



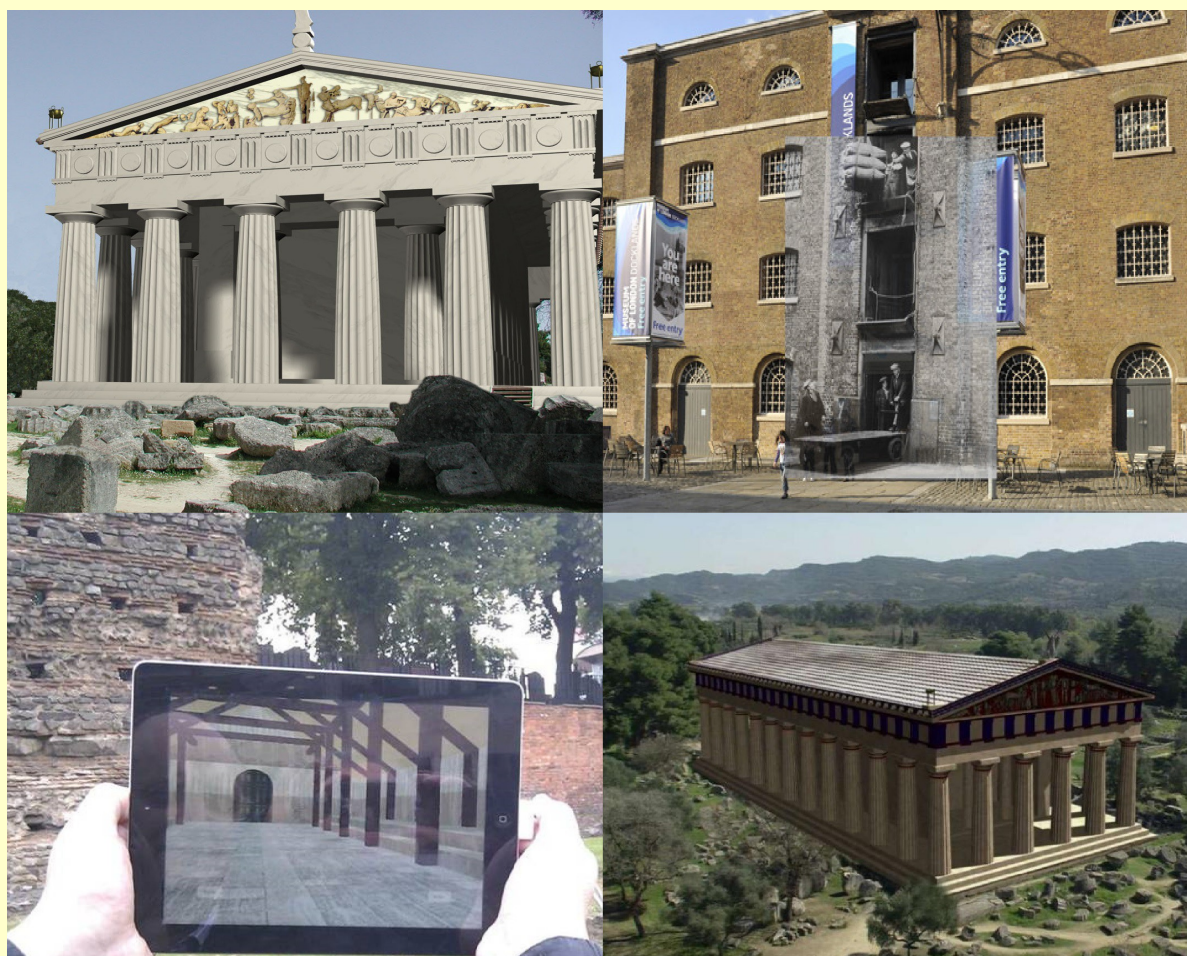
NATIONAL TECHNICAL UNIVERSITY
OF ATHENS
&
NATIONAL INSTITUTE OF APPLIED SCIENCES
OF STRASBOURG



A SURVEY OF AUGMENTED REALITY TECHNOLOGY IN THE FIELD OF CULTURAL HERITAGE

DIPLOMA THESIS

KYRIAZI PELAGIA



ATHENS, MARCH 2018



NATIONAL TECHNICAL UNIVERSITY
OF ATHENS
&
NATIONAL INSTITUTE OF APPLIED SCIENCES
OF STRASBOURG



A survey of Augmented Reality technology in the field of cultural heritage

Diploma Thesis
of
Kyriazi Pelagia

Supervisors:

Georgopoulos Andreas, Professor NTUA
Grussenmeyer Pierre, Professor INSA Strasbourg

Tripartite thesis committee:

Georgopoulos Andreas, Professor NTUA
Ioannidis Charalabos, Associate Professor NTUA
Doulamis Nikolaos, Assistant Professor NTUA

Athens, March 2018

Acknowledges

Having completed the present diploma thesis, I would like to thank all the people that contributed to its development in any possible way.

First and foremost, I would like to express my sincere gratitude to my research supervisor, Prof. Andreas Georgopoulos, director of the Laboratory of Photogrammetry within the Department of Topography and the School of Rural and Surveying Engineering of National Technical University of Athens (NTUA), for the valuable guidance that he gave me in working upon my thesis. I am grateful for his support and trust as he consistently allowed this dissertation to be my own work, but in the same time steered me in the right direction whenever I needed it. The door to Prof. Georgopoulos' office was always open, during all the years of my studies, whenever I ran into a trouble spot or had a question. As he has been my Photogrammetry professor too, I would like to mention that his teaching methods and enthusiasm made a strong impression on me. It would not be an exaggeration to say that he actually made me love the topic, in cooperation with my other two professors of this class, prof. Charalabos Ioannidis and prof. George Karras, to whom I am also grateful.

I had the opportunity to also have a second supervisor, Prof. Pierre Grussenmeyer, at the Department of Civil Engineering and Surveying of National Institute of Applied Science (INSA) of Strasbourg. Prof. Grussenmeyer's guidance in my first days in a foreign University was very important. He gave me access to the laboratory and research facilities, which allowed me to work on my diploma thesis during my six-month stay in France.

My sincere thanks also go to Athanasios Voulodimos, Research Associate in National Technical University of Athens, who devoted his time to my dissertation at any moment I asked for his help, although he had no obligation of doing so. His knowledge in computer science gave me a new perspective on programming in Java language. Additionally, I am thankful to prof. Nikolaos Doulamis for offering his help in case I wanted to follow another approach for the development of the application, even though I finally did not have the chance to work with him.

Getting through my dissertation required more than academic support, and I have many friends and family members to thank for listening to and, at times, having to tolerate me over the past years.

Life has been kind enough to offer me the friendship of many special people, whose love I have been getting for many years. I cannot begin to express my gratitude to the friends that I have made during my years of studying in high school and university. I will not adduce all their names, as I am sure that they know who I am talking about. I want them to know that I count them as friends for a lifetime and I will not stop reciprocating their encouragement and appreciation. To this category I include my "international" friends from Brazil, Mexico and Germany. Even though I first met them just a year ago, they were my family while I was far away from my real one. I could not have stood the challenges of living abroad without them to support me.

Most importantly, I strongly believe that none of this could have happened without my family. My parents, my brother and my grandmother were there for me every time I was ready to quit. During all the years of my life I remember them saying to me that I am strong to accomplish everything I want, which motivated me to try harder and to set higher targets. This last year they have seen me more stressed than I have ever been as there were many times I felt that I would disappoint them. Without even having to ask, though, they repeatedly informed me that all they want for me is to be happy. The fact that they kept supporting me no matter the circumstances is priceless and makes them role model family to my eyes. This dissertation stands as a testament to their unconditional love and encouragement.

The study and the rehabilitation project of the Holy Aedicule became possible and were executed under the governance of His Beatitude Patriarch of Jerusalem, Theophilos III. The Common Agreement of the Status Quo Christian Communities provided the statutory framework for the execution of the project; His Paternity the Custos of the Holy Land, Archbishop Pierbattista

Pizzaballa (until May 2016 – now the Apostolic Administrator of the Latin Patriarchate of Jerusalem), Fr. Francesco Patton (from June 2016), and His Beatitude the Armenian Patriarch of Jerusalem, Nourhan Manougian, authorized His Beatitude the Patriarch of Jerusalem, Theophilos III, and NTUA to perform this research and the project. Contributions from all over the world secured the project's funding. Worth noting Ioanna- Maria Ertegun Great Benefactor and Jack Shear Benefactor through WMF, Aegean Airlines as major transportation donor et al.

Acknowledgements are attributed to the interdisciplinary NTUA team for the Protection of Monuments, Professors Em. Korres, A. Georgopoulos, A. Moropoulou, C. Spyrakos, Ch. Mouzakis and specifically, A. Moropoulou, as Chief Scientific Supervisor, of the rehabilitation project.

Additionally, for the 3D models of the Aedicule used here, the contributions of Prof. A. Georgopoulos, Prof. Ch. Ioannidis, S. Soile, S. Tapinaki and R. Chliverou members of the Geometric Documentation team are gratefully acknowledged.

Abstract

Augmented Reality (AR) is a technology which allows the enhancement of the physical, real world environment, through its augmentation with computer generated sensory input. The idea of an electronic display that superimposes data over a real-world scene in order to provide more information about it, already exists from 1901, but it was only recently that the first relevant applications came to life. Nowadays, it has practical applications in various fields, not only scientific but also fields of everyday life. Many researchers and companies are trying to develop innovative AR systems.

One of the fields using Augmented Reality is the promoting of cultural heritage. By creating three dimensional models of monuments it is possible to display them over an image far away from their actual location, giving the opportunity to more people to get to know them and even interact with them. Their visualization through the years is also a potential, which can offer an interactive experience to the visitors. Especially in case the monument is nowadays partly destroyed, its presentation during the days of its glory can be very impressive and informative. Moreover, the construction of the digital model is a part of the geometrical documentation of the monument, which contributes to its completed documentation and to its maintenance and transmission to future generations. This transmission is one of the domain objectives of the architectures and the other scientists who are in charge of protecting the cultural property.

The present thesis is the result of the research done, in order to better understand the concept of Augmented Reality and how it can contribute to the field of cultural heritage. The vision-based methods of tracking the surface are mostly examined, as they are the most widespread techniques, as well as the most relevant to the field of photogrammetry. The thesis is separated in four chapters, each one having a specific role to play.

The first chapter is about the theoretical background. It begins with the definition of the Augmented Reality term, to help the readers comprehend its basic characteristics and distinguish it from other versions of reality. The historical review of how the whole idea came to life and developed follows, to provide an understanding of its evolution through the years and to present all the notable researchers who contributed to make this system available. The next important element is the different aspects of life where it has made inroads. The impact that has been left until now is crucial in order to estimate the future development and the even bigger modifications that are going to be made.

The second chapter aims at the explication of the system's technology. Three categories of hardware components are essential to create an AR system; sensors, processors and display devices. Their functionality is explained in this part. A brief presentation of how the whole system works follows, to explain how these components are used to produce the final result. Emphasis is being given to the tracking, as it is the most important procedure for the real-world scene augmentation. The problems, or better challenges, that the researchers have to face are also presented here. Last but not least, some of the projects in which monuments or archaeological sites have been enhanced by Augmented Reality systems are described.

The attempt to produce an application which superimposes the 3D model of the Tomb of Christ, situated in Jerusalem, on the image of its floor plan, is reported in the third chapter. The chapter begins with the history and the architecture of the Church of the Holy Sepulchre, where the monument is located and it continues with how the monument's digital construction has been done. Next, the tools and the components that have been used for the production of the application are presented. Finally, the system's building is described, giving

emphasis to the code's most important parts.

The last chapter is basically an epilogue. It includes conclusions concerning the thesis and an evaluation of the application which is based on the results. For the closure, expansions that could possibly be created in the future are discussed.

Key Words: Augmented Reality, Cultural Heritage, Android, Java

Table of Contents

Abstract.....	1
1. Augmented Reality.....	7
1.1. Definition.....	7
1.2. Types of Reality.....	8
1.3. Historical Research.....	9
1.4. AR for Maintenance of Cultural Heritage.....	15
1.5. Other uses and applications of AR.....	17
1.5.1. Military.....	17
1.5.2. Navigation and tourism.....	17
1.5.3. Medical.....	18
1.5.4. Education.....	18
1.5.5. Visual Art.....	19
1.5.6. Architecture.....	19
1.5.7. Industrial design.....	20
1.5.8. Commerce.....	20
1.5.9. Video games.....	21
1.5.10. Broadcast and Live Events.....	21
1.5.11. Others.....	22
2. Technology.....	23
2.1. Hardware Technology.....	23
2.1.1. Sensors.....	23
2.1.2. Processors.....	24
2.1.3. Displays.....	25
2.1.3.1. Head-Attached Displays.....	26
2.1.3.2. Hand-Held Displays.....	28
2.1.3.3. Spatial Displays.....	29
2.2. Software Technology.....	30
2.2.1. Tracking Methods.....	31
2.2.1.1. Visible Light Tracking.....	32
2.2.1.1.1. Fiducial Tracking.....	32
2.2.1.1.2. Natural Feature Tracking.....	33
2.2.1.1.3. Model Based Tracking.....	34
2.2.1.2. Infrared Light Tracking.....	34
2.2.1.3. 3D Structure Tracking.....	34
2.2.1.4. Location-Based tracking.....	35
2.2.1.5. Fusion strategies.....	36
2.2.2. Pose Calculation in visual tracking methods.....	36
2.3. Today's challenges.....	39
2.4. State of the Art – Cultural Heritage AR Applications.....	40
2.4.1. Archeoguide.....	41
2.4.2. LIFEPLUS: Revival of life in ancient Pompeii.....	42
2.4.3. Calakmul – Ancient Maya Metropolis.....	44
2.4.4. The PRISMA project.....	45

2.4.5. A Future for the Past.....	47
2.4.6. The House of Olbrich.....	48
2.4.7. The CHESS project.....	48
2.4.8. LecceAR.....	50
2.4.9. The MIRACLE project.....	51
2.4.9.1. Luostarinmaki Adventure.....	51
2.4.9.2. Wordsmith.....	52
2.4.9.3. House Casagrande.....	52
3. Producing an AR application.....	53
3.1. The Church of the Holy Sepulchre.....	53
3.1.1. History.....	53
3.1.2. Description and Architecture	54
3.2. Preparation of the 3D Model.....	56
3.2.1. Reconstructed 3D Model of The Tomb of Christ.....	57
3.2.2. Reducing 3D Model's Size.....	58
3.3. Development tools and equipment.....	60
3.3.1. Programming Language.....	60
3.3.2. Framework.....	61
3.3.2.1. Selection's procedure.....	61
3.3.2.2. ARToolKit – Information and features.....	62
3.3.2.3. ARToolKit – Requirements.....	63
3.3.3. System's technical specifications.....	64
3.4. Using native code.....	66
3.5. Project creation.....	67
3.5.1. Building the application.....	67
3.5.2. Running the application for the first time.....	76
3.5.2.1. Camera's Calibration.....	76
3.5.2.2. First results.....	76
3.5.3. OBJ loader's creation.....	77
4. Results and Commentation.....	80
4.1. Conclusions.....	80
4.2. Application's results.....	80
4.3. Future work.....	83
Reference List.....	86

1. Augmented Reality

1.1. Definition

Augmented reality is a rapidly and constantly evolving technology which enables to overlay computer-generated sensory input on a view of a physical, real-world scene, being view through a device, in order to create an enhanced version of reality. Nowadays, the input elements that complement the reality can be sound, video, graphics and GPS data. It is strongly believed, though, that in some years it will also be possible to use all of our senses, including smelling and tasting. These objects have to be presented as part of the image, giving the impression of coexisting in the same place and time with it [64].

There are plenty of definitions for what augmented reality is. The most common accepted is the one of Ronald Azuma, who defined augmented reality as a variation of Virtual Environments (VE), with the difference that while *VE technologies completely immerse a user inside a synthetic environment, AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world*. So, as he ends up, *AR supplements reality, rather than completely replacing it* [3].

In order to better explain this definition, an explanation of virtual environments (VE), or virtual reality (VR), is also necessary. In virtual reality projects, realistic images, sounds and other sensations are generated, using a software. These sensations replicate a three-dimensional environment, which may either copy real-world attributes, or be a completely imaginary world. The user can interact with everything existing in this environment, using specialized display screens or projectors and other devices [126].

Now that both these types of technology have been defined, it's easier to understand the crucial difference of Azuma's definition. VR actually brings the user into the digital world by cutting off outside stimuli, while the real-world scene has the dominant role at AR projects. In fact, as he also claims, *AR can be thought of as the "middle ground" between VE (completely synthetic) and telepresence (completely real)* [3].

In order to avoid misunderstandings about whether a system can be considered as AR system or not, Azuma also defined the characteristics of every AR system. According to him, AR systems [3]:

- Combine real and virtual
- Are interactive in real time
- Are registered in 3-D

Using these characteristic, it is getting easy to understand that a film, for example, cannot be considered as AR system, even if it overlays photorealistic virtual objects on 3-D real-world scenes, as long as it is not interactive in real time. Another example is the 2-D overlays used on top of live video, an interactive in real-time technology which adjust virtual elements on real-world scenes, but not in 3-D.

The possibility to distinguish between the Augmented Reality and other types of Reality systems, is the reason why Azuma's definition is considered really important for every AR research. There are, though, other definitions, which set AR as a sector of other, more general, types of reality. For the sake of completeness, these types of reality have to be presented as well.

1.2. Types of Reality

Mixed Reality (MR) or, according to others, Hybrid Reality aims to combine all possible variations of real and virtual objects, so as to produce new environments and visualizations. In order to connect completely real environments to completely virtual ones, Paul Milgram and Fumio Kishino used the term “virtuality continuum”, which include Augmented Reality, Augmented Virtuality and other mixed configurations. They defined Mixed Reality as anything *between the extrema of the virtuality continuum* [31].

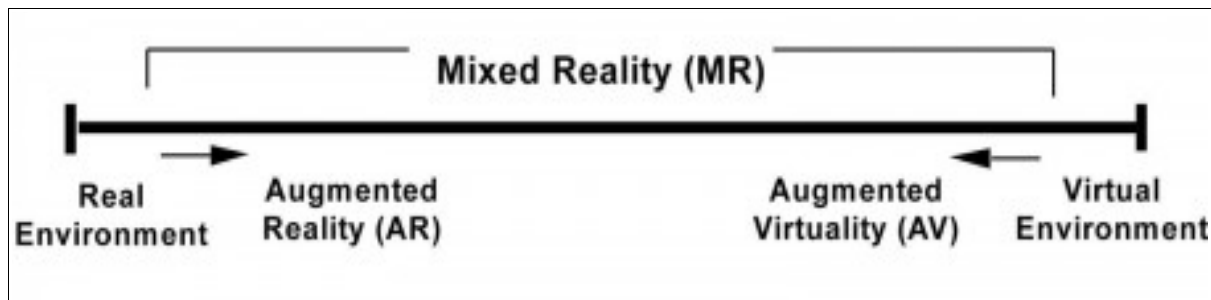


Figure 1: Reality - Virtuality Continuum [31]

According to their research, in a representation like the one of figure 1, placing real environments at one end of the continuum and virtual environments, meaning Virtual Reality projects, at the opposite extremum, it's possible to have at least 6 classes of intermediate category. These classes depend on the technology used to accomplish the different results. For instance, with monitor-based video displays it's possible to accomplish different results from the ones accomplished with head-mounted displays (HMD's) or with completely graphic display environments.

They defined AR as *any case in which an otherwise real environment is “augmented” by means of virtual (computer graphic) objects*, a definition with which 4 of the 6 classes are included in the AR term. The fifth class, which refers to *completely graphic display environments with video “reality” added*, is the one for whom the term Augmented Virtuality is used. Since the world being augmented is a virtual, and not a real one, the name AR would not be the appropriate one. Finally, sixth class, which introduce a virtual environment where someone can use real-world objects (parts of his body, for example), is the reason why the term Hybrid Reality has been imported, since a blending of many types of display media is used in this case.

An even more general framework, that also includes the reality - virtuality continuum, is the Mediated Reality. In the concept with the name Mediated Reality, the view of the real-world scene can be not only augmented, but generally modified, by a computer. This way the human perception can be modified by way of devices for augmenting, deliberately diminishing and otherwise altering sensory input [29].

The previous sentence includes one more reality-relevant term, the Diminished Reality. A Diminished Environment is the one where parts of the existing real-world scene have been diminished. In order to accomplish the diminishment, an object or a collection of objects has to be replaced with an appropriate background image [26]. That's the reason why, in some researches, Diminished Reality is thought to be the same as AR, since there is a computer-generated image, being input in both these cases.

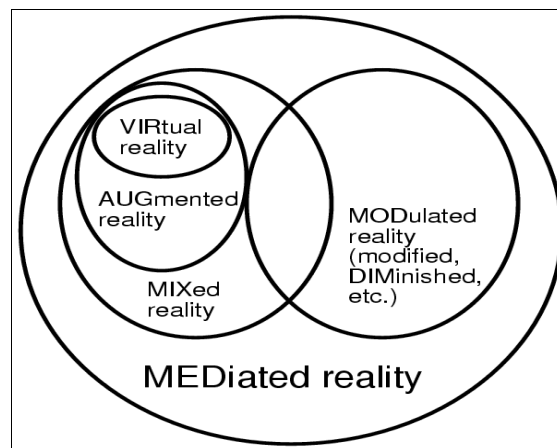


Figure 2: Mediated Reality [29]

One more concept complementary to AR is that of Amplified Reality. As it is defined in the relevant search of Falk, Redström and Björk [14], *to amplify reality is to enhance the publicly available properties of a physical object, by means of using embedded computational resources*. The difference between Augmented and Amplified Reality is that an amplified object is self-contained in regards to its properties. AR superimposes virtual properties on an object, enhancing impressions of real objects, but not actually changing it. It just changes the way a person experiences it, privately. On the other hand, Amplified Reality enhances the expressions of objects in the real world, emphasizing the importance of the shared experiences that result from publicly available properties of objects.

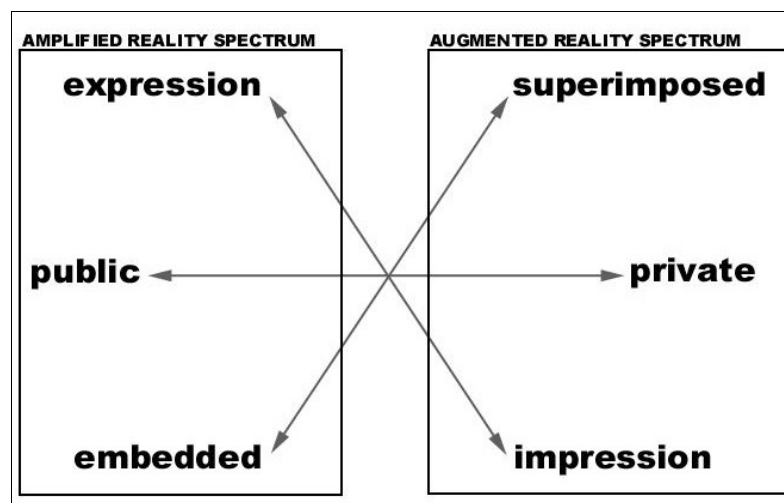


Figure 3: Main differences between Augmented and Amplified Reality [14]

1.3. Historical Research

Augmented Reality has been known to the masses only the last years. Nevertheless, the first mentions of the idea of an electronic device that overlays data on a real-world scene to enhance reality takes already place in 1901, by an author, L. Frank Baum [78]. In his novel, *The Master Key*, he imagines a pair of spectacles which overlays a letter on the forehead of people, indicating their character. Only whoever wears the device can see the letter, which is exactly the same logic as the one of the AR devices.

