feinberg, Daniel Frampton, Samuel Z. Guyer, Martin Hirzel, Antony Hosking, Maria Jump, Han Lee, Eliot B. Moss, Aashish Phansalkar, Darko Stefanovic, Thomas Van Drunen, Daniel von Dincklage, and Ben Wiedermann. The DaCapo Benchmarks: Java Benchmarking Development and Analysis. *Object-Oriented Programming, Systems and Languages and Applications (OOPSLA)*, pages 169 – 190. Association for Computing Machinery (ACM), October 2006.
- [112] Jerome Etienne. AR.js. <https://github.com/jeromeetienne/AR.js>. Accessed Online: 14.05.2018.
- [113] Apple, Inc. Augmented Reality. <https://developer.apple.com/augmented-reality/>. Accessed Online: 21.01.2020.
- [114] Hirokazu Kato, Mark Billingham, and Ivan Poupyrev. ARToolkit. <https://github.com/artoolkitx>. Accessed Online: 21.01.2020.
- [115] VisionStar Information Technology Co., Ltd. EasyAR. <https://www.easyar.com>. Accessed Online: 21.01.2020.
- [116] VisionStar Information Technology Co., Ltd. Kudan. <https://www.kudan.io>. Accessed Online: 21.01.2020.
- [117] Progetti di Impresa. Pikkart. <http://www.pikkart.com>. Accessed Online: 21.01.2020.
- [118] PTC Inc. Vuforia. <https://www.vuforia.com>. Accessed Online: 14.05.2018.
- [119] Wikitude GmbH. Wikitude. <https://www.wikitude.com>. Accessed Online: 21.01.2020.
- [120] Iulia Marneanu, Martin Ebner, and Thomas Rößler. Evaluation of Augmented Reality Frameworks for Android Development. *International Journal of Interactive Mobile Technologies (iJIM)*, 8(4):37 – 44. Public Knowledge Project (PKP), October 2014.
- [121] Fabrício Herpich, Renan Guarese, and Liane Tarouco. A Comparative Analysis of Augmented Reality Frameworks Aimed at the Development of Educational Applications. *Creative Education*, 8(9):1433 – 1451. Scientific Research Publishing Inc., July 2017.

- [122] Anna Poth. Evaluierung von Augmented Reality Frameworks anhand einer AR-Anwendung im IoT-Bereich. *Computer Science and Media on Research Gate*, pages 1 – 10. ResearchGate GmbH, August 2017.
- [123] Victoria Rautenbach, Serena Coetzee, and Danie Jooste. Results of an Evaluation of Augmented Reality Mobile Development Frameworks for Addresses in Augmented Reality. *Spatial Information Research*, 24(3):211 – 223. Springer, June 2016.
- [124] Socialcompare. Augmented Reality SDK Comparison. <http://socialcompare.com/en/comparison/augmented-reality-sdks>. Accessed Online: 02.02.2019.
- [125] Sanket Prabhu. Augmented Reality SDK Comparison: Parameters to select the right AR SDK. <http://www.arreverie.com/blogs/augmented-reality-sdk-comparison-parameters>. Accessed Online: 03.02.2019.
- [126] icd.com. Smartphone Market Share. <https://www.idc.com/promo/smartphone-market-share/os>. Accessed Online: 06.02.2019.
- [127] SQLite. 35% Faster Than The Filesystem. <https://sqlite.org/fasterthanfs.html>. Accessed Online: 14.11.2018.
- [128] Jing Pu, Zheng Song, and Eli Tilevich. Understanding the Energy, Performance, and Programming Effort Trade-Offs of Android Persistence Frameworks. *International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 433 – 438. Institute of Electrical and Electronics Engineers (IEEE), September 2016.
- [129] Pierre-Luc Manteaux, Chris Wojtan, Rahul Narain, Stéphane Redon, François Faure, and Marie-Paule Cani. Adaptive Physically Based Models in Computer Graphics. *Computer Graphics Forum*, 36(6):312 – 337. John Wiley & Sons, Inc., September 2017.