In 1955 a cinematographer, Morton Heilig, detailed his vision of a multi-sensory theater, simulated with “reality machines”, in a paper titled the “Cinema of the Future” [18]. Five year later, in 1962, he built a prototype of his vision, named Sensorama, a mechanical device which still functions today. Sensorama is thought to be a predator of digital computing. It had the capability to blow wind, vibrate and surround sound.

In 1968 the computer scientist Ivan Sutherland with his student Bob Sproull created a head-mounted display (HMD) system, named “the Sword of Damocles”. Using suitable two-dimensional images on the observer's retinas, they managed to create the illusion of three-dimensional display. Half-silvered mirrors were used as optical combiners, allowing the user to see both the computer-generated images reflected from cathode ray tubes (CRTs) and the real-world scene. The perspective image presented was changing as the user was moving his head, so all that had to be measured in order to present the appropriate display, was the position and the orientation of the optical system to the user's head. He managed to do this with mechanical and ultrasonic head position sensors. The user was surrounded with displayed 3D information of objects, appeared to hang in the space all around him. The existing technology permitted the objects displayed to only be transparent “wire frame” line drawings and not 3D objects, as in today's AR systems. Nevertheless, the “Sword of Damocles” is considered to be the first VR and AR HMD system. [48]

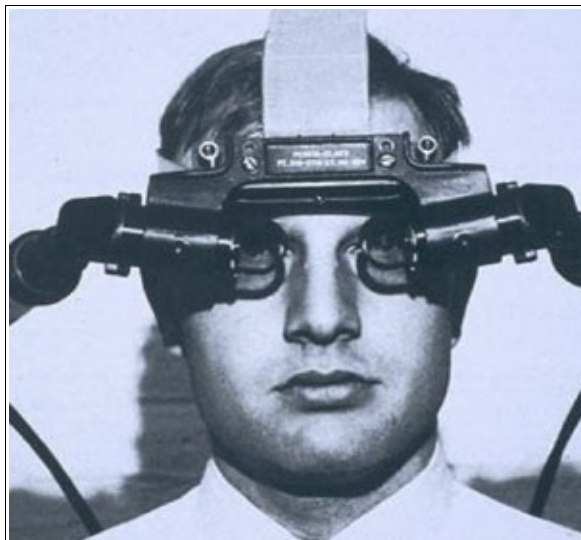


Figure 4: The Sword of Damocles [47]



Figure 5: Today's HMD system [99]

In 1975 established “Videoplace”, an artificial reality laboratory in which users could interact with virtual objects for the first time. Projectors, video cameras, special hardware and onscreen silhouettes have been used for this purpose. The users, situated in separate rooms, had their movements recorded on video and transferred to the silhouette representations in the Artificial Reality environment. All the users where seeing the same representation, including all of them. This way it was possible to interact with each other, no matter how far away the other rooms were [125].

In 1978, Steve Mann created the first wearable computer, a device which is using cameras, processors and display systems to help people see better, named “Digital Eye Glass”, as it causes the eye to appear as a glass prosthetic. To accomplish it, this machine

caused the human eye itself to become both an electronic camera and a television display, using “augmediated reality” vision systems [58].

The 1980s was the decade when AR technology started being used to enhance already existed sectors of life. In 1981, Dan Reitan was the first one to bring Augmented Reality to TV, as he uses it to enhance the television weather broadcasts. In 1987, Douglas George and Robert Morris created an AR system to superimpose information relevant to stars and celestials over the sky images, on an astronomical telescope-based AR system [64].

As the new technology had already started invading our life, in 1990 the term “Augmented Reality” was used for the first time, by Thomas Caudell, a former Boeing researcher, to refer to overlaying computer-presented material on top of the real world. Caudall was working on a system that would assist the workers in the assembly and installation of electrical cables in the aircrafts. The relevant research published in 1992 [12]. The term “Virtual Reality” had already been used in 1989 by Jaron Lanier.

In 1992 Louis Rosenberg developed the first fully immersive AR system, named virtual fixture. It is an overlay of augmented sensory information on a workspace, designed to assist in the user while performing real physical tasks. In order to achieve photorealistic result, two robots controlled by a full upper-body exoskeleton were used. The user, wearing the exoskeleton and a pair of special binoculars magnifiers, was seeing robot arms where his arms should be. The system's aim was to demonstrate benefits to human performance [40].

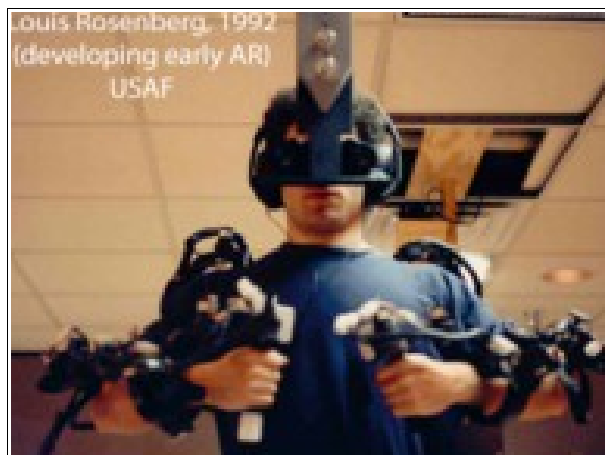


Figure 6: Virtual Fixtures [40]

The same year an AR system prototype, KARMA (Knowledge-based Augmented Reality Maintenance Assistance), was presented at the Graphics Interface conference by Steven Feiner, Blair MacIntyre and Doree Seligmann. KARMA is a guide of maintenance and repair for a laser printer, using see-through head-mounted display (HMD) [96].

In 1993, Loral Electronics Corporation, a defense contractor, presented a series of live demonstrations combining live AR-equipped vehicles and manned simulators. Vehicle crews were able to see virtual vehicles and weapon effects and interact with them in real-time on the same battlefield [5].

In 1994, in an Australian TV show called “Dancing in Cyberspace”, created by Julie

Martin, dancers and acrobats manipulated body-sized virtual objects projected onto the stage with them. The acrobats appeared immersed within the virtual environments [63].

In 1995, Benjamin Bederson introduced the term Audio Augmented Reality, by presenting a system that demonstrated an augmentation of the audition modality. This system, which uses a MD-player with audio-information relative to the tracked position of the user, is still in use as a museum guide [6].

The same year, Jun Rekimoto and Katashi Nagao created the NaviCam, the first palmtop computer equipped with a video camera mounted on the mobile screen, used for optical tracking. The computer was programmed to detect color-coded markers in the live camera image [42].

Inspired by these color-coded markers, Jun Rekimoto created a 2D square-shaped barcode system of markers, CyberCode, some years later. It was one of the first marker systems to allow camera tracking with six degrees of freedom. Nowadays, a lot of AR applications still use CyberCode [41].

In 1997, the first mobile augmented reality system (MARS) was presented by Steven Feiner et al. It used a wearable computer, a see-through head worn display with integral orientation tracker, GPS and digital radio for wireless web access [15].

In parallel, a similar system was developed by a group of researchers in Massachusetts Institute of Technology. A group of users, equipped with wearable computers interconnected over a network, explored possible applications of mobile AR [45].

In 1998, in a paper named “The office of the future”, ideas for spatially immersive display were presented. These ideas have been proved useful for a lot of computer vision type applications, including Augmented Reality [38].

The same year, an investigation into expanding augmented reality systems to outdoor environments was presented at the 2nd International Symposium on Wearable Computers. The researchers developed a system called “map-in-the-hat”, which is a backpack-based wearable computer with a see-through head-mounted display, an electronic compass and a GPS, used for navigation tasks [50]. With the technology development this system evolved to Tinmith project, which is known for many mobile outdoor Augmented Reality applications [122].

In 1999, NASA Johnson Space Center, was developing real-time 3D displays for flight guidance, overlaying three-dimensional terrain models, recommended landing areas and maps. They used real-time GPS position data and head-up display (HUD) to accomplish the successful result [109].

At the same time, the US Naval Research Laboratory developed wearable systems for dismounted soldier operating in urban environment, on a research called Battlefield Augmented Reality System (BARS). The system targets the augmentation of a battlefield scene with additional information about environmental infrastructure, but also about possible enemy ambushes [123].



Figure 7: BARS backpack [123]

1999 was also the year when ARToolKit, the open-source computer tracking library under the GPL license, currently hosted on GitHub, was released by the University of Washington HIT Lab (Human Interface Technology Laboratory). ARToolKit was originally developed by Hirokazu Kato and Mark Billinghurst and further developed by other HitLab scientists. [59]

Entering the new millennium, the technology of Augmented Reality was becoming more and more widespread and the number of researches released also increased. Starting from 2000, only the most important historical references for augmented technology development are presented.

In 2000, the first outdoor mobile AR game, ARQuake, was demonstrated by his creator, Bruce H. Thomas, in the International Symposium on Wearable Computers. It is an Augmented Reality version of the popular Quake game, using the wearable computer system Tinmith to track the position and the orientation of the user's body and head [49].

In 2002, the first International Symposium on Mixed Reality and Augmented Reality (ISMAR) was held in Germany. The predecessors of ISMAR, International Workshop on Augmented Reality (IWAR), International Symposium on Mixed Reality (ISMR) and International Symposium on Augmented Reality (ISAR), had taken place for the first time in 1998, 1999 and 2000 respectively. From 2002 and after, ISMAR is organized annually by IEEE Computer Society and IEEE VGTC at different countries, permitting the researchers to participate and exchange opinions and thoughts relevant to these technologies [80].

The following years, cell-phones games using AR started to be released. The first Augmented Reality game for cell-phones, named Mozzies, was released for Siemens SX1 in 2003, winning, also, an award of the best mobile game in 2003 [118].

Just a year later, one more game, named "The invisible train", is presented. The innovation of this game is that it is the first multi-user AR application for handheld devices [120].

Another important technology step was taken in 2004, when Mathias Möhring, Christian Lessig and Oliver Bimber presented a system which supported optical detection and differentiation of different 3D markers and correct integration of rendered 3D graphics

into the live video stream, on a consumer cell-phone. The technological constraints were creating limits to the quality of the results, but a first running system was realized, giving the opportunity of further development [32].

In 2006, Reitmayr and Drummond presented a model-based hybrid tracking system for outdoor Augmented Reality in urban environments, enabling accurate, real-time overlays on a handheld device. The system combines an edge-based tracker for accurate localization, gyroscope measurements to deal with fast motions and measurements of gravity and magnetic field to avoid drift [39].

The location-based programs used as travel guides were the next to be developed with AR systems. In 2008 the company Mobilizy GmbH created Wikitude, a mobile AR technology provider, which initially focused on providing location-based experiences. The Wikitude World Browser was using GPS, accelerometer and compass, in order to overlay Wikipedia entries on the real-world scene viewed from the mobile's camera [127].

A year later the company SPRXmobile also created a location-based program, named Layar, a more advanced variant of Wikitude. Using the same registration mechanism as Wikitude, it created an open client-server platform, overlaying entries not only from Wikipedia but also from Twitter and Brightkite, as different layers [98].

It is important to mention that the first AR Travel Guide had already been created by Columbia's Computer Graphics and User Interfaces Lab [7]. The first AR browser had also been created by Kooper and MacIntyre [25]. Both these systems were already in use from 2001.

In 2010, Microsoft announced the cooperation with Primesense, an Israeli company working on structured-light based 3D sensors. The result of this cooperation was Kinect, a line of motion sensing input devices, which enables the users to interact with their computer through a natural user interface using gestures and spoken commands [97].

2012 was the year when Google Glass was presented to the public. It is an optical HMD that can be controlled with an integrated touch-sensitive sensor or natural language commands. The glasses reach the Internet through Bluetooth, which connects to the wireless service on a user's cellphone [83].

The same year, NVidia also demonstrated a prototype of a head mounted display, supporting accurate accommodation, convergence and binocular-disparity depth cues. Thanks to its accurate accommodation, it can be seen as a next generation wearable display technology for AR [77].

In 2015, Microsoft announced Windows Holographic and the HoloLens AR headset. This device is a complete computer with a see-through display and several sensors. It utilizes various sensors and a processing unit to blend high definition holograms with the real world [79][101].

In 2016, Niantic released Pokémon Go for iOS and Android, the location-based game that brought Augmented Reality to the mainstream. The players use their mobile's cameras and gyroscopes in order to see and capture virtual creatures. It was credited with popularizing location-based and augmented-reality technology, promoting physical activity and helping local businesses to grow via foot traffic [107].

1.4. AR for Maintenance of Cultural Heritage

In order for someone to understand the meaning of cultural heritage, it would be useful to examine its words both separately and as a whole.

The word heritage includes the idea of preserving, excavating, displaying and restoring a collection of old things, but it's also much more than that. It describes the inherited traditions, monuments, history, a state's background. As a result, it strengthens the sense of identity and belonging to a community and, in the same time, it reinforces social cohesion and the sense of common responsibility. As for the word culture, it has been defined as *that complex whole which includes knowledge, belief, art, morals, law, custom and any other capabilities and habits acquired by man as a member of society* [124]. Culture is created by both forces encouraging change and the ones resisting to it. That's exactly the reason why it must be constantly preserved, but also adjustable to the civilization's development.

The unification of all these meanings in the term “cultural heritage” adduces the significance of the legacy of all the material and intangible attributes which are received from past generations, with the obligation to be maintained in the present, so as to be transmitted to the future. According to UNESCO, *The cultural heritage may be defined as the entire corpus of material signs – either artistic or symbolic – handed on by the past to each culture and, therefore, to the whole of humankind* [21]. This sentence sums up the concept of cultural heritage, also explaining why its preservation and presentation are a *corner-stone of any cultural policy* [21]. Each structural member of each monument transmits the technology of its time, as well as the spirit of the constructor.

During the effort to explain the significance of the cultural heritage, two meanings kept on appearing in different words: Preservation and presentation.

The first is the one directly connected to the term in everyone's mind nowadays. During the past years, unfortunately, the awareness for the maintenance of cultural heritage was not as widespread as today. As a result, a lot of monuments have been destroyed by natural or human factors. It is possible, thankfully, to construct digital models of their initial situation, completing the 3D model that can be constructed from the ruins with digital data, which can be created based on the corresponding registered description. As for the ones standing in one piece, the construction of their 3D model can be proved very practical in the “preservation” part, as future reference. This way, in case of any damage caused, it would be easier to repair them and to recreate their initial form. As it is said in the Venice Charter, which is a set of guidelines drawn up in 1964, that provides an international framework for the conservation and restoration of monuments and sites, this aim *must have recourse to all the sciences and techniques which can contribute to the study and safeguarding of the architectural heritage* [124]. The 3D model construction mentioned above is a way of contributing in the geometrical documentation, which is an important subset of a monument's complete documentation.

The second meaning, presentation, is exactly the sector in which the usage of the 3D models can make a huge difference compared to the traditional methods. Watching a 2D representation of any item is totally different than being able to interact with it, as it only offers the ability to see the part presented in a particular view, with all the occlusions that result from this fact. It is easier to understand the problem thinking of an item with a lot of different parts in it, for instance, a house. An architectural plan for a house usually contains a floor plan or a horizontal section cut and one or more face plans. However, it is impossible for every item in the house to be presented of every aspect. While that's not a problem when it comes to a house display, it is certainly important when the item presented is a monument,

where, as said above, each structural member transmits special features of its time. Using a 3D model instead, there is no reason to worry about which parts have to be presented, as it is possible to include them all. The user just has to turn the model to find the specific aspect he is interested in, a procedure he can repeat as many times as he wants.

Having understood how the 3D model's usage can contribute to both the preservation and the presentation of the cultural heritage, it's time to explain the difference Augmented Reality can make.

The technology of Augmented Reality gives to everyone the opportunity to admire a digital reconstruction, situated on its proper place, in real time, using a device's camera. This possibility allows the audience an interaction with the monument which could not be achieved otherwise. The feeling of admiring an architectural site in its natural space, where it was either constructed or intended to be, is completely different than seeing it in a foreign environment, which has no connection to its history. This is an argument used by historians and archeologists, who allege that the Parthenon Marbles, also called Elgin Marbles, now situated in the British Museum, should return back to Athens. As they say, presenting all the Marbles in their original historical and cultural environment would permit their better appreciation as a whole. The same argument could be used to justify every monument's existence to its initial place for the audience to visit it, because of its fuller understanding and interpretation that is provided this way.

A second and equally important result that emerges from using Augmented Reality in the Cultural Heritage field is that it captures the interest of a large amount of people. Technological progress has brought a lot of changes in everyday life, including the fields of amusement and education, which are both relevant to cultural issues. Nowadays, people tend to prefer learning about different cultures through a computer, not only because of the convenience, but also because of the access to a lot of information provided. There is the possibility, for example, to focus on a specific part of a monument, in which they are interested and learn information about it, rather than being obliged to read or listen to general facts. Augmented Reality, however, offers this possibility while standing in front of the structure, by including not only texts to read but also images or videos, explaining the special characteristics of each item. The user has the freedom to decide what he wants to listen about and what he would rather not, or even which way he prefers to do it, just like he could do using his computer.

Remaining in the field of attraction coming from technological progress, it is logical to assume that kids' interest in exploring archeological sites will be significantly raised thanks to Augmented Reality, because of two reasons. The obvious one is that the new generations tend to prefer more and more whatever includes technology in all its forms. It would be fascinating for children to have a device through which they can explore a whole new world of items co-existing with them inside their environment, while not being there in real life, something they may have combined with fiction and fantasy movies. The second reason comes with modifications which can be made in some devices aimed especially at kids. With proper reformation, the experience of visiting an archeological site could be turned into a fairytale or even a game. As a result, learning would be fun and would attract more and more children, creating a new generation of people interested in the cultural field.

Summing up, Augmented Reality can be used as a trump card in the field of Cultural Heritage. Allowing the audience's direct interaction with the item of admiration and the possibility to display the specific information they are interested in, changes the way of the learning experience. This way, a purpose under content research, attracting more people to museums and archeological sites, can be easily achieved. This new technology, which has come to create major changes, has just started revealing its possibilities in the specific field.

1.5. Other uses and applications of AR

The capability of augmenting a real-world scene with constructed data has generated the interest of people dealing with a lot of different sectors, scientific and not. That's the reason why nowadays, Augmented Reality applications cover a broad range of areas.

1.5.1. Military

The military funding of science has had a powerful transformative effect on the practice and products of scientific research, because advanced science-based technologies have been viewed as essential elements of a successful military. Augmented Reality is one of these elements, as it can be used to provide useful information to the soldiers during the battle, in real time.

A relevant application, where map overlays were applied to video, indicating satellites and orbital debris tracks in geographic coordinates, had already occurred by Rockwell International in 1993 [1]. This way potentially dangerous objects could be tracked.

Other elements provided could be geographical information including street names and railroads, airports and other points of interest, or even people. The US Army has armed the Shadow Unmanned Aerial System with the SmartCam3D AR system, which is able to locate data like these, providing the operator with improved situation awareness. From the soldier's viewpoint, people and various objects can also be marked with special indicators to warn of potential dangers.

All this information can be transmitted to the soldiers with the usage of special helmets, or adjusting HMDs to their helmets. Head-up displays (HUD) and helmet-mounted displays (HMD) are in use not only for fighter aircraft, but even in commercial aircraft. A HUD is a transparent display which enable the pilot to view information with the head positioned "up" and looking forward, without requiring him to look away from the usual viewpoint at lower instruments.

For all the reasons mentioned above, there are many research projects, especially at USA, trying to develop new Augmented Reality Systems for military. These projects are funded by services like Defense Advanced Research Projects Agency (DARPA), Army Research Laboratory (ARL) and Office of Naval Research (ONR).

1.5.2. Navigation and tourism

The effectiveness of navigation devices can be drastically improved with the usage of AR technology.

In the driving case, the information augmented can be not only destination directions and meters but also weather, terrain, road and traffic condition or even alerts to potential hazards in the path. The digital map including all these data could of course be viewed by a mobile device, as it happens nowadays, but it could also be displayed on the automobile's windshield, in future applications. This way it's easier for the driver to see the information without looking away from the street, making the driving safer than today.

In the case of a flight, it has already been mentioned that even in commercial aircrafts the pilots wear a HUD, with which it is possible to have all the information they need without looking away. The NASA X-38 was using a Hybrid Synthetic Vision System that overlaid map data on video during flight tests from 1998 to 2002. In cases of limited visibility, it was possible to rely on the maps overlays, as all the data needed were indicated on the video.

One type of navigation that can extremely change with the usage of AR technology is

the navigation for the pedestrians. The best example is the location-based programs which are being used as travel guides. Such programs have started being developed from 2001 and are mostly used by tourists. There are a lot of applications linked to geographical locations which provide information like entries from Wikipedia or other internet pages, regarding a location and its features, when it becomes visible to the user. For some of them it is also possible to see comments or other type of content provided by previous visitors. In other, more advanced applications it is also possible to see simulations of historical events, places and objects rendered into the landscape.



Figure 8: Wikitude: AR application for navigation [128]

1.5.3. Medical

During the last years Augmented Reality has been transforming medicine and healthcare sectors significantly, from improving medical training to making pharmacy benefit management more effective. The most important difference, though, concerns the assistance that can be given during the process of diagnosis and treatment, as long as the process of surgery.

In the field of diagnosis and treatment, data collected from different medical examinations, such as scanning, magnetic resonance and echography can be combined in order to create a final medical image. This combination could be superimposed over the patient during the examination, to help the doctor better understand the state of his health.

The same logic can be used in the operating room to improve the precision and the efficiency of surgeons at surgeries. A system like this permits the optimization of information directly to the patient's body during the operation. Using a remote control, the doctor could choose which image he needs to overlay, making the need of incision not necessary, in order to see right through the patient and localize critical structures. In addition, it is possible to give to the doctor specific instructions for the operation step by step, or even mark with different colors the organs that he should and the organs that he shouldn't touch.

1.5.4. Education

The technology of Augmented Reality can be used for educational reasons in almost every sector being mentioned in this chapter. For example, it could be used to help the training of future doctors by giving them the possibility to understand a body's anatomy without being present in an operation or future soldiers creating digital battlefield for them to practice, interacting with digital enemies.

As far as it concerns the elementary school and the high school, the AR technology gives to the students the opportunity to participate interactively in every course. Using special markers printed on the books' pages, they can see text, graphics or even video and audio data relative to the content of each chapter. The supplementary information provided could be used either to give more details or to spark the students' interest for the idea devised [65].

By 2016, AR has already started to become integrated in the classrooms and to transform the active learning experience.

1.5.5. Visual Art

AR is used in museums and art galleries in a lot of different ways, optical and acoustical.

A very common technique used in museums the last years is to provide an audio-device to the visitors as a museum guide. The audio-device has the capability of tracking the position of the user, in order to give him additional information for the exhibit located closer to him. Apart from the audio-data, it is also possible for the visitor to explore more optical-data, containing, for example, optical details of the work of art he is viewing, by scanning markers placed next to it.

Another interesting and innovative idea in the sector of art is the creation of an application which allows to the user to create a work of art in his mobile phone, viewing a specific marker through his camera. An example of such an application is "Konstruct", which allows the user to create a virtual sculpture by speaking, whistling or blowing into the device's microphone [82].

Except of museums and art galleries, the world of theater and concerts have also let AR technology to introduce in it. More than 20 years ago, in 1994, the TV show "Dancing in Cyberspace" had already used AR on stage [63]. Nowadays, even small theatrical productions use AR to create special effects, as it is cheaper than using theatrical scenery.

1.5.6. Architecture

A field that could be considered to be associated to art, is the one of architecture. AR is used in this sector too, aiding in visualizing building projects. Just as in the field of cultural heritage, same in this one, it is possible to construct a 3D model of a building. Afterwards, the model can be projected to the place where it is going to be built, offering the possibility not only to imagine, but to actually see the - after the construction - result, in advance. This possibility is given not only for the external, but also for the internal architectural projects. For someone who wants to decorate a room, it is obviously easier to check the best place for every furniture using AR, than having to move them in every possible place in order to decide.

The method for visualizing projected developments within an existing environment is called Inverse Photogrammetry. The custom photogrammetric procedure can be described as passive photogrammetry, since the purpose is nothing more than a digital construction of an object's geometry based on perspective views recorded on photographs. Inverse or active photogrammetry, as the term implies, works in reverse order; the perspective view of a digital object has to be recreated to its physical form. To anticipate a project's result, a planned construction, such as a new building, can be superimposed on existing photographs indicating its current view. In this way it is possible to examine the aesthetic and morphological shape in advance and perhaps modify parts of the design. [17]

1.5.7. Industrial design

One of the fields where AR application can change the process significantly is industrial manufacturing. AR can be used to support some activities in product development and manufacturing, through providing information available to reduce and simplify the user's decisions.

In this domain, the changes can concern the process of assembling guidance, training or simulation, design and planning, the process of maintenance and repair, the process of training, the process of quality control and commissioning and the process of monitoring and visualization [91]. That means that it can be used to enhance almost every stage of the product's life.

The Industrial Augmented Reality (IAR) is still considered to be at an infant development stage, despite the advances in technology. An example of its usage is given by Volkswagen, which used it to verify parts by analyzing their interfering edges and variance. Another project carried out by the same company was the superimposition of calculated and actual crash test imagery, in order to compare the results.

1.5.8. Commerce

In no other fields has the AR excitement exploded in such a huge way than in advertising and marketing. The constant seeking of new ways for the companies to make their products attractive to the customers has implemented a variety of AR applications.

A technique widely used today is using markers over a publication image, which activate a video version of the promotional material. It is an easy and cheap way to make, for example, the publications on magazine's pages interactive, to attract more consumers.

The barcode existed on a product's packaging can be a marker too. The costumers could scan a barcode and receive useful information about the specific product. IBM has demonstrated an app like this, with which the costumer can also choose what information he is interested in [93].

Another innovation is the possibility for the customers to view what's inside a product's packaging without opening it. Lego, which has a long history in using new technology to enhance its products, has already applied this system with great success, as the kids are generally easily impressed by new technology projects.

The internal architectural projects mentioned earlier can also be used for commercial reasons. The Swedish home furnishing company IKEA gives to its costumers the choice to test if new furniture looks good where it is intended to be placed before they buy it. Downloading a free catalog and using the printing version of it in the right spot, the product outline appears in the frame. It can also be rotated and viewed from other aspects.



Figure 9: IKEA Catalog Application [90]

1.5.9. Video games

As the commerce industry, the gaming industry is also looking constantly for new ideas to create games attractive to young people. Augmented Reality combines all the characteristics that excites the youth. As a result, a number of games were developed mostly for indoor but also for outdoor environments.

A very well-known game developed for outdoor environment is *Pokémon Go*, the game created by *Niantic* which made Augmented Reality the most popular game technology of 2016 [107]. It is important to mention, however, that most experts in AR and game development agree that the game is best described as a location-based game, as it is possible to play it without using augmented reality at all. From the other hand, *Clandestine: Anomaly*, also location-based, satisfy all the conditions to be concerned as an outdoor environment AR game. The players have to defend their city from alien enemies, seen through their device camera.

As far as it concerns indoor environment games, from the large number of them existing, *Table Ice Hockey* is one of the most interesting. Using *Playstation Vista* and with the help of markers the player can place an entire hockey ring wherever he wants and view the action from any angle. Another impressive game of this category is *Drakerz-Confrontation*, which brings trading cards to life, as 3D animated monsters pop off the cards to throw down in a battle, using a computer webcam.



Figure 10: Dragonz-Confrontation – Indoor AR game [75]

1.5.10. Broadcast and Live Events

The enhancement of the weather broadcasts was the first application of AR to be presented on television, in 1981. Nowadays animated visualizations projected on virtual geo-spaced maps is something very common in weathercasting. The audience watches the presenter standing in front of a map which is not actually exists and also interacting with it, thanks to augmented reality.

AR has become common in a lot of live events, including sports telecasting. An example is the presentation of a line or an area marked with a rectangle, which is used on football games to indicate transgression of rules from a player. This shape is only visible through the camera, as it is on augmented reality product. A second example is the advertisements that are projected on the center of the court during the breaks of the game,

also only visible through the camera.

1.5.11. Others

The usages mentioned above are the most widespread, but not the only AR applications. Ideas for other fields' usage also exist.

In the field of translation, AR systems can translate foreign text or even spoken words and display them in a user's view as printed subtitles.

In the field of spatial immersion and interaction, human presence can be digitized and a computer-generated model of them can be created, in order to be able to interact and perform various actions in a virtual space.

In the field of emergency management/search and rescue, AR systems can be used in public safety situations, to provide information to people for how they have to react during in an emergency. As it is proved that it is easiest for the people to stay calm if they know that there is something they can do in such situations, an AR guidance can be really helpful.

In workplace, there are a lot of applications capable of facilitate collaboration among distributed team members, like common visualization via touch screen tables, interactive digital whiteboards, shared designed spaces and others. It is also good to use AR to look "through" a machine in order to understand how it works, to see labels of parts of a system and to clarify operating instructions in general.

Last but not least, in the field of robotics, an existing AR system includes a simulator which can retrieve real-time data on the environment of the robot and deliver post-processed information to the robot swarm. This way it is possible to check how the robot would react in situations that is difficult or even impossible to create within a lab, like radioactive emissions in a nuclear disaster case.

2. Technology

As is the case for every contemporary system, hardware and software are both required in order to implement a compelling AR experience. The software is the mechanism telling the system what to do, while the hardware is the equipment that does it. Every system has to ensure their close collaboration so as to achieve an operational result.

In this chapter the different components needed for an Augmented Reality application are described. In addition, it is shown how these elements cooperate to produce the enhancement of the scenery with digital objects. For a better understanding, examples of their usage in innovative projects already completed are also included.

2.1. Hardware Technology

Computer hardware is the physical part of components of a computer. All the relevant to Augmented Reality systems have at least 3 basic hardware components, which are sensors, processors and displays. These elements can have different forms and they can also carry out different roles within different applications. It is crucial, though, these elements to be combined, to produce the result pursued.

2.1.1. Sensors

The main purpose of sensors is to provide information about the location and the orientation of the user to the application. Other information they can provide is geographical coordinates, images, temperature, pH, illumination and other data relevant to the environment [9].

The most common way of tracking, especially for indoor applications, is the optical tracking, which uses a camera. The light gathered through the lens represents an image, which is analyzed to provide the desired tracking information. Apart from the conventional cameras there are also special-purpose ones, operating, for instance, in other frequency ranges, to detect objects not visible in the visible spectrum. Such an example are the thermographic cameras which form an image using infrared radiation. Optical tracking is the most widespread way of tracking, as cameras are already available in many devices that can be used for AR applications, such as smartphones. It is important to ensure that the environment to provide information is suitable for tracking, in order for the system to be able to determine the location and the orientation. An environment can be considered suitable for optical tracking when specific conditions are fulfilled, like stable illumination and the existence of features that distinguish from their background. When these conditions are not met, fiducial symbols, called markers, can be introduced to overcome this shortcoming, something that is not always possible or desirable. Another problem when optical tracking systems are used is that they introduce latency to the system, as it takes time to analyze the image created, and determine the desired information.

A supplementary way of tracking is the acoustic tracking, which uses microphones attached to the object being tracked and others placed in the environment, to track acoustic data. Based on the timing and amplitude of sound sensed by each microphone, the location of the source of the sound is determined, using, in general, ultrasound, a sound with frequency higher than what the human hearing system can perceive. To track multiple objects in the same place, it is also possible to emit different frequencies of sound for each object. The problem of this method is that it requires a different sound source attached to

each object. As a result, it can only be used in environments in which it is known a priori what objects will be tracked. Another problem is that if there are other audio signals in the same frequency range inside the environment, the tracking is not possible.

Electromagnetic tracking, often used in VR systems, is another alternative to optical tracking. It uses orthogonal antennas, which are antennas for transmission and reception, designed for measuring the copolarized signal and the cross-polarized signal. In this system a transmitter with three orthogonal antennas emits a signal, sequentially through each of them. A corresponding receiver, also with three orthogonal antennas, is attached to the entity to be tracked. The signal acquired by each one of the antennas is analyzed to compute the location and orientation of the receiver. The tracking is performed in six degrees of freedom, providing very precise and accurate results. This system can be used at any lighting conditions, as it is not dependent on ambient light levels. It is sensitive, though, to metal in the environment where it is used, and the usable range is limited, meaning that the receivers need to be within a few meters of the nearest transmitter to provide information. An important disadvantage of the system is that it could not be widespread, as it is costly and not easy to use by non-professional users.

Similar to the electromagnetic tracking is the mechanical tracking, which uses linkages attached to the objects instead of antennas. A mechanical linkage is an assembly of bodies connected to manage forces and movement. When used for tracking purposes, they have sensors at each of their joints, reporting the angle between them. With the use of a potentiometer at the joint, the amount of resistance can be measured as a change in voltage, as the angle changes. The voltage can be used to determine the angle between linkages, computing, finally, the location and pose of every object to be tracked. This method is also very precise and, in addition, fast, but except for the advantages, it also has the same disadvantages with the previous one. That's the reason why it is mostly used in medical applications like the AR surgical simulator, where accurate results are of vital importance.

Many tracking systems use a combination of different sensor data, each one of which can be used as supplementary information, in conjunction with other tracking technologies, to create the best result. For example, depth sensor tracking is a way to have information about how far away an object is from the sensor, using different technologies, as optical, acoustical and even radar tracking, to measure this distance. This method only provides information about the depth and, furthermore, the resolution and the accuracy fall off quickly as the distance from the object is being increased. Accelerometers can obtain information about relative motion but not an exact location. Additionally, an error propagation is more possible to occur with systems which use them as their basis. GPS receivers provide location information and compasses provide information about the orientation of a system. When a system needs data relevant to relative orientation, as leaning, turning and twisting, a gyroscope can be used. The combination of some, or even all the above, is widely used for outdoor applications, where the other methods described may fail.

2.1.2. Processors

In general, a processor is an electronic circuit which performs operations on some external data source, usually memory or some other data stream [108]. It could be said that it is the “brain” of any technological system. In an AR system, the processor's role is to integrate inputs from sensors, to execute basic instructions from the application program based on the sensor information and to create the signals that drive the displays of the system [9].

Specifically, for Augmented Reality, the processing system does not consist only of general-purpose micro-processors as the Central Processing Unit (CPU), but also one or

more special-purpose Graphics Processing Units (GPU), which are responsible of executing three-dimensional graphics computations.

The way a processing system works depends on a number of specifications, such as the number of processors used, their speed, the available memory and storage, the graphics accelerators and the network bandwidth and latency. The number of processors can indicate the number of processor chips in the system, the number of cores, the number of desktop computers or other relevant elements. The basic processing required for AR tasks can be handled by any processor nowadays, apart from cases of particularly computationally complex tasks. In this last case there is need of not only more processing capability, but also really high speed and the maximum memory and storage possible. The available memory is the primary concern with handheld devices such as smartphones. Especially if using a GPU, it needs even more memory for graphic computations. The graphic accelerator is the hardware needed to drive the display system. It can support one or multiple display devices. It is also possible to use more than one accelerators, for instance one per display. If the system has to include stereoscopic imagery, special capabilities in the graphic accelerators are essential. The network bandwidth is how many data can be passed in an amount of time, expressed in bits per seconds. There is need of special attendance for the network bandwidth only in cases that many high-complexity computer graphics models have to be passed in a very short amount of time. On the other hand, the network latency is crucial when a real-time response is essential for AR applications, as it expresses the delay between when something should happen and when it actually does. It is the most significant source of problems, especially when a network is involved, as it takes even more time for the system to complete all the procedures.

There are several processor system architectures, depending on in which system the user intends to run the application. For example, there is an architecture for handheld systems such as smartphones, another for handheld systems connected to remote server, another for desktop/laptop computer etc. That depends on the computational power available in each one of these devices. It is also possible to mix different architectures in many ways, to find the best possible combination for a specific application. In order to have the most appropriate architecture it is necessary to make choices between power and portability or portability and network access.

2.1.3. Displays

The display is the device that provides the signals that a user can perceive with his senses. There are many types of displays, that can provide different sensations, such as audio displays and haptic displays. The current research examines only the visual displays.

The primary role of visual displays is to create signals of light to generate images somewhere on the optical path in-between the observer's eye and the real-world scene. A set of components which can be optical, electronic or mechanical are used for this purpose. Depending on the type of the display used, the place where the image is formed and the type of image produced, meaning planar or curved image, differs. Curved images, for example, can be formed by retinal and several projected-based displays, either on the observer's retina or directly on the physical object [10].

There are three basic categories of displays being used to augment the reality: Head-attached, hand-held and spatial displays. The different possibilities of where the image can be formed for each one of these displays, as well as what kind of image is produced in every case, is illustrated at the following figure:

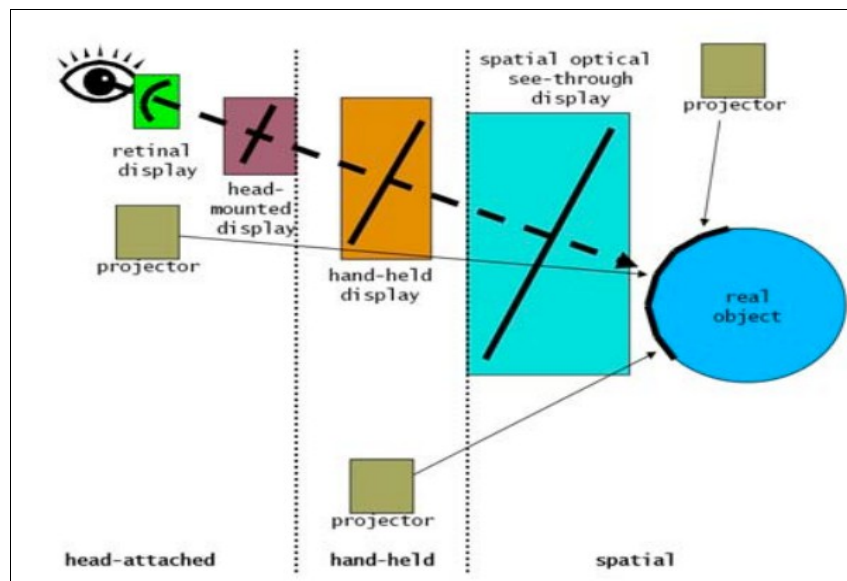


Figure 11: Image generation for different displays [9]

2.1.3.1. Head-Attached Displays

Head-Attached, or else Head-Worn Displays (HWD) are devices that the user wears on his head. Depending on the image generation technology, three main types exist: Retinal Displays, Head-Mounted Displays and Head-Mounted Projectors.

Retinal Displays apply low-power lasers to project images directly onto the retina of the human eye. Thanks to that fact, the field of view is wider than in a screen-based display, while the image is brighter and of higher resolution. It is also important that it consumes less power than its screen-based counterparts that need to illuminate all the pixels present in their screen array.

Head-Mounted Displays (HMD) use miniature displays in front of the eyes. Two types of HMD technology exist nowadays: The Video see-through HMD and the optical see-through HMD [3].

The video approach of HMD combines a closed-view video display and one or two miniature head mounted video cameras, incorporating the real world in the closed-video display by blending images from the video cameras. There are two methods to create the blending. Either by replacing a monochrome virtual background with the real-world video, which is named chroma-keying, or by using depth information at each pixel for the real-world placement of virtual objects. This second way would allow real objects to cover virtual objects and vice-versa.

With an optical see-through display, the real-world is seen through semi-transparent mirrors placed in front of the user's eyes. These mirrors are also used to reflect the computer generated images into the user's eyes, thereby combining the real and the virtual world views. The optical combiners usually reduce the amount of light that the user sees from the real world, so that they can reflect some of the light from the monitors into the user's eyes.

When someone tries to choose between these two technologies, one has to take into account the advantages of each one. The optical display's main advantage is its simplicity, as the real world is perceived directly, in contrast with the video display which has to digitize the real images, correct for distortion in the camera and then combine the real and the virtual

images, adding time and equipment cost. Moreover, real-world's resolution is significantly better, as there is no camera capable of creating an image with the same resolution as the human eye. Another advantage is that the user sees the image on its right place, while there is an eye offset when video technology is used. Finally, using optical technology is more secured, as in the case of powering down of the system the user keeps on seeing the real-world scene, which does not happen with video technology.

On the other hand, using video technology offers better registration between the real and the virtual world in many aspects. First of all, in optical systems the virtual objects cannot completely obscure real objects, as the screen is semi-transparent. As a result, the level of immersion in the augmented world is better with video see-through displays. The same problem exists with the contrast, as the aforementioned different resolution between the two worlds doesn't permit the user to sense that he is really placed inside the augmented environment. Furthermore, the distortion, which increases as the user looks away from the center of the view, creates a translocation of the virtual objects. With the use of the video technology the distortion also exists for the real-world scene and it can be corrected for them both digitally, in order not to create registration issues between the two worlds. Last but not least, the possibility to get rid of the latency in order to allow the movement in the real and virtual views to be synchronized is only possible with the video displays. The latency is not really vanished, it just exists for both these systems, enabling their presentation as one unit.

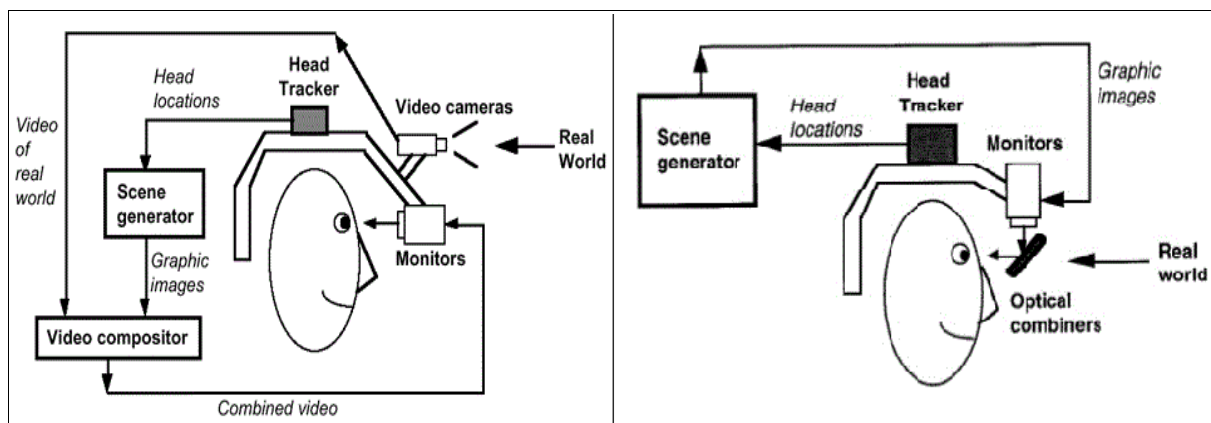


Figure 12: Video and optical see-through [3]

Head-Mounted Projectors or Projective Displays make use of miniature projectors or miniature LCD panels with backlighting. This way, a mirror beam-splitter is created, which redirects the projection frustum, meaning the region of space in the modeled world that may appear on the screen, so that the images are beamed onto retro-reflective surfaces, located in front of the viewer. Each one of the miniature projectors or LCD panels has the unique optical property to reflect light back along its incident direction. As a result, the images reflected are brighter than the ones created by normal surfaces that diffuse light.

Even if these devices decrease some problems of the HMDs, as the inconsistency of accommodation and convergence related to them and the parallax distortion that occurs if HMDs are worn incorrectly, they also have limits in their use. They offer limited resolution and the brightness of the images depends on the environmental lighting conditions, making their outdoor usage impossible.

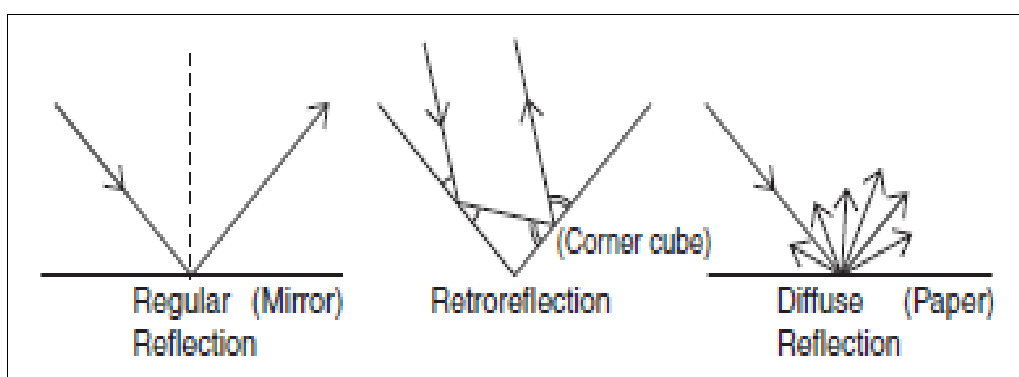


Figure 13: reflection properties [106]

During the last years the researchers are trying to create devices that can be used more easily, especially because of their portability. The result of this effort is the creation of two new head-attached displays: AR glasses and AR contact lenses.

The AR devices resembling glasses can either employ cameras to intercept the real-world view and re-display its augmented view through the eye pieces, or use a system which projects digital data through the surfaces of the eyewear lens pieces in order to complete the existing scene of the environment. Examples of this technology are (i) the Steve Mann's invention, EyeTap, which allows the user's eye to operate as both a monitor and a camera and (ii) the most recent creation of Google, Google Glass, which became available to the public in 2014 but its production stopped the following year, as it is being redesigned in order to be "perfect", as it has been declared.

The contact lenses that display AR imaging are also still in development. The prototype is going to contain the elements for display embedded into the lens including integrated circuitry, LEDs and an antenna for wireless communication [67].

2.1.3.2. Hand-Held Displays

Hand-Held Displays employ a small display that fits in a user's hand and they are portable and small-sized, in order for the user to be able to carry them with him. Personal Digital Assistants (PDAs), Tablet PCs and cell-phones have the possibility of combining processor, memory, display and interaction technology, all available in a single device which generates images at arm's reach.

Both video see-through and optical see-through technology is possible to be used, but the majority of the devices uses video see-through technology. With this, integrated to the device video cameras capture live video streams of the environment, which are overlaid by graphical augmentations and then displayed.