- [130] Android Developers. Save Data using SQLite. <https://developer.android.com/training/data-storage/sqlite#java>. Accessed Online: 14.11.2018.
- [131] Vahid Kazemi and Josephine Sullivan. One Millisecond Face Alignment With an Ensemble of Regression Trees. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1867 – 1874. Institute of Electrical and Electronics Engineers (IEEE), June 2014.
- [132] Manuel T. Schrempf. 1.9.0 Sceneform’s Augmented Face Mesh Indices From Their .fbx File as Pictures. <https://github.com/ManuelTS/augmentedFaceMeshIndices>. Accessed Online: 26.05.2019.

- [133] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291 – 7299. Institute of Electrical and Electronics Engineers (IEEE), July 2017.
- [134] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand Keypoint Detection in Single Images using Multiview Bootstrapping. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4645 – 4653. Institute of Electrical and Electronics Engineers (IEEE), July 2017.
- [135] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional Pose Machines. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724 – 4732. Institute of Electrical and Electronics Engineers (IEEE), June 2016.
- [136] Ke Sun, Cuiling Lan, Junliang Xing, Wenjun Zeng, Dong Liu, and Jingdong Wang. Human Pose Estimation Using Global and Local Normalization. *International Conference on Computer Vision (ICCV)*, pages 5600 – 5608. Institute of Electrical and Electronics Engineers (IEEE), October 2017.
- [137] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A Simple Yet Effective Baseline for 3D Human Pose Estimation. *International Conference on Computer Vision (ICCV)*, pages 2659 – 2668. Institute of Electrical and Electronics Engineers (IEEE), October 2017.
- [138] Hassam Alikarami, Farzin Yaghmaee, and Mohammad J. Fadaeieslam. Sparse Representation and Convolutional Neural Networks for 3D Human Pose Estimation. *Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*, pages 188 – 192. Amirkabir University of Technology, December 2017.
- [139] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial PoseNet: A Structure-Aware Convolutional Network for Human Pose Estimation. *International Conference on Computer Vision (ICCV)*, pages 1221 – 1230. Institute of Electrical and Electronics Engineers (IEEE), October 2017.
- [140] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-Shot Multi-Person 3D Body Pose Estimation From Monocular RGB. *International Conference on 3D Vision (3DV)*, pages 1 – 16. Institute of Electrical and Electronics Engineers (IEEE), August 2018.
- [141] Johannes W. Rohen, Chihiro Yokochi, and Elke Lütjen-Drecoll. *Anatomie. Der fotografische Atlas der systematischen und topografischen Anatomie des Menschen*. Schattauer, 8. edition, September 2015.

- [142] Manuel T. Schrempf. The .obj Converter fails since version 1.7.0. <https://github.com/google-ar/sceneform-android-sdk/issues/613>. Accessed Online: 19.10.2019.
- [143] Franz Fellner. Introducing Cinematic Rendering: A Novel Technique for Post-Processing Medical Imaging Data. *Journal of Biomedical Science and Engineering (JBiSE)*, 9(3):170 – 175. Scientific Research Publishing Inc., January 2016.
- [144] Google Inc. ARCore SDK for Android v1.10.0. <https://github.com/google-ar/arcore-android-sdk/releases/tag/v1.10.0>. Accessed Online: 26.10.2019.
- [145] Benjamin Schrom and Leon Wong and Christina Tong. What’s New in ARCore (Google I/O’19). <https://youtu.be/pdRpCJ5acrA?t=1768>. Accessed Online: 30.11.2019.
- [146] Vipin Garsa, Neeru Ghalawat, Ravi K. Narayan, and Suresh K. Rathee. Teaching Anatomy Using Colored Models. *International Journal of Advanced Research (IJAR)*, 3(6):282 – 284. 2017.
- [147] Philipp Stefan, Patrick Wucherer, Yuji Oyamada, Meng Ma, Alexander Schoch, Motoko Kanegae, Naoki Shimizu, Tatsuya Kodera, Sebastien Callier, Matthias Weigl, Maki Sugimoto, Pascal Fallavollita, Hideo Saito, and Nassir Navab. An AR Edutainment System Supporting Bone Anatomy Learning. *Institute of Electrical and Electronics Engineers (IEEE) Virtual Reality (VR)*, pages 113 – 114. Institute of Electrical and Electronics Engineers (IEEE), March 2014.
- [148] Daniel Pohlandt, Bernhard Preim, and Patrick Saalfeld. Supporting Anatomy Education with a 3D Puzzle in a Virtual Reality Environment - Results from a Pilot Study. *Proceedings of Mensch und Computer (MuC’19)*, pages 91 – 102. Association for Computing Machinery (ACM), September 2019.
- [149] Using the Android Sharesheet. Wikitude. <https://developer.android.com/training/sharing/send.html>. Accessed Online: 21.01.2020.
- [150] Andrew Elliot and Markus Maier. Color Psychology: Effects of Perceiving Color on Psychological Functioning in Humans. *Annual Review of Psychology*, 65(1):95 – 120. Annual Review, June 2013. PMID: 23808916.
- [151] Kentaro Hamada, Kaori Yoshida, Kei Ohnishi, and Mario Koppen. Color Effect on Subjective Perception of Progress Bar Speed. *International Conference on Intelligent Networking and Collaborative Systems*, pages 863 – 866. Springer, November 2011.
- [152] Google Inc. AnimatedVectorDrawable. <https://developer.android.com/reference/android/graphics/drawable/AnimatedVectorDrawable>. Accessed Online: 25.01.2020.

- [153] Alesha Unpingco and Alex Faaborg. Best Practices to Design AR Applications (Google I/O '18). <https://youtu.be/bNJJCREZgVM?t=14m31s>. Accessed Online: 24.11.2018.
- [154] Michael Halle. Foundational Model of Anatomy in an SQLite database. <https://github.com/mhalle/fma-sqlite>. Accessed Online: 13.10.2019.
- [155] Manuel T. Schrempf. List Index out of Range error. <https://github.com/mhalle/fma-sqlite/issues/2>. Accessed Online: 15.10.2019.
- [156] SharkD. Warning about invalid entries. <https://github.com/ManuelTS/fma2sqlite>. Accessed Online: 18.11.2019.
- [157] Manuel T. Schrempf. FMA 2 SQLite. <https://github.com/ManuelTS/fma2sqlite>. Accessed Online: 13.10.2019.
- [158] Manuel T. Schrempf. Convert mutliple .obj to .sfa and .sfb. <https://github.com/google-ar/sceneform-android-sdk/issues/217>. Accessed Online: 30.11.2019.
- [159] Manuel T. Schrempf. Files 2 SQLite (F2S). <https://github.com/ManuelTS/f2s>. Accessed Online: 14.11.2018.
- [160] Tukaani Developers. XZ-Utils. <https://tukaani.org/xz>. Accessed Online: 14.11.2018.
- [161] SQLite. SQLite FTS5 Extension. <https://www.sqlite.org/fts5.html>. Accessed Online: 14.11.2018.
- [162] Yuanjing Shi, Zhaoyan Shen, and Zili Shao. SQLiteKV: An Efficient LSM-Tree-Based SQLite-Like Database Engine for Mobile Devices. *23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 28 – 33. Institute of Electrical and Electronics Engineers (IEEE) Council on Electronic Design Automation (CEDA), January 2018.
- [163] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S. Huang. Interactive Facial Feature Localization. *European Conference on Computer Vision (ECCV)*, pages 679 – 692. Springer, October 2012.
- [164] Manuel T. Schrempf. xlib11-dev, libopenblas-dev, and liblapack-dev installed but not found. <https://github.com/davisking/dlib/issues/1621>. Accessed Online: 13.11.2019.
- [165] Jim Gray. Why Do Computers Stop and What Can Be Done About It. pages 1 – 46. Tandem Computers, Corporate Information Center, Juni 1985.