Hand-held devices have the advantages of the small size and portability, which make them a good alternative to head-attached devices for mobile applications. They also constitute the only technology with the prospect of gaining widespread acceptance, as the majority of people nowadays has a mobile device with a camera, capable of running an Augmented Reality application. Furthermore, more and more smart-phones with accelerometers, gyroscopes and compasses are released, while almost all of them already have GPS. These characteristics make them ideal devices for AR applications' development.

However, there are a lot of factors that create problems, mostly relevant with these devices' processors. The processing capability, the memory and the storage of hand-held devices introduce limitations to the complexity, not allowing the development of applications consisting on big 3D models or many graphics. The image analysis and rendering is possible to result delay and low frame rates. Another disadvantage is the limited screen size, which

restricts the covered field of view. In applications where the real and the digital world appear on a screen in the same time, the amount of the observer's visual field is the major factor that defines the degree of immersion. The fact that the user holds the device on his hand instead of wearing it, worsens the user's feeling of low immersion in a different reality even more. Finally, the limited quality for image processing tasks of these devices' cameras makes them effective only in a certain depth range.

2.1.3.3. Spatial Displays

The main difference between the aforementioned displays and the spatial ones is that it is not possible for the user to carry the technology of the spatial displays with him wherever he goes. In contrast, they are placed in position and they remain there, integrating the technology into the environment and obligating the user to go there to use them.

The Video See-Through and Optical See-Through approaches are two of the methods used by spatial displays, in order to augment reality. There is a third method used in this case, though, the Projection-Based Spatial Displays [10].

The Spatial Video See-Through Displays, sometimes referred to as “window on the world”, use a regular monitor to display the merged images of the created video. It is quite a common technique, as it only needs off-the-self hardware components and a standard PC in order to create and present the augmented scene, a fact that makes it the most cost efficient AR approach.

There are, however, many disadvantages originated by this method, with the most important one being the inability of the user to immerse into the augmented world, as his field of view is restricted to the monitor size, to its spatial alignment relative to him and to its distance to him. Smaller monitor sizes cause even smaller fields of view. It is also difficult for this method to support direct interaction with the real environment. Finally, the general disadvantages of the video see-through technology, as the limited resolution of the real environment, continue to exist in this case.

The Spatial Optical See-Through displays overlay the real environment with computer graphics in such a way that the graphical images and the image of the real environment are visible at the same time. The images are generated using spatial optical combiners such as planar or curved mirror beam splitters, transparent screens or optical holograms. Instead of following the user's movements, they may move around him.

The most important advantage of this method, is the easier eye accommodation, meaning the possibility of the eye to change optical power to maintain a clear image or focus on an object, as its distance varies. In addition, the calibration is more stable and the environment is better controllable, compared to the previous method. It also offers a higher resolution of the real-world scene and a larger field of view.

Speaking of resolution, the aforementioned incompatibility of the real and the virtual images' resolution exists in this method too, as well as the other problems of the optical see-through technology, relative to registration issues. Another important shortcoming is the incapability to support mobile applications, because of the spatial optical combiners being used. Last but not least, the limited size of screens and optical combiners causes the virtual objects outside the display area to be unnaturally cropped, an effect called window violation.

The Projection-Based Spatial Displays are a direct way of augmentation, as the images are directly projected on physical objects applying front projection, instead of being displayed on an image plane somewhere within the viewer's visual field. This offers the

possibility of generating images larger than the actual display device, anywhere in an interesting feature. In order to increase the potential display area, it is also possible to apply single or multiple static or steerable projectors.

The necessity for the use of stereoscopic projection depends on the applied changes. If the surface properties of the real objects, like the color and the texture, are the only things to change, a simple projection is enough, as the correct depth perception is provided by the physical depth of the surfaces. On the other hand, if new 3D elements are going to be displayed in front of the existing ones, stereoscopic projectors are required.

Apart from the theoretically unlimited field of view, the low cost is the other main reason why this method becomes more and more widespread. The eye accommodation is easier in this case too, compared to the non-spatial displays, as the virtual objects are typically rendered near their real-world location. Furthermore, the real-world resolution is not affected, while the new objects' resolution is scalable.

Nonetheless, several new problems are introduced. Firstly, the shadow-casting of whatever exists between the projectors and the surface, which is due to the utilized front-projection and makes the interaction with the augmented scene not possible. Secondly, when the surface to be augmented is non-planar, the projected images appear blurry, as conventional projectors can only focus on a single focal plane. Finally, as the number of applied projectors increases, the complexity of consistent geometric alignment and color calibration also increases.



Figure 14: Projection-based augmentation of a large environment [27]

2.2. Software Technology

When it comes to an AR system, a key issue is how realistically it integrates augmentations with the real world. In order to register virtual content in the real world, the software has to be able to locate the exact position where the augmented objects are meant to be placed, examining the images provided by the camera. To do this, the system needs to know where the user is and what he is looking at, which practically means that it needs to determine the location and the orientation of the camera, using the available sensors. Once the pose of the camera is known, the information is transferred to the processors, where

some functions are executed to combine the digital objects and the real-world view that the user will finally see through the display.

In the current chapter, the different methods for the detection of the user's position in different cases, as well as the way this data is edited by the processors to produce the final outcome are explained. The most serious problems of the technology that the researchers still try to face are also presented here.

2.2.1. Tracking Methods

Tracking, the function that detects the user's location, is critical to the accuracy of AR applications, as more realistic results can be obtained with accurate AR registration. For instance, inadequate registration accuracy can cause the user to reach wrong part of the real environment because the augmentation has been displayed in another part. The eyes of the users get used to the error in the virtual environment and after some time of use they start to accept these errors as correct, a situation which is not desirable.

Depending on the location of the sensor used, two methods are developed: the outside-in tracking and the inside-out tracking [24]. The sensor may be mechanical, magnetic, ultrasonic, inertial, optical or based on radio frequency identification.

When the sensor is fixed to a place in the environment and it detects objects (e.g. markers) that are placed on the user the method is called outside-in. The inside-out method is the opposite; the user carries the sensor and the markers are mounted around in the environment, at well-known locations, within the sensor's range. For the majority of AR applications, the second method is preferred, as the user has to carry a display device anyway, where the sensor can be adjusted.

Outside-in tracking can be used to produce a very accurate pose estimation, but the measurements of orientation are not as accurate as those of inside-out systems, where the sensor is on the display and the observed image changes as the user rotates the device. Furthermore, the range of an inside-out system is limited only by the number of fiducials placed in the world. On the contrary, to extend the space where the tracking is feasible in the outside-in case, it would be the number of cameras to be increased, which is not at all cost-efficient.

For the tracking with display devices such as smartphones, real-time location systems (RTLs) are typically embedded in them, the purpose of which is to detect the current geolocation of a target. Current RTLs are based on wireless technologies, GPS inputs and on optical ways of localization. Depending on the environment where the augmentation takes place, a categorization of different techniques can be done under three main cases: indoor methods, outdoor methods, and fusion strategies. A significant difference between indoor and outdoor environments is that the first ones are more manageable, thus more predictable, while the second ones are limitless in terms of location and orientation.

Indoor environments provide a structure domain for an AR application, with fixed dimensions. The structured domain ensures power for the tracking equipment and presents a controlled environment [24]. Furthermore, the user's movements are limited only to a specific region. These elements allow a wide range of tracking methods. The stable lighting and weather conditions are perfect for the development of vision-based solutions. Approaches using wireless technologies like Wi-Fi, Bluetooth and ultrawideband (UWB) are also feasible.

On the contrary, in outdoor working applications the light and weather conditions can change at any moment, making the optical tracking way more challenging. Another trouble causing factor when it comes to optical recognition is that the markers that enable the objects

to be found cannot last long in such conditions, which means that they will be destroyed sooner or later. The same problem exists when natural features of the objects are used instead of markers, as they can also be easily modified in external and uncontrollable environments. For all the above reasons, location-based tracking is considered a better option than the vision-based one when working outdoors [46].

All the approaches which can be used in both indoors and outdoors environments are further explained below.

The tracking method is called vision-based when optical sensors are involved. This kind of tracking has become very popular due to its minimal hardware requirements and improved computational power. There are three different techniques of vision-based tracking, depending on which type of optical sensors is used: Visible Light tracking, Infrared Light tracking and 3D Structure tracking. On the other hand, when the pose is estimated detecting the user's position the tracking method is called location-based.

2.2.1.1. Visible Light Tracking

The most common between the vision-based techniques is the Visible Light tracking, mostly because of the possibility to use simple cameras which can be found in a simple mobile device, like a smartphone. Thanks to that, applications working with Visible Light tracking can reach the masses, as they enable people to download them to their own devices. They can be divided into three categories: Fiducial Tracking, Natural Feature Tracking and Model Based Tracking [34].

2.2.1.1.1. Fiducial Tracking

Fiducials are artificial markers, the complexity of which varies depending on the application and the technology used. These markers, which serve as real world anchors, have geometric or color properties that make them easy to extract and identify in a video frame. In order to determine the pose of the camera and the scale of the object, a minimum of four points of known position need to be located. The points have to satisfy some requirements. For instance, if they are the corners of a square, it would be easy to find the object's position, but there would be no way to determine its orientation. There are two kinds of fiducial markers to meet these requirements; the Classical and the 2D Barcode one.

To ensure enough points are visible at any point in time, a simple way is to use a single planar fiducial, featuring a quadrilateral shape, the corners of which serve as the known reference points. This is the classical type. The quadrilateral shape is in white color, surrounded by black color and it includes an embedded high contrast image content, called pattern. The pattern is what makes the square marker unique.

The second type, the 2D Barcode Marker, is composed of predefined black and white patterns of various shapes and sizes, that look like QR codes. An example with squares in a simple grid arrangement (matrix pattern) can be seen in the Figure 15. Each unique matrix pattern is associated with a predetermined identifier. It is possible to use many of these markers in the same real-world scene, each one associated with a different object.

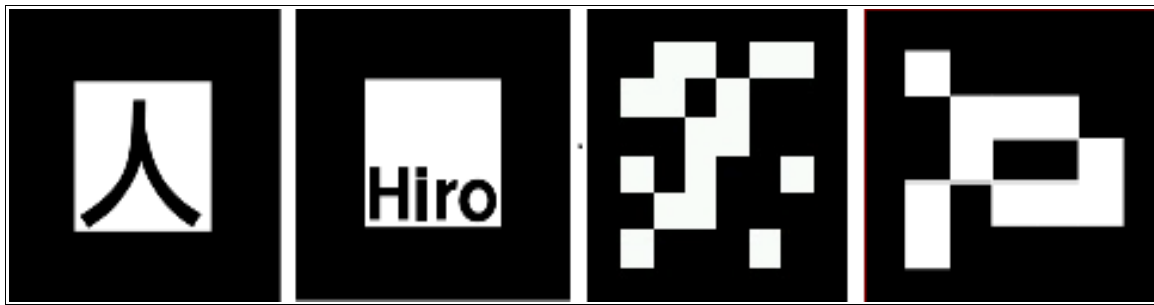


Figure 15: Classical Square Markers (the two first) and 2D Barcode Markers (the two last)

The simplicity and the high accuracy of this method made it an extremely popular technique for AR tracking. One of the most popular planar fiducial systems was the ARToolKit, the software development kit developed by Kato and Billinghurst in 1999 [62]. In ARToolKit, the fiducial markers are characterized by a thick black outlined square, generally filled with a black and white pattern for additional identification information. In order to determine potential edges of the markers, the system uses binary thresholding and partial line fitting. All the potential marker candidates, which means all the square regions, are used to calculate a six degree of freedom matrix transformation for the marker. After the transformation is determined, the system compares the internal pattern of each fiducial with a database of known patterns and matches the valid ones with the objects related to them.

Even though placing fiducials in the scene works very well for prototype applications in prepared environment, it is often undesirable because of the low aesthetic result they create, meaning the alternation of the scenery. Furthermore, the placement and maintenance of fiducials just may not be practical when dealing with large environments or even multiple instances of the same environment.

2.2.1.1.2. Natural Feature Tracking

As already explained, importing an artificial marker in the real-world scene is an important shortcoming of the previous method in some cases. For example, the modification is strictly forbidden or even not possible to happen when it comes to the augmentation of a work of art. For such cases, features that already exist in the natural environment have to be used. This requires complicated image-processing algorithms to detect features unique in their surroundings, such as points, corners and intersections of lines. A unique “descriptor” is then calculated for each one of the features detected. This descriptor allows the identification and differentiation of each feature, making the matching of them with objects possible.

Since matching features in real-time is difficult, markerless tracking systems typically operate under a number of simplifying assumptions, such as the assumption of smooth camera motion and frame-to-frame continuity. Some of the most common natural feature algorithms are the Scale Invariant Feature Transform (SIFT) [112], the Speeded Up Robust Features (SURF) [115], the Binary Robust Independent Elementary Features (BRIEF) [68], the Oriented FAST and Rotated BRIEF (ORB) [105], the Binary Robust Invariant Scalable Keypoints (BRISK) [55] and the Fast Retina Key points (FREAK) [56]. Each one of them has a different way of detecting features, with the last three to be created as developments of BRIEF. Even if it is difficult to distinguish one of these as the best algorithm, FREAK, which was the last one to be designed, has been shown to outperform the others. Testing environments have ranked FREAK as the most robust to image deformations and also the faster when it comes to computation time of the scale and rotation invariant descriptors [2].

2.2.1.1.3. Model Based Tracking

For this category, either a 3D reconstructed textured model of the tracking target or a predefined CAD line model, is used for the tracking [37]. The model is compared to the live view of the object in order to uncover the 6 degrees of freedom pose. For the initialization of the CAD based tracking, the user must align an initial pose of the line model to the actual object, which requires a certain level of expertise. Moreover, these systems have a quickly deteriorating performance in cluttered environments and especially when sudden illumination changes are involved. These factors make them less suitable for many types of applications.

2.2.1.2. Infrared Light Tracking

An infrared search and track (IRST) system is a method for detecting objects that give off infrared radiation, which is an electromagnetic radiation with longer wavelengths than those of visible light, therefore invisible to the human eye. It extends from the normal red edge of the visible spectrum at 700 nanometers to 1 millimeter.

For AR purposes, this method uses markers that are characterized as active, because they are luminescent themselves, contrary to the fiducial ones described earlier, which are considered to be passive [30][52]. Active markers are typically IR (infrared radiation) lights, mostly LEDs, that periodically flash. They are easily detected as they appear as bright spots in the camera images. Usually suitable filters are used on the sensor to make the markers the only thing visible to them, something that vastly reduces the difficulty and computational requirements for tracking. By synchronizing the time that they are on with the camera, it is easier to block out other IR lights in the tracking area, permitting to different objects to be tracked at the same time.

A disadvantage of this technique is that building active markers demands special equipment, such as an electronic circuit and batteries. That makes the method not only more difficult to be designed, but also more expensive.

2.2.1.3. 3D Structure Tracking

The 3D structure tracking is another markerless method, which attempts to detect objects using only features. The difference from the aforementioned natural feature tracking is that the environment in this case is unknown, which practically means that the sensor platform constructs its representation “on the fly”. The computational problem of constructing or updating a dense map of an unknown environment, while simultaneously keep track of a system's location within it has been named SLAM, Simultaneous Localization And Mapping, in robotic mapping and navigation [57].

Scanners, RGB cameras and RGB-D cameras (cameras that provide not only color information but also the estimated depth for each pixel), are only some of the sensors that can be used to estimate the position of the observer and to construct the 2D or 3D map of the environment. In the domain of Augmented Reality, the input of a Single monocular RGB camera is usually preferred. SLAM methods require a wide panel of algorithms to ensure the robustness of the provided localization.

In the present time most AR SLAM technologies enable only floor recognition and position tracking, to place digital objects around the user. This means that the system does not actually know what is going on in the environment to correctly react to it. More advanced SLAM technologies like Google Tango, can create a mesh of the environment, which offers the ability to identify not only the floor, but also walls and objects, allowing interaction with everything around [116].

The concept of SLAM has been around since the late 80s. Even if the 3D Structure Tracking is not as popular in AR applications as the previous methods, lately there has been an interest in techniques that use free tracking of unknown environments, mostly because of the projects of robot navigation that are being developed. Wikitude, one of the most well-known AR providers, has started including this technology in its SDK since 2015. The latest update of ARKit, the Apple's AR system, also supports it.

2.2.1.4. Location-Based tracking

The majority of the AR outdoor applications use tracking based on GNSS, which stands for Global Navigation Satellite System, sensors. A pure GNSS receiver, however, can only report position. In order to also receive orientation information, additional sensors like compasses and tilt sensors have to be adjusted to the system.

A disadvantage of this method is that the accuracy of a GNSS system ranges from several meters to a few centimeters. The correction of the calculation's accuracy with additional techniques, such as the use of reference signals transmitted by Differential GPS (DGPS) beacons improves the result, but not enough to be compared to the optical techniques. Furthermore, compasses are simple and generally cheap but not entirely accurate, since the earth's magnetic field is relatively weak. In addition, magnetic and metal objects nearby may cause a negative impact on the orientation accuracy.

High precision is always desirable, however not always essential. In simple navigation or commercial AR systems, for instance, the result may not be seriously affected if the digital object is not located in the exact position where it should be. In other cases, this method can only be used as part of a fusion strategy.

The inertial navigation system (INS) is a navigation aid that uses accelerometers, gyroscopes and occasionally magnetometers, to calculate the pose and the velocity of a moving object, relative to a known starting point. Inertial measurement units (IMUs) typically contain three orthogonal rate-gyroscopes and three orthogonal accelerometers, measuring angular velocity and linear acceleration respectively. By processing signals from these devices, it is possible to track their position and orientation, as well as the position and orientation of the system to which they are attached.

The main problem with this kind of tracking is that errors add up over time, causing drift. To take care of this, the velocity and/or position must be repeatedly corrected using another data source. That is what magnetometers are used for. With their aid, it is possible to detect the trackers' orientation relative to the Earth's magnetic field. The magnetic field is not an entirely accurate data source, but it prevents the orientation values from drifting too much.

As in the previous case, IMUs are able to stand alone when high precision is not essential, otherwise they can only be part of the system. They are not as widely used as GNSS, though, because of the cost demanded to support their technology.

Another similar approach suggests a network solution, which can be achieved with the installation of a wireless local area network (WLAN) to cover the area. WLAN is mostly used for indoor applications, where the most widespread location-based approach, GNSS tracking, is not feasible. For this method to operate outdoors, directional antennas have to be located in a number of access points, to implement the WLAN. The antennas can also link to their neighbors using secondary point-to-point links, in order to make the installation more stable. An application that took advantage of this technology, combining it with other tracking methods is Archeoguide, a personalized electronic guide to the outdoor historical site of Ancient Olympia, Greece, a system which is going to be explained later.

2.2.1.5. Fusion strategies

As already mentioned, some of the tracking systems are unable to provide all the necessary data on their own, especially in outdoor projects. The solution presented is hybridization, meaning multiple types of tracking devices working together, each complementing or correcting the others. Fusion systems can be either loosely or tightly coupled, depending on the interrelationship between the sensors [11].

An example of a fusion system where the different components complement each other is the combination of a pure GPS receiver and a compass or tilt sensors. This composition, though, is always used as a unit, resulting to it not being considered a hybrid system any more when it comes to AR applications. That is the reason why it has been mentioned as an independent, low-accuracy, outdoor method. A GPS sensor can also be combined with visual tracking methods. Whereas a visual tracking method may be able to detect the orientation of the camera, if the system is unaware of its location, it might take an unreasonably long time to examine all possible locations. As a solution, the system may use an approach where GPS indicates the location and visual tracking is then able to deduce the orientation in real-time.

Another popular combinational solution is the visual inertial tracking, due to the complementary characteristics of both sensors. Vision allows estimation of the camera position directly from the images observed and it is good for small acceleration and velocity, but it is not robust against 3d transformations. For inertial trackers, when standing alone, noise can result in an accumulation of position and orientation errors and they also have long term stability problems. When these sensors are used together, faster computation can be achieved and the drift errors of the inertial sensor can be corrected with vision. Applications generally use low frequency vision data and high frequency inertial data [13].

2.2.2. Pose Calculation in visual tracking methods

Summing up how the Augmented Reality technology works, no matter which components are included, the basic steps are three; the tracking of the object of interest through the sensors, the computation of their pose in order to accordingly modify the digital elements that are going to be superimposed on the real-world scene and their projection on the display's screen. The first and the third steps have already been analyzed, but the question of what transformations are applied to the computer-generated elements is yet to be explained.

There are a lot of different methods for this procedure, depending on the exact components that constitute the system, making the reference of them all exceeding the purpose of this thesis. As the visual tracking methods are the most popular ones and they all share the same basic idea of pose calculation, an explanation of the transformations needed in this case is sufficient.

In every application that involves a camera, the task is to detect elements in a frame. These elements may either be markers, natural features, lines or specific shapes. Their 3D physical relations such as the shape and the size of the markers or the relative position of the object's unique features are already known, permitting their comparison with the resulting image. Assuming that the camera's internal structure is also known from the calibration report, the image coordinates, x , are modified with the procedure of interior orientation and are corresponded to the real-world coordinates, X . In the scientific field of computer vision, the procedure of transforming the three dimensional projective space to the two dimensional space of an image is achieved with the central projection [81], which is expressed through

homogeneous coordinates, using the equation

$$x = P X \quad \rightarrow \quad \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix}$$

where $X = [X \ Y \ Z \ W]^T$ is the vector that describes a real world point and $x = [x \ y \ w]^T$ is the vector that describes its projection to the image's space, both expressed in homogenous coordinates. The 3×4 matrix P is called projective camera matrix. The 3D plane that intersects with the axis X , Y is the front side of the camera, or principal plane, where $w=1$, which is why the equation is usually expressed as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

With the assumptions that the origin of the coordinate reference system is the principal point, meaning the point where the optical axis intersects the image plane, the center of projection is located at $(0,0,0)$ and the principal axis is parallel to Z axis, as seen in figure 16, the relations

$$\frac{c}{Z} = \frac{x_c}{X} \quad \rightarrow \quad x_c = \frac{c}{Z} * X$$

and

$$\frac{c}{Z} = \frac{y_c}{Y} \quad \rightarrow \quad y_c = \frac{c}{Z} * Y$$

can be used, where c is the principal distance of the camera, meaning the distance between the image plane and the projection center and x_c , y_c are the image-coordinates of the point (X, Y, Z) . In this case the equation can be written as:

$$\begin{bmatrix} x_c \\ y_c \\ c \end{bmatrix} = \frac{c}{Z} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

However, in general, the origin of the coordinate reference system is not the principal point. In computer vision the origin of the coordinate reference system is usually at the top-left pixel, which means that the principal point's coordinates, x_0 , y_0 , expressed in pixel size, also have to be added. The image coordinates are thus described by the following relations, in pixel size:

$$x_c = \frac{c_x}{Z} \cdot X + x_0$$

and

$$y_c = \frac{c_y}{Z} \cdot X + x_0$$

In these relations, c_x is the camera's principal distance, expressed in size equal to the length of the pixels' width and c_y is the camera's principal distance, expressed in size equal to the length of pixels' height. As a result, the equation of the central projection can be written

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} c_x & s & x_0 \\ 0 & \alpha c_x & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where s is the skew, a factor only used if the pixel array in the sensor is skewed and can in most cases safely be set to zero and $\alpha = c_y/c_x$, is the aspect ratio, which is equal to 1 for square pixel elements. The 3x3 matrix can also be written

$$K = \begin{bmatrix} c_x & s & x_0 \\ 0 & c_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

and it is called the Intrinsic Calibration Matrix [74]. It describes the projection properties of the camera. It only depends on the camera properties, as the camera's principal distances are the only unknown variables that have an effect on the image formation.

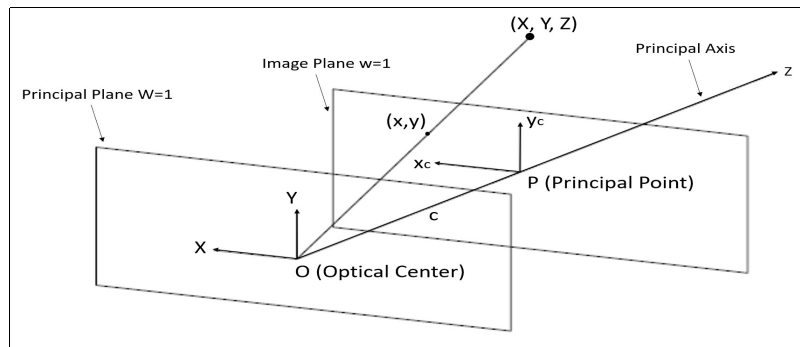


Figure 16: Central projection geometry

Another factor that has to be taken into consideration is that in general case, the center of projection is not located at (0,0,0) and the principal axis is not parallel to Z axis, which means that the coordinate system has to be rotated and moved. For this reason, the image coordinates are finally described by the relation

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} c_x & s & x_0 \\ 0 & \alpha c_x & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

which can also be written

$$x = K [R | t] X \rightarrow x = K [R | t] X$$

$[R|t]$ is the Extrinsic Camera Matrix [73], which describes the camera's location in the world and what direction it is pointing at. It has two components; $R_{3 \times 3}$ is a rotation matrix that represents the directions of the world-axes in camera coordinates and $t_{3 \times 1}$ is a 3D translation vector that can be interpreted as the position of the world origin in camera coordinates. The matrix can be written as

$$R | t = \begin{bmatrix} r1,1 & r1,2 & r1,3 & t1 \\ r2,1 & r2,2 & r2,3 & t2 \\ r3,1 & r3,2 & r3,3 & t3 \end{bmatrix},$$

where r_{ij} are the elements of the rotation matrix and t_i are the elements of the translation vector. It is important that the Extrinsic Matrix describes how the world is transformed relative to the camera, not how to transform points in camera coordinates to world coordinates.

An actual camera produces systematic geometrical errors, distortions, which occur in the camera lens due to lens imperfections. The camera model used in this case assumes that the principal point coincides with the center of distortion, an assumption which is eligible as most lenses currently manufactured do not have imperfection in the centering. This part of the distortion can thus be separated from the camera model.

The real location of each point in world coordinates is known. The system has one equation for each of the two coordinates of each of the points and six independent parameters. Thus, it is possible to estimate the transformation matrix. Once the transformation is applied, the virtual object is registered on the screen to the real world [44].

2.3. Today's challenges

Augmented Reality is a constantly evolving technology, which, nowadays, lies at the core of the technological world, as it is being introduced in more and more fields of life. However, as it is getting ready for mass consumer adoption, there are some significant challenges that must be overcome.

The first and maybe the most important challenge is relevant to the latency, which is being introduced to the system during the tracking, because of the time required for the image viewed by the camera to be analyzed. Each device has an associated delay between its observations of the world and the moment when the AR display presented to the user appears to be affected by a change in the data. These differences in delay, called relative latencies, are the main source of misregistration, although they are in the range of 30 ms. According to Holloway [20], 1 ms of latency can cause 1 mm of registration error, which is a potential source of up to 30 mm of registration error.

Latency is not the only problem to be faced when trying for a correct registration. Tracking is one of the most important enabling technologies. Even if AR tracking systems has evolved from simple fiducial marker-based systems to natural feature tracking, there is still need for significant work to be done, before the vision of augmentation systems that work no

matter the circumstances is achieved. There are also a lot to be done in order to achieve satisfactory results under various circumstances, as, for example, in the case of changing illumination, in order to be able to seamlessly blend the physical and digital worlds.

Ivan Sutherland had a vision of creating the “Ultimate Display” [47]. He imagined it as a room, within which the computer can control the existence of matter, meaning that anything displayed in such a room could actually be used as a real object. Current displays are still very far from such a system, which shows that there are important research opportunities available in the design of displays. The ideal display would be one that provides a wide field of view, supports occlusion of the real world when an imaginary object is in front of it and vice versa and provides images at a variety of focal planes, all in a small unencumbering form factor.

The next problem the researchers are trying to face is the unreliable computing power. Processors are growing stronger and smaller as they evolve. The fact that they are getting stronger has led to the creation of new algorithms, like those for natural feature tracking and new possibilities in every aspect Augmented Reality technology. The fact that they are getting smaller is the reason why it is now possible to use more and more portable devices for AR applications. Nevertheless, these devices also require powerful batteries. Smartphones, for example, are getting more powerful as years pass, but their battery lasts less time, as they have too many applications running, contributing to its need for recharging. The balance between the high-capacity, the low-consumption and the small size has yet to be found, in order the wearable displays to be able to support the new applications.

Finally, there is a problem relevant with the audience impact. Augmented Reality is a technology that looks like it came from science fiction movies and at the same time, a lot of applications that could be used in everyday life have already been released. Nevertheless, it doesn't seem like people actually believe that this technology is going to change their future life. As a result, until recently it was mostly used as a marketing tool for advertisers, which is not at all representative of its possibilities. A number of researchers had raised social acceptance as an important issue for AR and were examining the element missing for the applications to become compelling, something that could result in mass adoption of the technology. During the last year it seems like a change has started occurring, as the number of people beginning to get interested in exploring the new applications is being constantly increased.

2.4. State of the Art – Cultural Heritage AR Applications

As mentioned above, Augmented Reality can provoke important changes in the Cultural Heritage field when it comes to the presentation part. An Augmented Reality-based heritage experience gives the visitors the opportunity to feel they are present at significant places and times in the past and use a variety of senses to experience what it would have felt like to be there. This possibility adds a new dimension in understanding history, enhancing education and increasing the entertainment level of a visit.

This fact has become more obvious in the last years, leading to the development of many relevant projects, which have been introduced as tools, either to promote scientific work or to be used as electronic aids, providing users with information. In these projects, either the item of interest's special characteristics are highlighted or, in cases when it has been modified over the years, its original form can be viewed. Some of these are the ones following.

2.4.1. Archeoguide

One of the first personalized electronic guides to outdoor archeological sites is Archeoguide [51]. The prototype was built in 2001, around the historical site of Olympia, Greece, which is the birthplace of the ancient Olympic games, currently mostly composed of ruins. A combination of mobile computing, Augmented Reality, 3D visualization and networking is used, in an attempt to offer both on-site and online tours of the physical site to the users. Apart from the building reconstructions, the project also introduces augmented videos showing avatar athletes competing in the ancient Olympia stadium.



Figure 17: Archeoguide – virtual reconstruction of ancient temple [51]

The system contains a site information server, which is used as the central repository for archiving multimedia information, mobile units based on laptop, pen-tablet and palmtop computers and a wireless local area network infrastructure, to communicate the information provided by the server, to the clients. Touring users in the archaeological site carry the mobile units and wear an HMD device with variable transparency augmented reality glasses.

The server is the most important component, as it is the brain of the system. All the objects necessary, such as 2D images, 3D models, video clips, text and audio objects are stored there, organized in a hierarchical tree structure. The attributes of each building part and each artifact can include name, type, date, location, category and detail level, so as to make it easier for the suitable part to be chosen and presented each time when an item is called by the system. Using this type of organization makes it also possible for the users to design their own tours on a digital map and having the related audiovisual information provided to them while they follow their route.

For the server to provide the suitable information, the user's position is constantly calculated by GPS signals received from satellites. The tools used to identify not only the user's position, but also his view, DGPS and compass, are attached to the mobile units. These units are also the devices where the tracking and the rendering are taking place, which means they have to contain a really strong operating system, capable of handling heavy processing in short response time. The final accuracy achieved is less than 1 meter for the position and 1.5 degrees for the viewing angle [51].

The design of the network infrastructure is also an important factor, as the waiting time has to be minimized as much as possible. Even when the majority of the information is downloaded at the tour's beginning, there will definitely be a necessity for updates defined by

the users' movements and point of view. To cover the whole site, three access points with directional antennas which implement the WLAN are required, located outside of its perimeter to eliminate both physical damage and visual disturbance. Solar panels were used to power the access points. An additional wireless link was also needed to communicate correction signals to the DGPS receivers. With the specific design, the network can support up to 50 users at a time with a bandwidth of 11 Mbps.

The most important decision for every Augmented Reality project is the method which will be used for the tracking and the rendering to be accomplished.

Although it has already been said that a location-based tracking has been chosen, for the specific project the creators have decided to also use vision-based tracking, to increase the realism of the augmentations' integration with the real world. For that purpose, the Visible Light tracking has been chosen. Because of the nature of the items of interest, using markers would cause aesthetic problems, which is why the Natural Feature option was selected. However, as this approach was considered fragile and uncertain in real time applications back in 2001, the Archeoguide project uses a method that reduces complex 3D scene tracking and user motion to a 2D problem. In this method, a set of five to ten calibration reference images are compared to every frame of the user's view live video image and a correlation score is the criterion to decide the most suitable reference image.

The next step is the evaluation of a 2D transformation of the image decided by the tracking procedure. The transformation is described by a matrix, containing the translation, rotation and scaling parameters. The same transformation is applied to the 3D models of the reconstructed monuments which are aligned to the image. Afterwards, the transformed 3D models are rendered on top of each live video frame, offering to the user the view of the augmented world, through special glasses he wears.

The final evaluation of the project was positive, as the users praised the AR glasses for realism and said that the avatar animations helped them better understand the site's history and use. However, further work was essential in some points.

There are some problems possible to occur when it comes to the use of reference images for tracking. For example, new objects, even visitors, can intervene in the scenery, changing a part of the live image the system receives. This is why an algorithm using all the pixels of the image is implemented, rather than one that uses only local properties. The light conditions can still change because of the different position of the sun or the clouds, but the new shadows or highlights created do not significantly affect the result. The Fourier-based approach which is used for Archeoguide is effective in such differences and also fast, but it limits the user's viewpoint and forces to a restricted camera motion.

Another negative factor emerges from the "reduction of the dimensions" mentioned. A small error in visualization is introduced to the models' reconstruction, as their final view is possibly slightly different than the one it was supposed to be presented.

Furthermore, the necessary equipment was considered inconvenient to be transferred by the users, which is logical, especially taking into consideration that in 2001 the laptops were bigger and heavier than nowadays.

2.4.2. LIFEPLUS: Revival of life in ancient Pompeii

In 2002, a group of scientists decided to bring ancient frescos of ancient Pompeii to life, through a 3D animated revival of their content, superimposed on a creation of their historic environment [35]. The goal of this project, named LIFEPLUS, was to expand the limits of Augmented Reality, while helping the public grasp and understand the culture of the

ruined city.

In LIFEPLUS, a guided tour of the streets of Pompeii in AD 79, just a few minutes before being destroyed by a major eruption of the Vesuvius volcano, is presented. Besides the development of the mobile AR guide, virtual simulations of fauna and flora are also included, allowing the addition of characters to the AR experience, which was an innovation at the time. These characters perform historical narratives, based on the scenes represented in the archaeological site's frescos. Thus, visitors can experience a high degree of realistic immersion.

As for the majority of cultural heritage applications, also for the LIFEPLUS case, the modification of the scenery would cause problems to its verisimilitude, which is why fiducial markers are not allowed. As a result, the tracking was markerless and the image recognition had to be done only with natural features. To achieve high speed computation, instead of testing the relative changes in rotation and translation, the specific system calculates the absolute ones for every frame. This leads to instant detection of the designated area after tracking has been lost because of excessive motion blur or occlusion. To accomplish that, the operation occurred in two stages; one for recording the scene for a few seconds, which only has to be performed once by the animators and one for displaying the augmented items, after a database of key features and structure of the scene has been created by the tracking setup software. That meant that a video-see-through display was being used in this case.

Aside from the video see through monoscopic HMD, which is necessary for the aforementioned procedure, there is also need of a Mobile computing equipment, an earphone and a firewire camera. Firewire cameras use an interface standard for a serial bus for high-speed communications and isochronous real-time data transfer, thanks to which they are able to transmit audio, video and control data. Using this equipment, the system is able to enhance the reality with a combination of technologies, like real-time character rendering in AR, real-time camera tracking, facial stimulation and speech, hair simulation, body and cloth animation, skin rendering and interactive shading, 3D sound, plant simulation and behavioral scripting of actions. In addition to these, a DGPS and Digital Compass support is also used for the on-site localization of the visitor.

For the realization of the project, a deep study of realistic cloth simulation systems has been done for the first time. The research, from which a new simulated model is created, is focused on the real-time constraints for the simulation and the visual cloth motion features to which the observer is sensitive. It includes categorization of the clothes in relation to the geometric deformations that happen to them according to the body model movement, with different method used for the ones that stick to the body and for the ones that move within a distance to the body surface and for the ones flowing around it.

A face modeling study has also been fulfilled, from which a database of human faces has been created. The initial models used are based to the ones from the paintings, to which interactive texture mapping has been added. The real time facial emotion animation includes the usage of both static expressions, like smile and sadness, which has to be done for each face-model separately and parameterized mesh deformation algorithm which can be applied to any compatible facial mesh. Another disadvantage of the first method is its data storage space requirements. However, the aesthetic result is better and more precise than the one created with the parameterized approach. For the realistic speech animation, a data base of facial expressions corresponding to the voice signals has also been created, as it enables more natural mouth shapes.

Finally, the designers considered important the enhancement of the scenery with real time plant simulation, for increasing the immersion and interaction. The virtual plants are species developed according to the spatial and historical contexts, a fact that contributes to

the cultural meanings of the project.

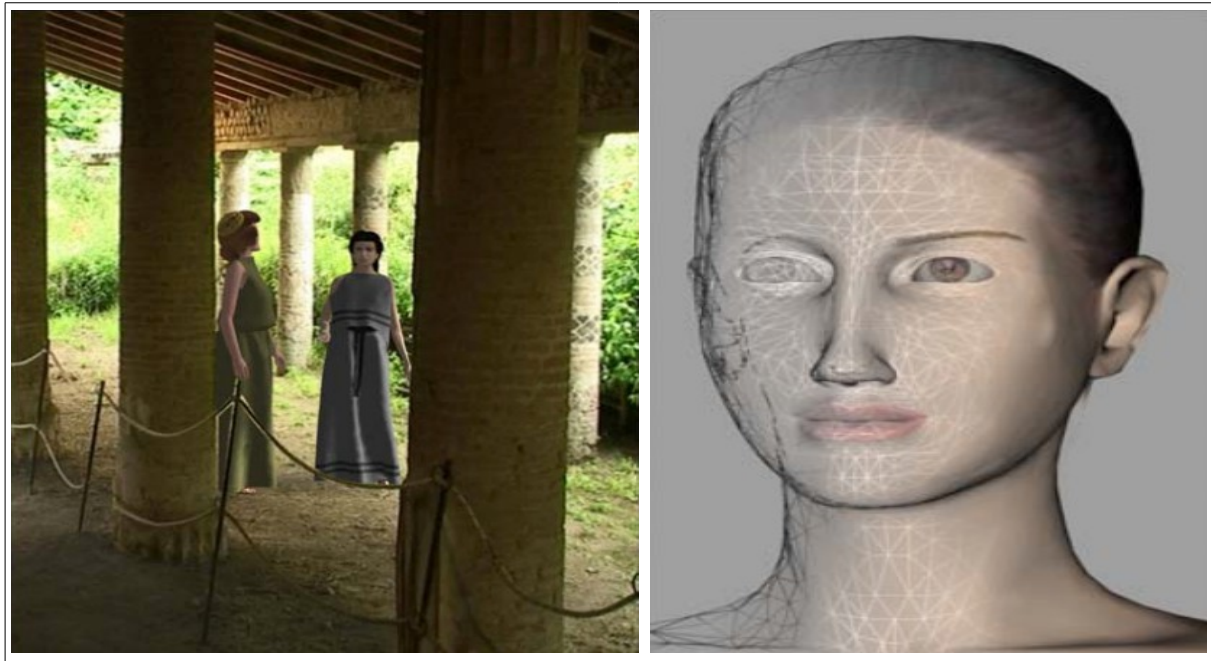


Figure 18: LIFEPLUS - Augmented View (left) - 3D facial mesh model (right) [35]

Eight years after the project's initial publication, a second version has been released, where the modelling and the animation of the virtual humans are described [28]. According to this, the models have originally been created by automatic generation methods and then manually edited and refined. In this way it is possible to provide realistic results of any body-geometry, modifying the base parameters of the template models. Therefore, for the motion addition to the models, a VICON Optical Motion Capture system based on markers has been used. In order for the movements to be ascribed with realism, real actors rehearsed the scenarios that were prepared with the assistance of the historical advisors. Their movements have then been converted to keyframed animations and have been connected to each other, to create blending types of motion.

2.4.3. Calakmul – Ancient Maya Metropolis

Calakmul is an important ancient Maya metropolis located in the State of Campeche, Mexico and situated deep in a tropical forest, which is why it was only discovered some decades ago. Although it has been declared a World Heritage site by UNESCO, none of its findings had been exhibited and no museum had been established until recently. The ancient buildings are still partially hidden by the trees that create a protected Biosphere, which has to remain untouched.

The construction of a site museum with virtual objects and a walkthrough in a virtual recreation of the city was believed to be able to alleviate the difficulty of appreciating the urban design and the characteristics of the important structures. The possibility to not only visualize them but also interact with them, would provide a better understanding of the findings and would attract more visitors. That is how the idea of applying virtual technologies to the city was born. The relevant project was realized in 2004 [43].

Augmented Reality was decided to be used for the presentation of the tomb of king

Jaguar Claw. The tomb is one of the most interesting findings in Calakmul, although it is currently ruined. Using AR technology, the visitor is able to admire two different views of the funeral chamber: Its actual current condition and its past view. The second one is presented with a virtual recreation of its hypothetical original elements, according to laboratory tests, which are superimposed over a view of the remains.

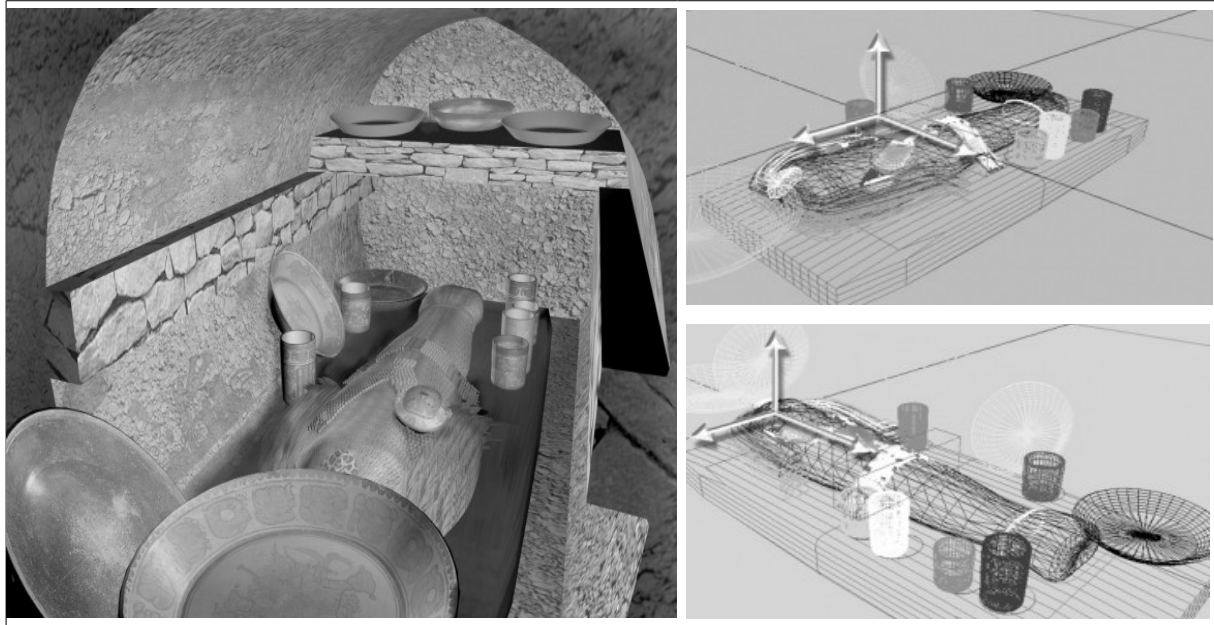


Figure 19: Ancient Maya tomb – virtual reproductions [43]

The project allows for the tomb to be viewed by visitors from specific positions, through AR glasses. The 3D model is the reconstruction of the entire burial with all its offerings. Three different patterns of this model have been created, saved in three files from different origins. The type of display can be described as a video see through Head Mounted Display, adjusted as an output device. The AR glasses play the role of the display in this case.

For the development of the application, the ARToolKit software has been chosen. The code has been used without important changes, with its original fiducial tracking function. The video is recorded by a web camera and transferred to a computer, where the processor converts it to high contrast images. These images are being searched by the software's algorithm for possible patterns. Their orientation is calculated by the system with the comparison of the patterns found in relation to the original orientation of the saved reference images. Once the pattern, as well as its relative position and orientation, are identified, the corresponding model is superimposed on the image with the same position and orientation.

The project's logo was used as the pattern to be tracked, with some modifications applied on it. In particular, the changes pertained to its colors, which were slightly adjusted so they could blend into a black and white square when the images were converted to high contrast ones. The logo did not significantly affect the museography appearance.

2.4.4. The PRISMA project

In a lot of tourist destinations it is common for someone to encounter tower viewers. A tower viewer could be a telescope or a pair of binoculars, permanently mounted on a rock,

usually located at high altitude. Their purpose is to give to the tourists the possibility to admire the landscape or to view attractions and events of interest, and thus help them decide which places they would like to visit later on. However useful they may be, there is an important limitation when they are used for something more than just admiring the landscape: Without additional information, it is really hard for people not familiar with the area to understand what exactly they see. Even in cases it is easy to understand the type of the attraction from its features, it is really hard to distinguish further details as e.g. its name or its location.

In 2005, an innovative way of enhancing the view seen by such devices has been presented in the PRISMA project [69]. The idea was the creation of augmented binoculars, a device which would superimpose animations, virtual recreations and other multimedia information that the observer could retrieve, about monuments and historical buildings of a city. The aim of the project was not of course to replace the visits to the real environment, but to increase the tourists' interest of it.

In order to test how the aforementioned idea could work, the augmented binoculars have been located on one of the hills of Donostia-san Sebastian city in the North of Spain. Two of the main tourist attractions of the city can be seen from this spot: Santa Clara Island, situated in the middle of the bay and the Cathedral. The information which is being augmented by the system includes texts, like directions and telephone numbers, 2D images, such as maps, photographs and videos and 3D models. The user is able to choose what kind of information he is interested in, through a pop-up menu that appears on the screen when he clicks a marker of an object.