- [166] Google Inc. Detect faces. <https://cloud.google.com/vision/docs/detecting-faces>. Accessed Online: 18.10.2019.

- [167] Diomidis Spinellis. Code Documentation. *Institute of Electrical and Electronics Engineers (IEEE) Software*, 27(4):18 – 19. Institute of Electrical and Electronics Engineers (IEEE), July 2010.
- [168] Jeffrey Kotula. Source Code Documentation: An Engineering Deliverable. *Proceedings 34th International Conference on Technology of Object-Oriented Languages and Systems*, pages 505 – 507. Association for Computing Machinery (ACM), August 2000.
- [169] Manuel T. Schrempf. Is it possible to get documentation in *Android Studio*? <https://github.com/google-ar/arcore-android-sdk/issues/733>. Accessed Online: 11.11.2019.
- [170] Jon Bentley. Programming Pearls. *Communications of the Association for Computing Machinery (CACM)*, 28(9):896 – 901. Association for Computing Machinery (ACM), September 1985.
- [171] Google Inc. Activity. <https://developer.android.com/reference/android/app/Activity>. Accessed Online: 23.10.2019.
- [172] Manuel T. Schrempf. Render models behind the AugmentedFace Mesh inside the head. <https://github.com/mhalle/fma-sqlite/issues/768>. Accessed Online: 25.10.2019.
- [173] Eve Hoggan, Miguel Nacenta, Per Kristensson, John Williamson, Antti Oulasvirta, and Anu Lehtiö. Multi-touch Pinch Gestures: Performance and Ergonomics. *Convergence of Interactive Tabletops and Surfaces (ITS)*, pages 219 – 222. Association for Computing Machinery (ACM), October 2013.
- [174] michaelvogt. Custom material. <https://github.com/google-ar/sceneform-android-sdk/issues/580>. Accessed Online: 13.11.2019.
- [175] Timo Ropinski, Jörg-Stefan Praßni, Jan Roters, and Klaus Hinrichs. Internal Labels as Shape Cues for Medical Illustration. *Vision, Modeling and Visualization (VMV)*, 7(1):203 – 212. Eurographics, January 2007.
- [176] emetsds. [device support] Nokia 7 Plus (Android 9) is not supported. <https://github.com/google-ar/arcore-android-sdk/issues/764>. Accessed Online: 30.11.2019.
- [177] Google Inc. Start an activity with a shared element. <https://developer.android.com/training/transitions/start-activity#start-with-element>. Accessed Online: 19.11.2019.
- [178] Manuel T. Schrempf. Video stream changes, background black and white in relation to distance. <https://github.com/google-ar/sceneform-android-sdk/issues/387>. Accessed Online: 30.10.2019.

- [179] Chris Kelly, Elly Nattinger, and Luca Prasso. Exploring AR interaction (Google I/O '18). <https://youtu.be/bUGhG-AZpu0?t=800>. Accessed Online: 30.10.2019.
- [180] Alicia O'Cathain and Kate Thomas. "Any other comments?" Open Questions on Questionnaires - A Bane or a Bonus to Research? *BMC Medical Research Methodology*, 4(1):25 – 32. Springer Nature, December 2004.
- [181] Eleanor Singer and Mick P. Couper. Some Methodological Uses of Responses to Open Questions and Other Verbatim Comments in Quantitative Surveys. *Methods, Data, Analyses (MDA)*, 11(2):115 – 134. GESIS, January 2017.
- [182] Dr. Angelina Whalley. Körperwelten. <https://koerperwelten.de>. Accessed Online: 01.12.2019.
- [183] Antonella Poce, Francesca Amenduni, Carlo De Medio, Mara Valente, and Maria Rosaria Re. Adopting Augmented Reality to Engage Higher Education Students in a Museum University Collection: the Experience at Roma Tre University. *Visual Pattern Extraction and Recognition for Cultural Heritage Understanding (VIPERC)*, 10(12):373 – 384. Springer, November 2019.