The technology of the project is using the video see through Head Mounted Display, adjusted as an output device, with binoculars to play the role of the visualization device. For the tracking, a camera is necessary to capture the spectator's field of view and an inertial sensor to track the rotation and the orientation of the binoculars. There is also need of a processing unit to perform the essential operations, such as to convert these data in an orientation vector, to synchronize the images, to add the multimedia information to every frame and to send the augmented stream back to the display.

For the development of the application, ARToolKit libraries have been used as base. The initial code has been modified to calculate the camera position and orientation from the data collected by the inertial sensors rather than using physical markers for the tracking.

What importantly diversifies this project from all the others presented so far is that the scene that has to be augmented is spherical. The graphical objects need to be superimposed on this scene, which means that they ought to be placed with a method that takes into consideration the aforementioned factor. For this to happen, the method created provides information not only for the virtual objects' distance from the spectator, but also about the angles they form with the Zero-View-Vector.

A "Cameramove" method transforms the data from the inertial sensors into two floating values which correspond to the horizontal and vertical angles from a reference position. These values are used to synchronize the virtual camera movements to the ones of the real camera. However, in the demo edition the real camera had a specific velocity, in which the virtual one was synchronized. As a result, there was no need to use any sensor data.

The PRISMA project offered the initial idea, but a lot of progress has been done since then. Nowadays the user can zoom and navigate in specific areas and, as in a recently developed project, visualize an area as it was at the past or even as it is predicted to be in the future [16]. The possibilities of future work are endless, restricted only by people's imagination limits.

2.4.5. A Future for the Past

“A Future for the Past” is the name of an exhibition which took place at Allard Pierson Museum in Amsterdam for the celebration of its 75th anniversary, in 2008. During this exhibition Augmented Reality technology was used to present two excavation sites, which have their beginning in the Roman Empire period: Satricum, a Latin village situated on an acropolis and the Roman Forum, situated at the center of the city of Rome. The purpose was to help the audience better understand the connection between the archaeological artifacts of the museum and the places where they were found.

For this project, the original sites are brought back into the museum as large scale photographs, where AR information layers are superimposed. The displayed data are presented in the form of text, 2D images and videos and 3D reconstructions. The points of interest artificially added to the pictures help to bring together the past, the present and the future of places, which are situated far away from the museum, but still tightly linked to it.



Figure 20: The project's visualization devices – The revolving Movable screen (left) – UMPC (right) [53]

The system consists of three parts [54]. The static part, which basically means the large-scale photographs of the excavation site, the computer-generated data and the visualization device that connects them. Two different visualization devices have been used, both with the video-see-through technology. The first one is the Movable screen, a stationary display that can be rotated around 360 degrees, popping up the information associated with the tracked points of interest in a slideshow form. The second one, UMPC, which stands for Ultra-Mobile Personal Computer is a tablet that enables the appearance of relevant videos, images and description texts that the visitor can take his time to read, before he heads to the next visualization.

The tracking process was developed within the iTACITUS project, presented in [53]. For the recognition of the spots that initiate the augmentations, reference images are used. For each frame, the system analyses the image from the camera to find out if it is a given scene. To reduce computational time, only one reference image is tested for each frame. When a spot is detected, the system switches to the frame-to-frame tracking phase, following the position of the project. Its position and orientation are computed and the data to be displayed are transformed to the relevant pose.

2.4.6. The House of Olbrich

In the majority of the projects mentioned so far, the necessity for heavy wearable devices was considered a disadvantage by the users. The technological evolution, though, has permitted smaller and lighter devices to be able to support projects that used to demand a lot of space and computational power. Thanks to that, Augmented Reality has the potential to become a mainstream medium, ready for mass markets. This project, developed on 2011, presents an AR history and architecture guide that realizes the visualizations through a mobile phone.

The Josef Maria Olbrich House, used as the main example, is the leading architect's residence of the Darmstadt's Mathildenhöhe Artist's Colony, where a group of Jugendstil artists lived and worked. The building was famous for its artistic elements of Art Nouveau, but it was destroyed during World War 2 and has been only rudimentary restored ever since. The application presents elements concerning its architect, design and history, once the user takes a snapshot of it [23]. 3D reconstructions are also available.

In this case, the developers decided to apply computer vision techniques of 3D feature recognition and to face complex tracking situations, as lighting and weather condition changes and occlusions. GPS and compass readings are still used, but only as an initial check, to detect whether the user is close enough to the building for the augmentations to work properly.

When the user is on a right position, he has to take a photo of the facade. The picture is then sent to the server for tracking. The installation of markers on the facade is not possible, which has lead once more to a Natural Feature approach that works with reference images. If the image taken matches with one of the reference images, the system calculates the pose, makes the necessary transformations and adds the augmented result, which reveals how the building looked like before it was destroyed, through sketches and yellow hotspots. The yellow hotspots indicate the elements missing from the current view. Each one of them corresponds to an info panel with Virtual Reality 3D reconstructions, detailed textual information and featured media content.

Keeping the application's structure simplified as described was the way the designers have chosen, for the application to remain unaffected to the conditions that usually ruin the outdoor AR applications. For example, the constantly changing lighting conditions would demand too many models to be constructed in order to assure a photo realistic result for each case. Using the sketch, these alternations do not affect the augmented view, allowing thus the building of a light app construction that the hand-held devices can process. The 3D reconstruction is still supported in an off-line edition. In fact, the users can also interact with the model, as they are able to rotate it or zoom in and out.

In case a tracking failure occurs, there is also a backup plan, which works with the common method. The GPS and compass readings are used to calculate the approximate user's position and a 3D model appears. The model reacts on user movements and its pose changes as he is walking around the building.

2.4.7. The CHESS project

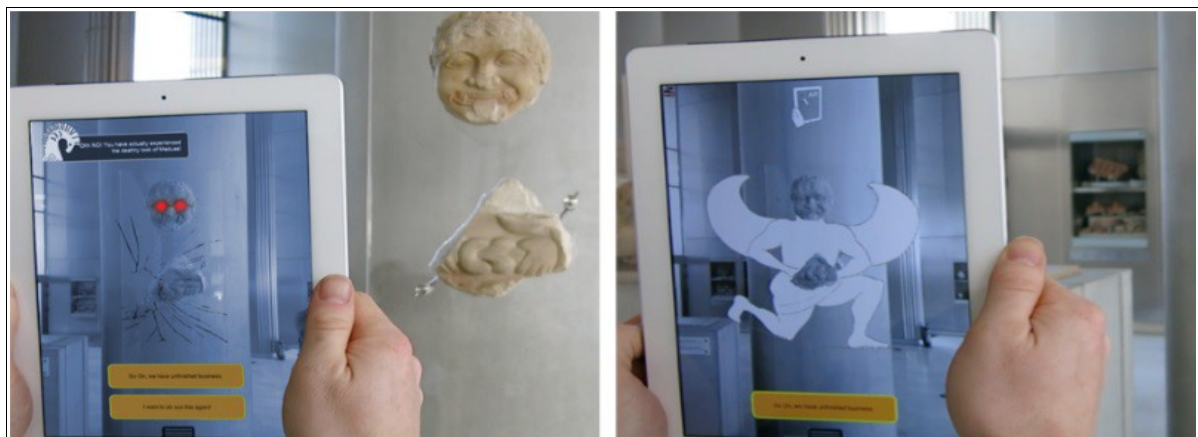
As handheld devices became popular in the field of AR, they were included in a lot of new projects. The majority of them, though, were enhanced museum guides, using augmentations mostly to provide interpretive information about the exhibits. The CHESS project, developed in 2012, is among the first ones to provide handheld Augmented Reality storytelling experiences, illustrated by museum exhibits [36]. It aims at designing and

evaluating interactive stories for visitors of the Acropolis Museum of Athens. The stories presented are personalized to the audience characteristics or preferences, offering alternative plots according to the visitor's profile, position and behavior. They may consist of narrations, images and videos, 2D or 3D models, games and any combination of them.

An issue which concerned the designers was the level of mediation between the physical objects and the augmented data. Two approaches are used in the project. A formal one, where the purpose is to explain the exhibits in a descriptive way and an explorative one, which is focused on the creation of an experience not only to inform but also to entertain the users, containing a higher level of interaction. The second one is mostly targeted to increase children's interest in plots-cases designed as games. Nevertheless, it is also used in versions oriented toward adults, like the case of the Kore statue, which, through a first-person narration, is transformed into a character with its own history and attitude.

A server-client architecture working with wireless networks is the base of the system. The server decides the plot that is going to be presented, taking into account the visitor's profile. A management component is integrated, to ensure that the activities proposed are available on time. Once the system has managed and composed the activities, it sends them to the client, meaning the mobile devices. For some of the activities, however, the majority of the data is already stored on the mobile device, to avoid increasing loading time which may result from network overload. The client has an application, named instantAR, installed, to receive the data and take charge of the rest of the procedure. The framework, which is designed for iOS devices, includes a browser component, a tracking and a rendering engine and a video preview in the background.

As the project is realized in a museum, it is important not to clutter the environment or the archaeological original exhibits with artificial markers, which is why the Natural Feature approach has once more been chosen. With the development of a high accuracy Computer Vision based tracking of 3D targets, it is possible to work with the physical objects interactively. The detection of the special features for every item to be augmented has been completed in a post-processing step and 3D feature maps have been generated and added to the mobile devices. Both them and 2D reference images extracted from the videos that are recorded by the visitors, are used for the tracking.



**Figure 21: The Medusa scenario [22]–
Plot designed for kids (left) – Original shape of the statue, visually reconstructed for the adults (right)**

As poor lighting conditions and occlusions are usually two of the most trouble causing factors in AR applications, what really facilitates the tracking process is the way the Acropolis

Museum is constructed. The Archaic Gallery, where a lot of the archaeological objects linked with stories are located, has very stable lighting conditions throughout the day. There are also high marble bases for the exhibits, thanks to which it is easier to avoid obstacles that could be interposing between them and the camera. Even in this case, on crowded days it is getting difficult for the user to move around the object, keeping his distance from it constant.

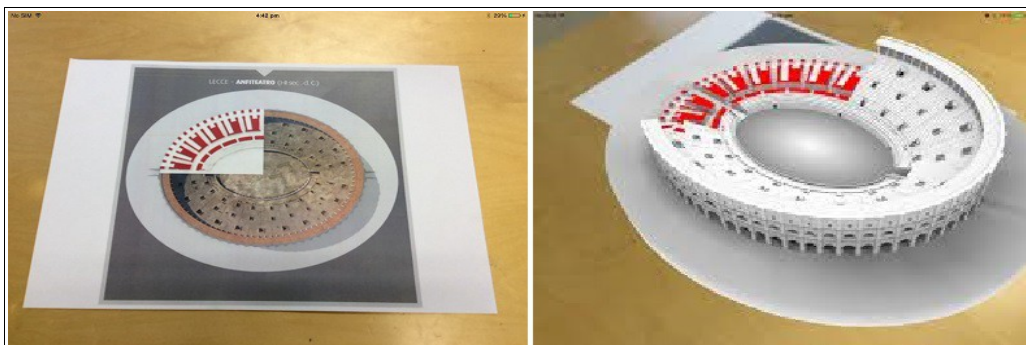
What makes the situation harder is that the method of the tracking procedure used requires a specific initial position of the camera. This problem is caused because the white marble surfaces of the statues do not have distinguished features that can be extracted from all viewing angles. The tests indicated that the visitors are often confused, as they don't know exactly where to stand and they are wondering whether the system works or not. A strong guidance from the system is required to overcome this problem.

2.4.8. LecceAR

Lecce is a historic city in southern Italy, commonly nicknamed “The Florence of the South” because of the rich Baroque architectural monuments that have been found there. Its Roman Amphitheater, situated in Piazza Sant’Oronzo, is believed to be the symbol of its ancient origins. The LecceAR project, which was part of the opening exhibition of the Living Lab in the MUST museum in Lecce, provided visitors with a 3D reconstruction of the amphitheater, where lost or hidden details were visible [4].

The application is designed to run on mobile devices like smartphones and tablets, superimposing the 3D model on top of a target image, which can be either synthetic or natural [4]. The image used was a floor plan of the building. For the construction of the visible structures' model, data gathered by both laser scanning and digital photogrammetry technologies have been merged to identify morphological and textural elements. Nowadays, only a third of the ancient building is unearthed. To complete the original appearance, thus, a lot of assumptions about the missing part have been made. The final shape has been determined according to the construction techniques in use in the Roman period.

The application's matching and tracking module is responsible of recognizing a known target image in the field of view. Once the image is visible, the module initializes the tracker and the relevant location and orientation of the mobile are estimated. The tracking continues for each frame as the device is moved, for as long as the target is still detected in the video stream. This information is transported to the 3D rendering module, which uses them as an input to calculate the transformation needed. The 3D model is transformed and thus aligned with the target image.



**Figure 22: LecceAR – Screenshots off the app [4] –
The target as framed by the device (left) – The 3D model superimposed on the target image (right)**

The main element that makes the specific application distinguish from others is its flexibility and capability to render large and complex 3D models. The renderer, which is developed in OpenGL for the iOS platform, can be encapsulated inside a UIView, the basic iOS widget for visualizing graphics on a screen. This makes it very handy and increases the flexibility of the system in coping with different rendering and shading needs.

Another important attribute is that despite the absence of artificial markers, the alignment between the physical and the virtual world continues even when the target image becomes only partially visible. For the initialization of the tracking procedure the whole target must be in the field of view, because the module needs it to compute the geometric transformation. After it is computed, though, the planar form allows to the procedure to be continued, something that wouldn't be possible in a 3D Natural Feature approach.

2.4.9. The MIRACLE project

MIRACLE [19] is a two-year research project (2015-2016) with a vision to find cost-efficient ways to create attractive Mixed Reality Applications for Culture and Learning Experiences. A number of pilot cases have been tested through demonstrator applications and their results are presented in a handbook. A toolkit has also been created and contains elements based to the experience gathered [102].

Some of the most relevant to the Cultural field cases of the project are presented below.

2.4.9.1. Luostarinmaki Adventure

With the Luostarinmaki Adventure, which takes place in the Luostarinmaki Handicrafts Museum in Turku of south-western Finland, the user experiences interaction with digitally created characters who are supposed to have lived in the area in the middle of 1850's. The application is constructed as a game, the events of which are happening during the summer of 1855. The player is supposed to be a 23-year-old man from the countryside, who is unfamiliar with many things about the area. The character has some tasks to fulfill, and for that purpose he wanders around exploring the city. There he meets people, who must explain to him important facts about their everyday life. As the city of the game is actually the museum and the inhabitants are digital characters that the user can see through a hand-held device, a tablet, the game works as a window to history.

A markerless point cloud tracking has been used for the creation of the experience. To achieve this approach a virtual representation of the environment has been created from a large number of photographs and has been used as the tracking target. For the visualization of the characters the designers used the Unity Multipurpose Avatar plugin and the clothes were modelled to them with the Blender software. The dialogues between the player and the characters appear on the screen in text form. The user asks questions by tapping one of the alternative sentences and they answer with lines of speech drawn over their heads.

The evaluation which took part in parallel with the application development with questionnaires to the users indicated that the experience was pleasant for the majority of them. There were some technical problems, like the unstable tracking and the weight of the tablet that tired some visitors. The most important issue, though, was that the immersion in the game was too much for some of the users and as a result they were not paying attention to the surrounding museum.

2.4.9.2. Wordsmith

Another storytelling experience is offered through the Wordsmith application. The scenario takes place inside the Turku Cathedral and presents the changes that happened in the local life from 1514, when the Protestant Reformation started, and during the next 150 years, from a viewpoint of a fictional family. Several objects and historical characters are digitally attending the church, visible on a screen. The application has taken its name from one of the heroes of the story.

The framework is similar with the one designed for the Luostarinmaki Adventure. Markerless tracking technology based on point clouds are once more used to avoid visible markers and a tablet device is the display through which the digital elements are presented. The app can also be downloaded to Android or iOS personal tablets. A difference is that a secondary tracking system, which consists of Bluetooth beacons discreetly placed in objects like lamps, is imported. The beacon signals are playing the role that the GPS plays in outdoor applications, which is the approximate location of the user. Once it is calculated, the app displays an image guiding him to the nearest scene.

For the visualizations of the characters several tools are used by a graphical designer. Furthermore, a historian has made their clothes in MarvelousDesigner, a clothing design software. To achieve realistic movements, the scenarios were played by actors and recorded with special capture equipment. The app also offers the possibility to learn more information about the characters and the historical events. Some of the supplementary information is included, while web pages links can lead to more extended descriptions.

2.4.9.3. House Casagrande

This application is the first from the list which includes the concept of diminished reality, which means to virtually remove some real objects from the scene. In this case, the residential quarter that is now occupied by the House Casagrande in Turku has to be presented in former periods of time, which requires some of the existing features to be removed, as they occupy the space where the digital models have to be superimposed.

An architect created the historical buildings' 3D models as they were in different periods between 1588 and 1790, with guidance from archaeologists and historians who were based on descriptions from archaeological sources. The models were afterwards manually simplified, removing details that would not be visible through the augmentation and adding textures to make them photorealistic.

The next step was to decide which diminishing method should be used. There are 3 methods to remove existing objects. The most invasive one is the *clean plate compositing*, that would model most of the surroundings, creating almost the same result as virtual reality would do. The second one, called *diminished*, would leave some of the surroundings, using a complex process. The project is finally realized with the last one, *distorting and colorizing the area*, which uses a mist-like effect to partially hide the buildings. Its advantage is its simplicity and that it matches the real light and weather conditions.

3. Producing an AR application

To travel in time and space is the dream of everyone who is interested in history and cultural heritage. Nowadays, thanks to the technology revolution that brought internet to our everyday lives, it is easier than ever to have access to information which concerns places far away or of another era. Interacting with them, though, is not equally easy. The wish of not only having access to relevant knowledge, but also being part of experiences that have been created based on it, is a dream that can get real with Augmented Reality.

For the current chapter, an attempt to produce such an experience for a monument situated to a remote location has been realized. The idea was to create an application that would superimpose a computer-generated 3D model of this monument's current situation over a target image. As the image to be used as base is the monument's floor plan, the application's idea is similar to that of LecceAR project, mentioned in the second chapter. The main difference is that instead of the iOS, the Android platform has been selected. The monument to be presented is the Tomb of Christ, situated in the Church of the Holy Sepulchre in Jerusalem.

3.1. The Church of the Holy Sepulchre

Over the years a lot of places have been believed to be the location where Jesus was temporarily entombed for the three days before His Resurrection. The most widespread aspect, supported by the Catholic Church, Eastern Orthodox churches and Oriental Orthodox churches, proposes The Church of the Holy Sepulchre as the one that contains Jesus's empty tomb [110]. This church, otherwise called Church of the Anastasis, is located in the Christian Quarter of the Old City of Jerusalem. Apart from the tomb, it also contains the site where Jesus of Nazareth was crucified, according to traditions dating back to at least the fourth century [72]. The Status Quo of the Holy Land Sites, a 150-year-old understanding between religious communities in Jerusalem and Bethlehem, applies to the site, which is also a part of the UNESCO World Heritage Site Old City of Jerusalem.

3.1.1. History

The building's initial form was that of a temple, constructed by the Roman Emperor Hadrian in the second century AD and dedicated to the goddess Venus. Its replacement to a church realized by the first Christian Emperor, Constantine the Great, in about 325 to 326. It is believed that it was his mother, Helena, who discovered the tomb, as well as the three crosses on which Jesus and the two thieves were crucified [70]. The control of Jerusalem and thereby the Church of the Holy Sepulchre has changed hands several times throughout its life, fact that led to a lot of interventions.

In 614 the Church was damaged by fire when the Persians invaded Jerusalem and captured the Holy Cross. It was restored by the Emperor Heraclius in 630 but surrendered to Muslim control in 638. During the next centuries it was damaged because of two earthquakes, repaired by Patriarch Thomas in 810, yet damaged again because of new fires that occurred in 841, 938 and 966 and burned the inside of the basilica. During these years it continued to function as a Christian church, something that changed in October 18, 1009, when a campaign against Christian places of worship in Palestine and Egypt almost totally destroyed it, leaving only few parts of the early church remaining. Christian Europe's intense reactions led to an agreement that allowed the rebuilding and redecoration of the Church,

which was completed by Emperor Constantine IX Monomachos and Patriarch Nicephorus of Constantinople in 1048. The great basilica remained in ruins, as a total replacement was far beyond available resources.

The fate of Jerusalem was a main concern of Pope Urban II, who called for the First Crusade in 1099. The crusaders refurbished the church into the current Romanesque style during the reign of Queen Melisende in 1149 and they made it the seat of the first Latin Patriarchs and the site of the kingdom's scriptorium. The first Sultan of Egypt and Syria, Saladin conquered the city in 1187, allowing though for Christian pilgrims to visit the site. Both the city and the church were regained by the Roman Catholic Church in 1229 and came to Orthodox hands in 1244, when the Crusades finished.

Subsequent centuries were not altogether kind to the Church of the Holy Sepulchre, which suffered from damage, desecration, and neglect. A further renovation occurred in 1555 by the Franciscans, who also demanded control of the church. It is believed that this is when the burial bed has been covered in a marble plaque. A new fire in 1808 caused the dome of the Rotunda to collapse and smashed the Aedicule's exterior decoration, leading to new restorations in 1809 by architect Nikolaos Ch. Komnenos of Mytilene. Not until 1959 did the three major communities - Latins, Greeks and Armenians - agree on a major renovation plan. The guiding principle was that only elements incapable of fulfilling their structural function would be replaced [71] [119].

In October 2016, the Edicule that covers the tomb underwent a significant restoration by an interdisciplinary team from the National Technical University of Athens that also aimed to its geometric, structural and architectural documentation [33]. The tomb was opened for the first time in centuries. Apart from the marble cladding that covered the burial bed, another marble slab with a cross carved into its surface was exposed. This indicates that the original limestone burial bed is intact, which suggests that its location has not changed through time.

3.1.2. Description and Architecture

The church's chaotic history is evident in its current appearance. It is hard to classify it as a specific architectural style, as Byzantine, medieval, Crusader, and modern elements mix in an odd mish-mash of styles and each governing Christian community has decorated its shrines in its own distinctive way. In the courtyard, which is known as parvis, except for the church, there are also a few smaller structures, like Orthodox, Roman-Catholic and Coptic monasteries and chapels, broken columns and the tomb of a royal counselor [71].

The original construction included two connected churches over the two different holy sites, the tomb and the site of Golgotha. It included a great basilica, an enclosed colonnaded atrium and a rotunda. Five small chapels were attached to the east of the court of resurrection at the rebuilding that took place in 1048. A cistern situated under the former basilica, where Helena had found the Holy Cross, is also believed to be discovered in this period; it later became the "Chapel of the Discovery of the Cross". By the crusader period a new chapel dedicated to Helen was constructed in a space that used to be part of Hadrian's temple, which was discovered while attempting to reach the cistern. The original excavation tunnel was transformed into a staircase that is used to descend to the "Chapel of Saint Helen". That was also when the bell tower was added and the church took its Romanesque style. When the Franciscans rebuilt the Aedicule that covers the tomb in 1555, an ante-chamber, known as the "Chapel of the Angel", was created. The Ottoman-Baroque style was added by architect Nikolaos Ch. Komnenos of Mytilene, who rebuilt the Rotunda and the Aedicule's exterior after the fire of 1808. He also changed the interior of the ante-chamber to a square ground-plan, in place of the previously semi-circular western and applied a cladding of red marble to the Aedicule. The current dome dates from 1870.