- [184] Felix Bork, Leonard Stratmann, Stefan Enssle, Ulrich Eck, Nassir Navab, Jens Waschke, and Daniela Kugelmann. The Benefits of an Augmented Reality Magic Mirror System for Integrated Radiology Teaching in Gross Anatomy. *Anatomical Sciences Education*, 12(6):585 – 598. John Wiley & Sons, Inc., 2019.
- [185] Sumitra Nuanmeesri, Preedawon Kadmateekarun, and Lap Poomhiran. Augmented Reality to Teach Human Heart Anatomy and Blood Flow. *Turkish Online Journal of Educational Technology (TOJET)*, 18(1):15 – 24. The Turkish Online Journal of Educational Technology, January 2019.
- [186] Martin V. Pusic, Kathy Boutis, Rose Hatala, and David A. Cook. Learning Curves in Health Professions Education. *Academic Medicine*, 90(8):1034 – 1042. Association of American Medical Colleges, August 2015.
- [187] Sevda Küçük, Samet Kapakin, and Yüksel Göktaş. Learning Anatomy via Mobile Augmented Reality: Effects on Achievement and Cognitive Load. *Anatomical Sciences Education*, 9(5):411 – 421. American Association for Anatomy, March 2016.
- [188] Irwin Silverman, Jean Choi, and Michael Peters. The Hunter-Gatherer Theory of Sex Differences in Spatial Abilities: Data from 40 Countries. *Archives of Sexual Behavior*, 36(2):261 – 268. Springer, March 2007.
- [189] Isabelle D. Cherney, Kyle Bersted, and Joseph Smetter. Training Spatial Skills in Men and Women. *Perceptual and Motor Skills*, 119(1):82 – 99. SAGE Publications Inc., August 2014.

- [190] Angus Deaton and Nancy Cartwright. Understanding and Misunderstanding Randomized Controlled Trials. *Social Science & Medicine*, 210(1):2 – 21. Elsevier, August 2018.
- [191] Hakkun Elmunsyah, Wahyu N. Hidayat, and Khoirudin Asfani. Influence of Augmented Reality on the Achievement and Student Learning Independence. *International Seminar on Application for Technology of Information and Communication (iSemantic)*, pages 559 – 563. Institute of Electrical and Electronics Engineers (IEEE), September 2019.
- [192] Breanne K. Litts and Whitney E. Lewis. Mobile Augmented Reality: Exploring a New Genre of Learning. *GetMobile: Mobile Computing and Communications*, 22(3):5 – 9. Association for Computing Machinery (ACM), January 2019.
- [193] Wu Cai and Qingtao Chen. An Experimental Research of Augmented Reality Technology from the Perspective of Mobile Learning. *International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 912 – 915. Institute of Electrical and Electronics Engineers (IEEE), December 2018.
- [194] Monika Pölönen, Toni Järvenpää, and Jukka Häkkinen. Reading E-Books on a Near-To-Eye Display: Comparison Between a Small-Sized Multimedia Display and a Hard Copy. *Displays*, 33(3):157 – 167. Elsevier, July 2012.
- [195] Meredith David and James Roberts. Phubbed and Alone: Phone Snubbing, Social Exclusion, and Attachment to Social Media. *Journal of the Association for Consumer Research (JACR)*, 2(2):155 – 163. University of Chicago Press, March 2017.
- [196] Aleksandar Dodevski, Strahil Gazepov, Biljana Dodevska, and Alen Gorgiev. Myopia in Pre-School Children and Mobile Phones. *Knowledge - International Journal*, 31(4):991 – 993. Institute of Knowledge Management, June 2019.
- [197] Ben Turney. Anatomy in a Modern Medical Curriculum. *The Annals of the Royal College of Surgeons of England*, 89(2):104 – 107. The Royal College of Surgeons of England, March 2007.
- [198] David Lempert. The "Newspeak" Dictionary of "Development": Deconstructing Development Discourse and Calling Things by Their True Names. *Transcience: Journal of Global Studies*, 6(2):38 – 52. Humboldt-Universität zu Berlin, August 2015.