In figure 23, a floor plan of the Church is presented. As it can be seen, when entering from the parvis, the first element to encounter is the Stone of Anointing or Stone of Unction. The present stone was only added in the 1810 reconstruction, it is believed however to be the spot where Jesus's body was prepared for the burial. A stairway to the right leads to Calvary (Golgotha), considered as the site of Jesus's crucifixion, where the apex of the Rock of Calvary is contained. The Chapel of the Calvary is split into two halves, a Greek Orthodox and a Catholic, each one with its own altar. Beneath the main altar, the one in the northern half, there is a hole in the rock, supposed to be the place where the cross was raised, fact that makes the Calvary the most visited site in the Church, along with the Tomb of Jesus.

The main altar of the Church is housed in the Greek Orthodox Katholikon, originally structured by the Crusaders. The compass, an omphalos that was once believed to be the center of the world, is also kept there, under a second smaller dome. This dome is topped by the monumental Golgotha Crucifix, a symbol of the efforts of unity in the community of Christian Faith. At the rear of the Katholikon is the iconostasis, divided into two by a patterned series of red marble arches and columns, behind which are located the traditional Greek Orthodox icons. On either side of the iconostasis the two Patriarchal thrones are set, reserved for visits of the Orthodox Patriarchs of Antioch and Jerusalem.

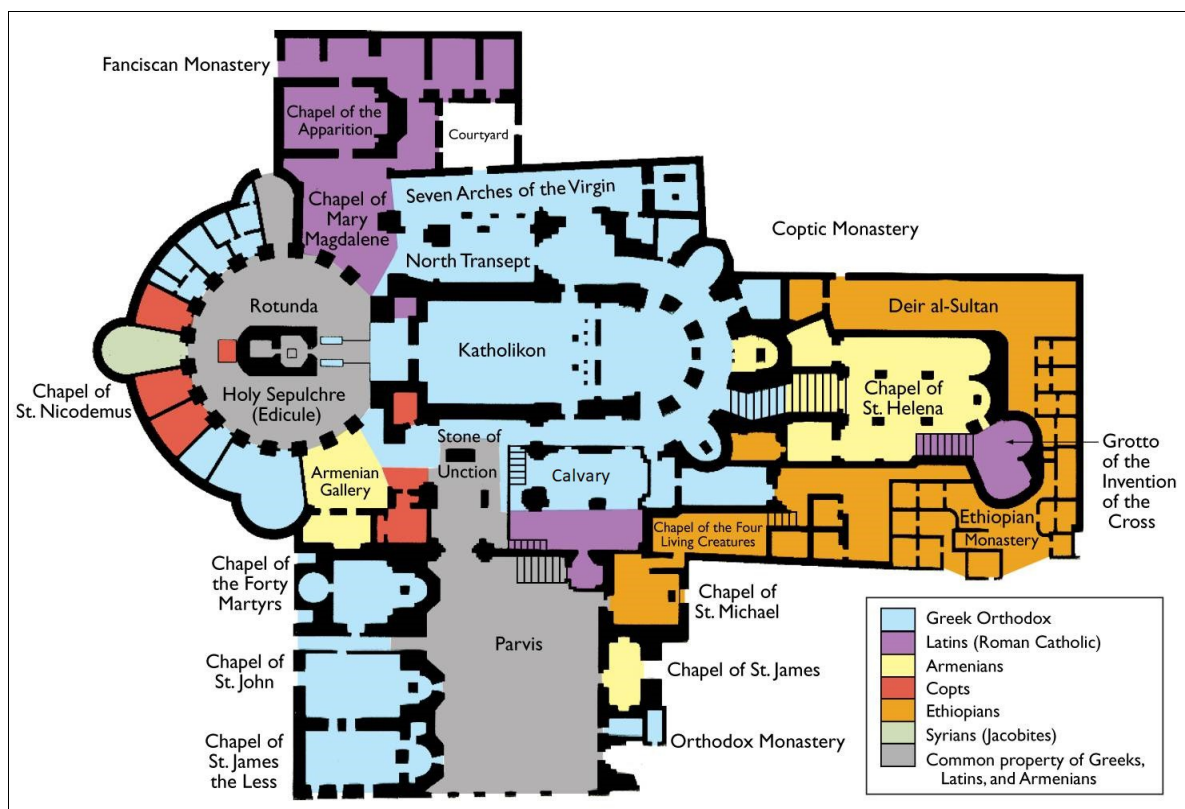


Figure 23: Floor Plan of the Church of the Holy Sepulchre [66]

The space identified as the burial site of Jesus is contained inside the Rotunda, called the Anastasis, which means Resurrection in Greek. The Rotunda's dome is one of the most prominent features of the Church, a soaring structure that has been a part of its several versions, including the one that stands today. Though simply decorated, the dome is dramatic, with warm, golden rays that stream out from its center oculus.

The tomb is located on the west side of the Katholikon, nowadays enclosed in a small shrine, called the Aedicule in Latin or Kouvouklion in Greek, situated in the center of the Rotunda. The original Sepulchre had the Jewish burial carved stone monument form, which means that it was an uncovered monument with an octagonal roof carved in rock. It only took its present form during the restoration of 1810 [100]. The Aedicule is currently covered by a flat roof with a small Russian-style dome covering its west side. The double curvature lead cupola is supported by a parapet wall with pillars and pillar supports, while it rests on a high arched rim. The side panels are decorated with inscriptions in Greek inviting people and nations to praise the Risen Christ.

The only entrance to the holy Aedicule is on the east side, facing the Katholikon. The facade is luxuriously decorated with many elements, as four monolithic Corinthian style columns made by reddish-white marble, a silver/gold plated relief of Christ's resurrection and a western-style oil-painting of the same theme, rows of hanging oil lamps, a design of a royal crown held by two angelic forms and a lot of inscriptions in Greek, dedicated to Christ. There are twelve candlesticks that encircle the site, six large and six small ones. They belong equally to the Orthodox, Latin and Armenians. As for the rest of the elements, the majority of them were offered by the Orthodox. The later additions offered by the Latin and Armenians occupy a large part of the surface of the Greek inscriptions. The entrance's original door, which was made by wood and was decorated with mother-of-pearl, has been replaced since the fire of 1808 and it is currently kept to the museum of the Patriarchate [100].

The Aedicule is separated in two rooms; the first one contains the Angel's Stone, which is believed to be a fragment of the large stone that sealed the tomb, while the second one is the Holy Tomb itself [71]. The tomb is covered by a marble cladding to prevent pilgrims from removing bits of the original rock as souvenirs. They can see it, however, as there are several thick window wells extending through the marble sheath. They reveal an underlying limestone rock, which may be part of the original living rock of the tomb.

Attached to the rear of the shrine is the Chapel of the Copts. A portion of the bed of rock from which the tomb of Christ's burial was excavated is exposed inside, situated beneath the altar [100].

Jerusalem's role in first century Christianity gives it a significant role for Christians, especially during the Holy Week, when the main facts concerning the Salvation occurred in the city: Jesus's arrest in Gethsemane, His trial, His crucifixion at Golgotha, His burial nearby and His resurrection and ascension and prophecy to return. The Church of the Holy Sepulchre is directly tied to these facts, which makes it one of the most important shrines in Christianity. It has been a special place of pilgrimage since it was founded.

Every year on Great Saturday, the day preceding Orthodox Easter, a miracle supposedly occurs at the Church; a blue light emanates within Jesus Christ's tomb, eventually forming the Holy Fire, from which candles of the pilgrims in attendance are lighted. According to the pilgrims, this Fire does not burn them. It is also said to spontaneously light other lamps and candles around the Church. There are even special flights, being received by church and state leaders, to take the Holy Fire to certain Orthodox countries, such as Greece [117].

3.2. Preparation of the 3D Model

In order to include the model in the application, it not only has to be created, but also to be amended for the code to be able to support it. The next chapters are about the Tomb's digital reconstruction and the techniques that have been used to reduce its size.

3.2.1. Reconstructed 3D Model of The Tomb of Christ

Each part of the 3d model was produced as a unit by the interdisciplinary team from the National Technical University of Athens that restored the Aedicule, as part of its geometric, structural and architectural documentation. Their construction realized through an automated photogrammetric image-based method. 30–40 laser scans and many high-resolution photographs have been used for this task.

In figure 24, the composition of these parts to a complete 3D model is presented from its five aspects. The composition has been made independently from the their production, using the Autodesk Meshmixer toolset.

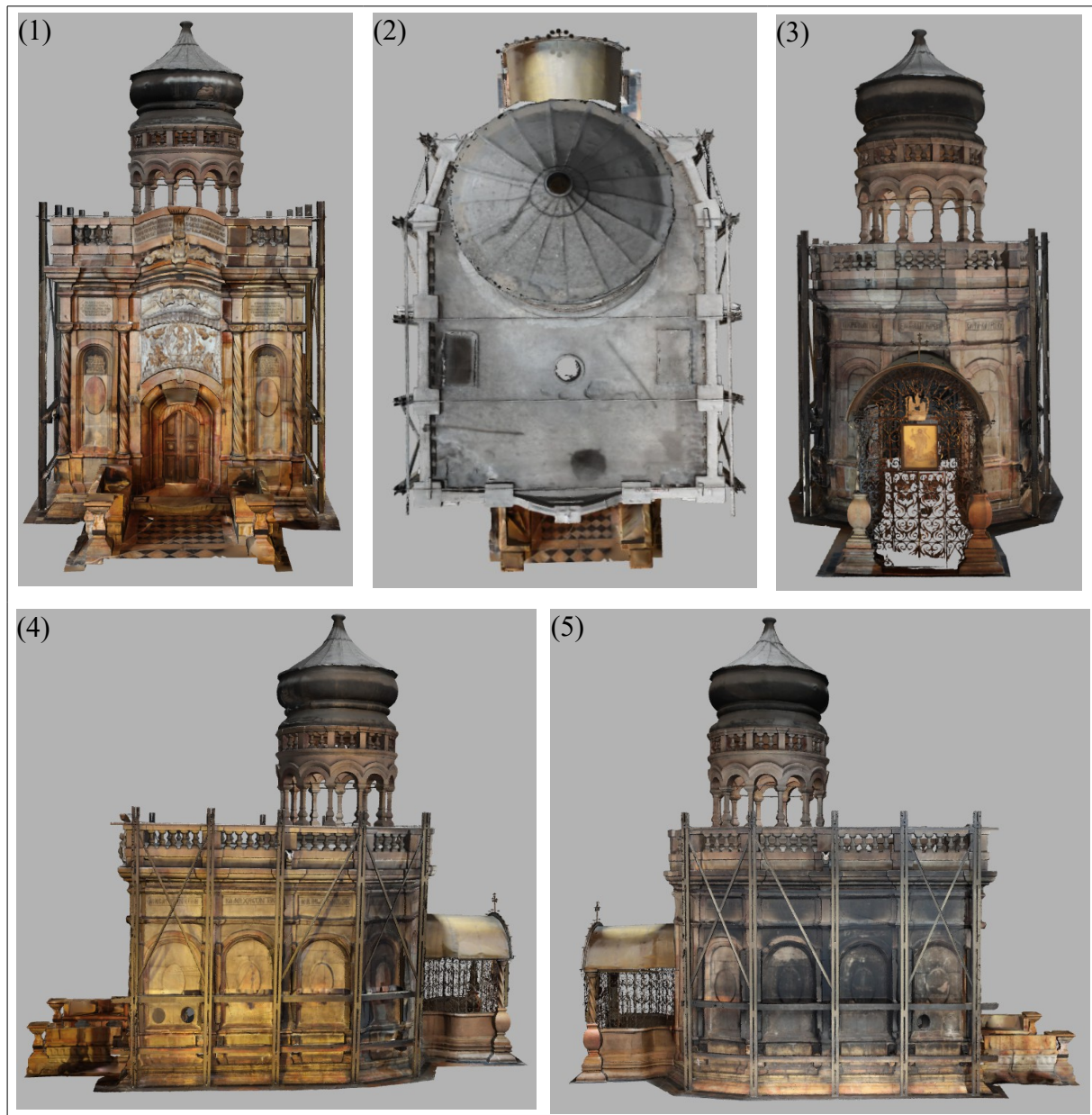


Figure 24: Complete 3D model of the Aedicule
(1) East side – (2) Seen from above – (3) West side – (4) North side – (5) South side

The 3D model of both the internal and the external parts of the Holy Edicule consists of 18010861 vertices and 35349698 triangles and the file's size is 3.24 GB. It is obvious that it is not possible for such a huge file to be imported in an Augmented Reality application, especially to one that is going to use a smartphone's processors and memory. Thankfully, it is not necessary for the application to incorporate the internal parts, as only the external ones will be visible anyway.

When only the absolutely necessary elements are maintained, the model consists of 4309424 indices and 8606037 triangles, while the file's size is 609 MB. Although these values are significantly reduced comparing to the initial ones, they still forbid the model's usage for the desirable project. In order to be able to use it, techniques to further reduce its complexity have to be applied.

3.2.2. Reducing 3D Model's Size

Reducing the complexity of a 3D model can be extremely important – whether the number of objects, the polygon count, the texture size, or any combination of these. Smaller models are more efficient and can significantly improve rendering times. Especially when it comes to Augmented Reality applications, where the real time performance is a necessity, the final file must use the minimum detail possible.

What is even more important, however, is to present a realistic result. The changes to the project should not be major, otherwise the appearance of the object may be destroyed. When a reduction of the model's size is taking place, a balance has to be stroked between these two factors; the minimum complexity to achieve the maximum realism has to be found.

One of the first steps in simplifying models is to reduce the number of individual objects. The first thing to consider is if all of the objects will be visible in a presentation rendering. However, in this specific case, all the external objects are visible, while the internal ones have already been removed. The next step is to reduce the polygon count, meaning the number of polygons that are used to represent an object.

Meshmixer, which is a software designed to work with triangle meshes, has a Reduce tool that allows to re-mesh a set of selected faces in a way that they approximate the original shape with a new one, made with a smaller number of triangles. There are three ways of doing it, depending on what the Reduce operation is trying to achieve.

In Percentage mode, the goal is to reduce the triangle count to a fraction of the input triangle count, in Triangle Budget mode a specific triangle count has to be achieved and in max Deviation mode, the goal is to reduce as much as possible while ensuring that the surface does not shift too far from the input surface, where this distance is defined by the user. There are two different Types of Reduce operation. In the Uniform mode, regularly-shaped triangles are achieved, using a shortest-edges-first criterion. In the Shape Preserving mode, which is the software's default, edges which will have the smallest effect on the shape are incrementally collapsed.

Two more tools can affect the procedure's result; the Preserve Boundaries and the Preserve Group Boundaries. Boundary loops are splits in a face that extend to the boundary. When the first tool is activated, it controls whether any boundary loops are allowed to be modified and it pins boundary vertices in place to ensure that they will remain unaffected. The second tool controls whether the border loops of facegroups can be constrained. When activated, the vertices on facegroup boundaries can only slide along their respective boundary edge loops.

To select the method that combines minimum complexity with maximum realism and gives the best result for the specific model, a lot of experimental attempts have been realized. The reduce operation has been attempted both to each part of the model separately, followed by their composition to one model and directly to the whole model. It has been proved that the best technique is the first one.

The Triangle Budget mode was the one selected for the isolated models' production, using the Shape Preserving mode. Both the Preserve Boundaries and Preserve Group Boundaries tools have been activated. The composed model that has been created in this way consists of 1492241 indices and 2548361 triangles and its size is 351 MB.



Figure 25: Details of the final Model, which was created with Meshmixer's Triangle Budget method

Further reduction has also been tried, using the composed model as a whole, however the results were not adequate. As it can be seen in figure 26, the attempt to reduce the percentage of the model to 75% has significantly diminished the realism, while the same has happened when the maxDeviation mode has been used, setting the distance to 0.02

mm. In the same time, the size reduction was negligible.

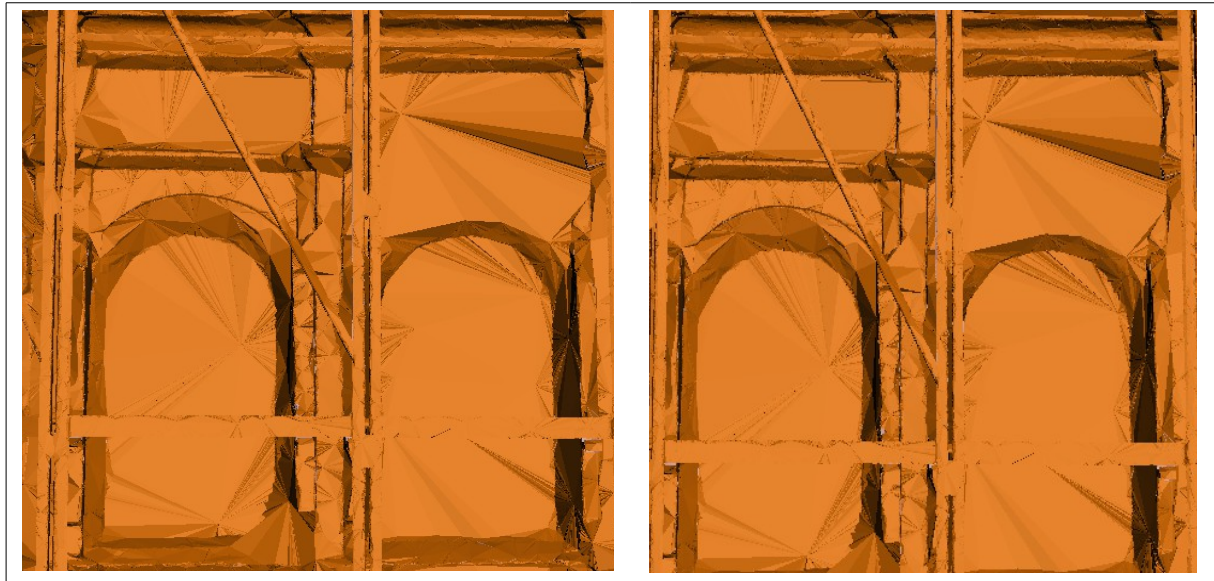


Figure 26: Further size reduction with Percentage mode (left) and Max Deviation Mode (right), which were not kept in the final model.

3.3. Development tools and equipment

The current chapter's aim is to give information about the development equipment that has been used in the procedure of producing the application. The reasons why the specific tools have been selected among the possible choices are also explained here. The last part is about the technical specifications that has to be fulfilled for the realization of the project.

3.3.1. Programming Language

In mathematics, computer science and linguistics, a formal language is a set of strings formed by symbols, letters and tokens, called words, together with a set of rules that are specific to it. The programming languages belong in this category. They specify a set of instructions that can be used to implement specific algorithms.

The application is intended for users with Android phones and tablets. The official language for Android development is Java, while it is also possible to develop C and C++ using the Android Native Development (NDK) kit. These three programming languages have been voted among the most popular ones, with Java coming first in the list for years [103] [121]. For these reasons Java language has been chosen to be used for the current project.

Java is a general-purpose computer programming language, that is concurrent, class-based, object-oriented and specifically designed to have as few implementation dependencies as possible [94].

The term concurrent means that a lot of computations can be executed at the same time, without waiting for other computations to be completed. The parallel execution allows more tasks to be completed in a given time compared to sequential algorithms, where one operation has to be completed for the next to begin. Especially in input-output intensive programs the responsiveness is significantly increased, as the time that would be spent

waiting for the data importation can be used for another task.

A language is considered object-oriented when it not only includes the basic capabilities for an object, like identity, properties and attributes, but it is also characterized by polymorphism and inheritance. Polymorphism offers the capability to overload the name of a function with multiple behaviors based on the objects that are passed to it, while with inheritance it is possible to create extended sub-classes that inherit all the data constraints and behaviors of a parent class.

An object-oriented language is considered class-based when the inheritance is achieved by defining classes of objects instead of directly using the objects. A class is actually a definition of all objects of a specific type and each object is considered an instance of that class. Classes may also contain procedures, known as methods.

Java's popularity with developers, apart from its steadily good performance, is also due to its portability, scalability and large community [94].

Java is a compiled programming language, but rather than compile straight to executable machine code, it compiles to an intermediate binary form, called JVM (Java Virtual Machine) byte code. That means that a source code written in one platform can be executed in any platform without need of recompiling, as declared by the slogan "Write once, run anywhere" (WORA), which was created to illustrate Java's benefits. Internal details of JVM differ from platform to platform but since they all still understand the same Java Bytecode, the portability is not particularly affected.

Scalability is the ability of a project to scale, so as to work equally well with a small and with a large set of data. It is also the potential of a system to be enlarged to accommodate a growing amount of work. Java is built in a way that allows the creation of extended versions of a code, while simultaneously maintaining the initial form. For the same reasons, old versions of the language will still run perfectly even after new versions are released, so that rewriting the code won't be necessary.

When it comes to programming languages, the more their community increases the more popular they become and vice versa. A large community means that developers can easily find help on websites about problems they might encounter and a bigger variety of ready-to-use libraries exists. Furthermore, as the number of users is growing, more and more projects are developed in Java and more and more systems are using it, which ensures its future popularity.

3.3.2. Framework

One of the most important decisions for a project's production is the framework that will be used to create the desirable result. A framework provides functionalities and solutions for a particular problem area, serving as a support or guide for the building. A software framework offers common code that provides generic functionality, which can be overridden or specialized by user code in order to provide specific functionality.

3.3.2.1. Selection's procedure

There is a wide variety of Augmented Reality related frameworks, each one running in specific platforms and offering specific features. For the selection of the most suitable one, reviews indicating the most popular choices were examined. Searching the Augmented Reality Software Development Kits [114] and after consulting internet sites and forums, where opinions concerning the results accomplished with them were given, four development kits appeared as the most appropriate choices, among the existing free ones. Their basic

characteristics at the time of the selection are presented in the following table:

AR Framework	Company	License	Supported Platforms	Basic Tracking Features
Vuforia	Qualcomm	Free and Commercial	Android, iOS, Unity	Marker, Natural Feature, Visual Search
Metaio	Metaio	Free and Commercial	Android, iOS, Windows, Google Glass, EPSON Moverio BT-200, Vuzix M-100, Unity	Marker, Natural Feature, GPS, IMU Sensors, FaceTracking
ARToolKit	DAQRI	OpenSource	Android, iOS, Windows, Linux, Mac OS X, Silicon Graphics International	Marker, Natural Feature
Wikitude	Wikitude	Free and Commercial	Android, iOS, Google Glass, Epson Moverio, Vuzix M-100, OptinVent ORA1, PhoneGap, Titanium, Xamarin	GPS, IMU Sensors, ContentAPI

The existence of the Android platform is necessary, as the application is intended for users with Android phones and tablets. It is also good to have access to Windows and iOS platforms, which gives the possibility of future expansions to Windows phones and iPhones. All these frameworks support Android and iOS platforms, but Metaio and ARToolKit are the only ones to also support Windows platform.

As far as it concerns the features, it is necessary for the framework to include Marker and Natural Feature Tracking, as a combination of these two methods is possible to be needed. These features were included in Vuforia, Metaio and ARToolKit Frameworks, but not in WikiTude. Nowadays, Wikitude has also included vision-based tracking.

Based on these elements, the first choice was Metaio, as it included all the necessary characteristics and it is believed to be one of the best SDKs. However, when the research for documentation began, it turned out that it was not possible to be used, as Apple had acquired Metaio one year earlier and it was not available anymore for any public projects.