- [199] Meilan Zhang, Robert P. Trussell, Daniel A. Tillman, and Song A. An. Tracking the Rise of Web Information Needs for Mobile Education and an Emerging Trend of dDigital Divide. *Computers in the Schools*, 32(2):83 – 104. Taylor & Francis, June 2015.

- [200] Felix Bork, Alexander Lehner, Daniela Kugelmann, Ulrich Eck, Jens Waschke, and Nassir Navab. VesARlius: An Augmented Reality System for Large-Group Co-located Anatomy Learning. *International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 122 – 123. Institute of Electrical and Electronics Engineers (IEEE), January 2020.
- [201] Daniela Kugelmann, Leonard Stratmann, Nils Nühlen, Felix Bork, Saskia Hoffmann, Golbarg Samarbarksh, Anna Pferschy, Anna Maria von der Heide, Andreas Eimannsberger, Pascal Fallavollita, Nassir Navab, and Jens Waschke. An Augmented Reality Magic Mirror as Additive Teaching Device for Gross Anatomy. *Annals of Anatomy*, 215(1):71 – 77. Elsevier, January 2018.
- [202] Joseph Henrich, Steven J. Heine, and Ara Norenzayan. The Weirdest People in the World? *Behavioral and Brain Sciences*, 33(2):61 – 83. Cambridge University Press, June 2010.
- [203] Juan A. Juanes, Daniel Hernández, Pablo Ruisoto, Elena García, Gabriel Villarrubia, and Alberto Prats. Augmented Reality Techniques, Using Mobile Devices, for Learning Human Anatomy. *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM)*, pages 7 – 11. Association for Computing Machinery (ACM), October 2014.
- [204] Ching-Sheng Wang, Shih-Hui Hung, and Ding-Jung Chiang. A Markerless Augmented Reality Mobile Navigation System with Multiple Targets Display Function. *International Conference on Applied System Innovation (ICASI)*, pages 408 – 411. Institute of Electrical and Electronics Engineers (IEEE), May 2017.
- [205] Google Inc. Neural Networks API. <https://developer.android.com/ndk/guides/neuralnetworks>. Accessed Online: 07.05.2019.
- [206] Pontificia Universidade Católica do Rio de Janeiro. Lua. <https://www.lua.org>. Accessed Online: 07.05.2019.
- [207] roseborough.com. LuaJ API. <http://lua.j.org/lua.j.html>. Accessed Online: 07.05.2019.
- [208] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning Feature Pyramids for Human Pose Estimation. *International Conference on Computer Vision (ICCV)*, pages 1290 – 1299. Institute of Electrical and Electronics Engineers (IEEE), Oct 2017.
- [209] Google Inc. RenderScript Overview. <https://developer.android.com/guide/topics/renderscript/compute>. Accessed Online: 08.05.2019.
- [210] Dicko Ali-Hamadi, Tiantian Liu, Benjamin Gilles, Ladislav Kavan, François Faure, Olivier Palombi, and Marie-Paule Cani. Anatomy Transfer. *Transactions on Graphics (TOG)*, 32(6):188 – 196. Association for Computing Machinery (ACM), November 2013.

- [211] Enrico Tanuwidjaja, Derek Huynh, Kirsten Koa, Calvin Nguyen, Churen Shao, Patrick Torbett, Colleen Emmenegger, and Nadir Weibel. Chroma: A Wearable Augmented-reality Solution for Color Blindness. *International Joint Conference on Pervasive and Ubiquitous Computing (UBICOMP)*, pages 799 – 810. Springer, September 2014.
- [212] João Marcelo Xavier Natário Teixeira, Lucas Figueiredo, Lucas Maggi, Veronica Teichrieb, Marcel Santos, and Cristiano Araújo. An Analytics Framework for Augmented Reality Applications. *SBC Journal on Interactive Systems*, 9(2):26 – 37. Brazilian Computing Society, June 2018.
- [213] Manuel T. Schrempf. [Question][Improvement] ARCore Node implementation. <https://github.com/google-ar/sceneform-android-sdk/issues/334>. Accessed Online: 11.11.2019.