The second choice, which also includes all the necessary for the specific application characteristics is ARToolKit and it is the one that is going to be used. An advantage that was considered important at the time of the selection was that apart from the supported platforms mentioned above, ARToolKIT also includes a plugin for the Unity game engine, which supports Unity on AS X, Windows, Android and iOS and a plugin for OpenScene Graph. These plugins could make the 3D-model's construction and usage much easier.

3.3.2.2. ARToolKit – Information and features

ARToolKit is an open-source computer tracking library for creation of strong Augmented Reality applications that overlay virtual imagery on the real world [59]. It was originally developed by Hirokazu Kato of Nara Institute of Science and Technology and Mark

Billinghurst, in 1999 and released by the University of Washington HIT Lab (Human Interface Technology Laboratory). Later it further developed by other HIT Lab scientists. In 2001 its first open-source version was released through the HIT Lab.

This framework is one of the first AR SDKs for mobile, seen running first on Symbian in 2005, then on iOS in 2008 and finally on Android in 2010. Currently, it is maintained as an open-source project hosted on Gitub [61]. It was acquired by DAQRI and re-released open-source, starting in 2015, including all the features that were previously available only in the professional licensed version.

What this framework actually does, is that it furnishes predefined functions which can be called in a specific order for developing an AR project. There is also the possibility to use different parts separately. Using OpenGL for the rendering part, GLUT for the event handler aspect and a hardware-dependent video library, it attends to minimize libraries dependencies without sacrificing efficiency.

As long as it concerns the framework's features, it is important that it is fast enough for real time AR applications and it is optimized for mobile devices. It also offers a variety of methods to be used [60].

One of the most difficult parts of developing an AR application is precisely calculating the user's viewpoint in real time. When ARToolKit was first created only marker-based tracking was possible, as the algorithms used for Natural Feature Tracking are considered complicated even with the current technology. As the computational power of devices improved, the platform started introducing this type of tracking too. Nowadays it also offers full marker-less tracking. For the marker tracking, it gives the possibility to choose between the usage of classical square markers, 2D barcode markers and multimarker. In ARToolKit the term multimarker has a special meaning. Rather than meaning the use of more than one marker at a time, it refers specifically to the use of multiple square markers fixed to a single object. This way, even when one marker is obscured, another may be visible, so the system is still able to estimate the pose. A combination of the above tracking systems is also a choice with this kit. Additionally, it is possible to use stereo tracking, if more than one cameras are used.

To get best tracking accuracy it is important to calibrate the camera. In the ARToolKit software there is a camera parameter file with default camera properties, sufficient for basic tracking for a wide range of different cameras, while documentation is also offered explaining the way to calibrate the camera independently.

Both video and optical see-through technologies are able to be used, although a Head-Mounted Device would be essential to use the second one.

3.3.2.3. ARToolKit – Requirements

ARToolKit has been ported to various platforms, including the Android mobile one. Android is an open-source software stack for mobile devices, that provides the operating system, as well as middleware, key applications and APIs. For application development, Android offers SDK and development tools, based around the Java programming language. To be able to program in Java, there are additional SDKs and tools that have to be installed.

The most important one is the Android Studio Integrated Development Environment (IDE), which is the official IDE for Android Platform development. It was announced in 2013 and it is fully available under the Apache License 2.0. It uses the Gradle build automation system, which build upon the concepts of Apache Ant and Apache Maven build automation tools. The Android Software Development Kit is also downloaded with the Android studio IDE.

The Standard Edition Java Development Kit (JDK), which includes tools for

developing, debugging and monitoring Java applications, is the next to be installed. It is released by Oracle Corporation, under the GNU General Public License (GPL), in 2006.

Java is sufficient for the majority of applications developed with ARToolKit. Nevertheless, there is the possibility to implement parts of a Java application in C or C++, using the Java Native Interface (JNI) framework. Using native code offers the possibility to increase an application's performance, while it provides access at existing C and C++ libraries. Both of these are good motivations for including native coding in an application. In order to do it, there are two more requirements:

The Native Development Kit (NDK), which provides the necessary tools headers and libraries to take advantage of JNI on Android.

The Git distributed version control system, which is used for tracking changes in computer files and coordinating work on these files and the Git BUSH (Bourne Again Shell), which is a text-only command interface that has features which allow automated scripts to be run.

3.3.3. System's technical specifications

From the information given above it is easy to understand that the system's hardware has to be strong enough to support an Augmented Reality project. This chapter is about the components of both the computer and the mobile device that are used for the current application.

The smartphones are designed to keep all their critical components in a relatively small area on the device, called system-on-a-chip or SoC, in order to reduce the size of the component board inside and to help make the device faster and more battery efficient. The SoC includes the processor cores, the graphic cores, the RAM and possibly ROM and more. The three first elements mentioned are the most important ones when it comes to an AR app [113]. Needless to say, two more components are essential; the display for the optical visualization and a camera, in this case operating both as a sensor for the tracking and as a mean of viewing the real-world scene.

The processor is the brain of the smartphone, where all basic calculations are done and the code is executed. The Central Processor Unit (CPU) handles the majority of things on a smartphone, from running the operator system to touch-screen functions. Same with an AR app, CPU is responsible of running it, meaning that it ordiates all the other components, of calculating the transformations needed and of reacting to modifications made by the user. Processors are named according to their numbers of complete independent units known as cores. In theory, the higher the number of cores running in parallel is, the faster the performance and task handling will be. The speed of each one of the cores, which can be counted in Hz, matters too. In practice there are also other factors to define the performance, as the approaches in the creation of each processor [85].

The Graphic Processing Unit (GPU) is used to handle more complex tasks, where the central one would fail, such as the rendering of graphical and visual data like Web pages, gameplay, or, in the specific case, 3D models and high-resolution videos. In the event of a GPU absence, the CPU could handle these tasks too, however not equally efficiently and not with equally smooth results. This technique was used at the original low-end Android devices, but it was afterwards corrected. At the present time systems, CPU will usually only help out of certain calculations in this sector. For an application like the one designed in this case it would not be an exaggeration to claim that GPU plays the most important role, as the rendering is the critical part for a realistic presentation. The speed of the graphic processors

is also counted in Hz [86].

RAM is one of the critical components of the smartphone along with the processing cores and dedicated graphics. It is where critical files needed by the processors, such as application data, game graphics or operating system components, are stored, waiting to be accessed. In an AR app that would mean information to help the tracking procedure, 3D models and the image to be tracked, among other data. Smartphones use Dynamic RAM, DRAM, the contents of which can be changed quickly to cope with whatever tasks the system is trying to perform. When the system is powered off, the contents of the RAM are lost and they are filled once more at the next boot. ROM is a different type of memory, a Read-Only one as its full name is, which means that the data stored in it are either modified slowly and with extreme difficulty or not at all. For this reason, it is mostly used to store firmware, a type of software that is closely tied to specific hardware and unlikely to need frequent updates, or application software. Unlike RAM, ROM retains its contents even when the computer is turned off. Erasing this memory would damage the system permanently. Apart from that, ROM does not play a specific role in the construction of an AR application. Both the size and the speed are important when it comes to RAM, as having large amounts of memory improves the performance, but only when it can be accessed quickly. The size is counted in Bytes while the speed is once more counted in Hz [87].

Two main types of smartphone displays are utilized in the present time; Liquid Crystal Displays (LCDs) and Active-Matrix Organic Light-Emitting Diodes (AMOLEDs). LCDs offer accurate color reproduction and can be bright and clearly readable, but they are problematic when it comes to viewing angles and high contrast. AMOLEDs, on the contrary, have excellent viewing angles, high contrast and vibrant colors, but the color reproduction is not accurate, as mild color tinting is often present. No matter which of these are used, the size and the resolution are crucial for the final result, as they define the ppi density, meaning the number of pixels that fit in one inch. Individual pixels should not be distinguished at a reasonable distance from the user's eyes in order for the images and the text to be presented at high quality [88].

Last but not least, the type of the camera is very important for AR applications. The quality of a digital camera is determined by many factors, as the resolution, the lens aperture, which also depends on the focal length, their zoom range and quality and even the software used. The number of frames per second (fps) is also important for video recording. As a matter of fact, when it comes to AR apps the number of fps is one of the critical things to examine, considering of course that all the other elements are usually satisfactory enough in modern smartphones [89]. In general, the resolution is the criterion mostly used for a smartphone's camera selection, something that does not examine all the factors as it should.

An average smartphone in use is currently equipped with an at least Quad-core 1.2 GHz CPU, although the ones launched during the last year contained Hexa-core or Octa-core ones with speed more than 2 GHz. The graphic core speed is usually between 500 and 700 MHz and the storage capacity is considered good at 64 GB for ROM and at least 2 GB for RAM. 300 ppi is an adequate display density. The mobile device where the application is supposed to run is the Samsung Galaxy A5 (2015), which fulfils the minimum requirements. It contains a Quad-core 1.2 GHz CPU, an Adreno 306 GPU, the core speed of which is 450 MHz, 64 GB ROM and 2 GB RAM capacity and 294 ppi density. Its frame rate is at 30 fps, an appropriate value for AR app production [111].

While the smartphone must have specific features to allow the function of the application, the computer has to be able to also support its building, which requires a large number of tools to be downloaded and installed. To achieve it, a strong system is required. The components that define a desktop system's strength are the motherboard, the CPU, the graphic card, the RAM and the hard drive.

A motherboard is essential for the computer's operation, as it holds its microprocessor chip and lets everything else connect to it. It has CPU and memory sockets attached on it, providing many functions including support for a heat sink and easy and economical replacement. The electrical interface that is formed holds and allows communication between not only the central processing unit (CPU) and the memory, but also many of the other electronic components of a system, as connectors for other peripherals are also provided on the printed circuit board (PCB) [104].

The usage of the CPU is basically the same as for the mobile phones. It performs all the basic arithmetic, logical, control and input/output (I/O) operations specified by instructions, the sequence of which is called a program. The higher the speed of the processor, the faster the computer will be able to handle multiple programs and activities. At this point in time their speed is about 300 Mhz, which means that the computer can execute 100 million instructions in one second [76].

The graphic card uses a GPU as its brain, just like the motherboard uses a CPU. This programmable logic chip is specialized for display functions, rendering images, animations and video for the computer's screen. Its highly parallel structure makes it more efficient than general-purpose CPUs at manipulating computer graphics and image processing, just like it happens for the smartphones. The card can either be integrated or dedicated, depending on whether it uses the system's memory or has an independent source of memory for graphics processing. The first one is generally cheaper but less capable, as it can only use 1-5% of the computer's RAM for its processing [84].

The hard drive and the RAM have exactly the same purpose as the ROM and RAM in the mobile devices. Here too, the contents of RAM are always erased when the computer shuts down, unlike the ones of the hard drive, which can be powered down and then back on without losing its data.

The amount of memory and computational speed a computer needs is significantly differentiated accordingly to the way the device is used. The programs that it has to run for the production of an AR application present a similar performance of those used for heavy gaming, which also needs large 3D models rendering and high-speed computations. Some of the programs that were installed for this task even had recommendations concerning the size and velocity of specific components. The system has been built with all the above elements in mind, ensuring its efficiency. Its basic components are the intel core i7-7700 CPU (3.60 GHz), the MSI Radeon RX 470 8GB (RX 470 GAMING X 8G) graphic card, Samsung MZVLW256HEHP PM961 256GB Internal SSD hard drive and the Corsair Vengeance LPX 8GB DDR4-2400MHz RAM.

3.4. Using native code

As already mentioned, ARToolkit includes parts of its code in C or C++ using the JNI naming scheme. The Java Native Interface (JNI) is a standard programming framework that enables Java code running in a Java Virtual Machine (JVM) to call and be called by native applications and libraries written in C and C++. Thanks to this tool, a native method can create Java objects and then inspect and use these objects to perform its tasks. A native method can also inspect and use objects created by Java application code.

In the JNI framework, native functions are implemented in separate .c or .cpp files. When the JVM invokes the function, it passes a JNIEnv pointer, a jobject pointer, and any Java arguments declared by the Java method. The first pointer includes all the functions necessary to work with Java objects, allowing the conversion of arrays and strings from Java to native and vice versa. The second pointer is a reference to the Java object inside which this native method has been declared [95].

To permit the development with native code, the ARToolkit SDK includes components in both C/C++ and Java. The first component, ARToolkitWrapper, is a C++ wrapper, meaning a native shared library that provides high level access to ARToolkit functions and marker management. The second, ARBaseLib, is a Java Android library that communicates with ARToolkitWrapper. The coordination between these corresponding libraries enables access to C/C++ code, increasing the application's performance.

3.5. Project creation

Producing an application can be proved really handy, especially for someone that has not relevant experience in the past. There are a lot of things to take into consideration and too many new information to become familiar with.

The first step was to read the ARToolKIT documentation in order to understand how the system works. To use all the aforementioned tools, it was also necessary to follow tutorials in order to learn the basic functions for Java programming and the way Android Studio can be used for an application's development.

The next step was to install all the required tools and to configure their installation, using the Hello-JNI Tutorial, as it was proposed by the ARToolkit's documentation page. Afterwards, a lot of different environment variables had to be set to the computer, to indicate the paths to specific folders. As it was decided the new application to be created by modifying parts of an existing code, the next thing to do was to build the native dependencies for the example projects provided.

3.5.1. Building the application

For the creation of the application, different parts of the provided examples have been combined. The biggest part of the new code written concerns changes with an aim to adjust the result for the devices and the platforms utilized and to update certain problematic sections that influenced the functionality and the performance.

The system is developed in 6 steps. At the beginning the camera opens and the video recording is initialized. Following, a video input frame is grabbed. If the image to be tracked is viewed by the camera, it is detected, otherwise the next video input frame is examined. When the detection of the image is successful, the relevant frame is transported to the processor, where the camera transformation is calculated. Afterwards, the 3D model is drawn in the right pose and it appears on the device's screen. The final step is the closure of the application when the user selects it.

Following, the code for these steps will be analyzed. It is not possible to provide and explain the whole code, as it would demand too much space and time, while at the same time it would be an exaggeration, but it is important to include some basic parts of each one of the steps mentioned above, in order to indicate how the system works.

➤ Opening the camera

```
try {  
    camera = Camera.open();  
}  
} catch (RuntimeException exception) {  
    Log.e(TAG, "Cannot open camera. It may be in use by another process.");  
}
```

The `Camera.open()` method is obviously responsible of opening the camera. In order to predict and be able to handle possible errors, though, this method alone is not sufficient. `RuntimeException` is the parent class in all exceptions of the Java programming language that are expected to crash or break down the program or application when they occur. In this case, if such an exception occurs, the `log.e()` method sends to the user the error log message that the camera cannot be opened, as long as the possible cause of the error.

➤ Initializing video recording

The initialization of the video recording is an example of the JNI framework usage. First of all, the `nativeStart()` method is called inside the application's main activity.

```
public static native boolean nativeStart();
```

This line points to a `.cpp` file, where the corresponding C++ member function is located. Member function of a class is actually how a method is called in C++, although the name method is still used by programmers coming from other languages like Java. The pointers `JNIEnv` and `object` that have been described in the chapter where the JNI usage was explained (3.4), can be seen here.

```
JNIEXPORT jboolean JNICALL JNI_FUNCTION_NATIVE
(nativeStart(JNIEnv* env, jobject object))
{
    gVid = ar2VideoOpen("");
    if (!gVid) {
        LOGE("Error: ar2VideoOpen.\n");
        return (false);
    }
    return (true);
}
```

The `ar2VideoOpen("")` function returns a pointer to an `AR2VideoParamT` structure, an opaque structure which holds information and configuration for the video stream. Its original declaration occurs in a `video.h` file, which is a library that provides multi-platform video input support for `ARToolKit`, abstracting access to hardware video input available on different machines and operating systems. In order to be used, the `video.h` library has to be declared in an include statement at the beginning of the `.cpp` file.

Through the pointer, the function is passed to other `ar2Video*` functions to specify which video stream is being operated upon, which corresponds to the `gVid` variable. In case the video path is not successfully opened, `NULL` will be returned. When this happens, the `if` statement will be accessed and an error will appear to the user's screen.

➤ Catching a video input frame

In this case too, the `.cpp` file is called by the line

```
public static native boolean nativeVideoInit
(int w, int h, int cameraIndex, boolean cameraIsFrontFacing);
```

where `w` and `h` are the camera capture width and height expressed in pixel, used to determine the image's approximate aspect ratio, `cameraIndex` is the camera id number,

provided by the operating system and `camerasFrontFacing` gives the information about whether the camera is front or rear facing. The corresponding code in the `.cpp` file is

```
JNIEXPORT jboolean JNICALL JNIFUNCTION_NATIVE(nativeVideoInit
(JNIEnv* env, jobject object, jint w, jint h, jint cameraIndex,
jboolean camerasFrontFacing))
{
    gPixelFormat = AR_PIXEL_FORMAT_NV21;
    gVideoFrameSize = (sizeof(ARUint8)*(w*h + 2*w/2*h/2));
    gVideoFrame = (ARUint8*) (malloc(gVideoFrameSize));
    if (!gVideoFrame) {
        gVideoFrameSize = 0;
        LOGE("Error allocating frame buffer");
        return false;
    }
    videoWidth = w;
    videoHeight = h;
    gCameraIndex = cameraIndex;
    gCameraIsFrontFacing = camerasFrontFacing;
    LOGI("Video camera %d (%s), %dx%d format %s, %d-byte buffer.", gCameraIndex,
        (gCameraIsFrontFacing ? "front" : "rear"), w, h,
        arUtilGetPixelFormatName(gPixelFormat), gVideoFrameSize);

    ar2VideoSetParami(gVid, AR_VIDEO_PARAM_ANDROID_WIDTH, videoWidth);
    ar2VideoSetParami(gVid, AR_VIDEO_PARAM_ANDROID_HEIGHT, videoHeight);
    ar2VideoSetParami(gVid, AR_VIDEO_PARAM_ANDROID_PIXELFORMAT,
        (int)gPixelFormat);
    ar2VideoSetParami(gVid, AR_VIDEO_PARAM_ANDROID_CAMERA_INDEX, gCameraIndex);
    ar2VideoSetParami(gVid, AR_VIDEO_PARAM_ANDROID_CAMERA_FACE,
        gCameraIsFrontFacing);
    ar2VideoSetParami(gVid, AR_VIDEO_PARAM_ANDROID_INTERNET_STATE,
        gInternetState);

    if (ar2VideoGetCParamAsync(gVid, nativeVideoGetCparamCallback, NULL) < 0) {
        LOGE("Error getting cparam.\n");
        nativeVideoGetCparamCallback(NULL, NULL);
    }
    return (true);
}
```

This part of the code is not only used to catch a frame, but also to allocate the amount of memory this frame demands. The size of the frame is calculated and given to the `gVideoFrameSize` variable in order to be used as an argument by the `malloc` function, which returns a pointer to the lowest byte in the allocated memory block that is suitably aligned for the object type. The cast `(ARUint8*)` says that the memory ensured will be returned in a `(ARUint8*)` type, meaning an unsigned char type. The `gVideoFrame`, where the final value is assigned, is a buffer that contains the current video frame. If the size returned is zero, the `gVideoFrameSize` will also take the zero value and an error will appear to the user, thanks to the if statement.

The rest of the block's purpose is to tell `arVideo` which the parameters of the camera as the width, the height, the pixel format etc will be for the specific frame, which is assigned to the `gVid` variable. If the synchronization of the parameters fails, an error appears again. In this case, the parameters of the `nativeVideoGetCparamCallback` function become null and the system uses the default values.

➤ Natural Feature Tracking and transformation code

Once again, native code is used. The line

```
public static native void nativeVideoFrame(byte[] image);
```

calls the .cpp file and the corresponding C++ code is

```
JNIEXPORT void JNICALL JNICALL_NATIVE(nativeVideoFrame(JNIEnv* env, jobject
obj, jbyteArray pinArray))
{
    int i, j, k;
    jbyte* inArray;

    if (!videoInited) {
        return;
    }
    if (!gARViewInited) {
        return;
    }
    if (!nftDataLoaded) {
        nftDataLoaded = true;
        threadWaitQuit(nftDataLoadingThreadHandle);
        threadFree(&nftDataLoadingThreadHandle);
    }
    env->GetByteArrayRegion(pinArray, 0, gVideoFrameSize,
        (jbyte *)gVideoFrame);
    videoFrameNeedsPixelBufferDataUpload = true;

    if (trackingThreadHandle) {
        float err;
        int ret;
        int pageNo;

        if( detectedPage == -2 ) {
            trackingInitStart( trackingThreadHandle, gVideoFrame );
            detectedPage = -1;
        }
        if( detectedPage == -1 ) {
            ret = trackingInitGetResult( trackingThreadHandle, trackingTrans,
                &pageNo);
            if( ret == 1 ) {
                if (pageNo >= 0 && pageNo < surfaceSetCount) {
                    detectedPage = pageNo;
                    ar2SetInitTrans(surfaceSet[detectedPage], trackingTrans);
                } else {
                    detectedPage = -2;
                }
            } else if( ret < 0 ) {
                detectedPage = -2;
            }
        }
        if( detectedPage >= 0 && detectedPage < surfaceSetCount) {
            if( ar2Tracking(ar2Handle, surfaceSet[detectedPage], gVideoFrame,
                trackingTrans, &err) < 0 ) {
                detectedPage = -2;
            }
        }
    }
}
```

```
    }  
    }  
  
    } else {  
        detectedPage = -2;  
    }  
}
```

First of all, the function checks if the video and the ARview have started, as there is no point to track until they are both initialized. Then, if the nft data are not already loaded, it loads them, it copies the input frame in pinArray and it uploads the buffer. From this moment, the marker detection on the incoming image can begin.

The variable detectedPage can take the value -2 if something has gone wrong with the tracking and it has ended, -1 if it has been well initialized and ≥ 0 if the tracking is still to be initialized. The ret variable takes the value 1 when a page is detected, otherwise it takes a negative value. If the pageNo indicates that the detected page is right, meaning that it is among the ones that have to be detected, the ar2SetInitTrans function sets an initial transform for subsequent NFT texture tracking. It thus returns the value 0 if it is successful or -1 in case of an error. Any time a problem occurs, the detectedPage is getting the value -2, to give the system the information that the tracking procedure has not been initialized.

```
for (i = 0; i < markersNFTCount; i++) {  
    markersNFT[i].validPrev = markersNFT[i].valid;  
    if (markersNFT[i].pageNo >= 0 && markersNFT[i].pageNo == detectedPage)  
    {  
        markersNFT[i].valid = TRUE;  
        for (j = 0; j < 3; j++)  
            for (k = 0; k < 4; k++) {  
                markersNFT[i].trans[j][k] = trackingTrans[j][k];  
            }  
    }  
    else markersNFT[i].valid = FALSE;  
  
    if (markersNFT[i].valid) {  
        if (!markersNFT[i].validPrev) {  
            VirtualEnvironmentHandleARMarkerAppeared(i);  
        }  
  
        arglCameraViewRHf(markersNFT[i].trans, markersNFT[i].pose.T,  
                           1.0f);  
        VirtualEnvironmentHandleARMarkerWasUpdated(i, markersNFT[i].pose);  
    } else {  
        if (markersNFT[i].validPrev) {  
            VirtualEnvironmentHandleARMarkerDisappeared(i);  
        }  
    }  
}  
} // nativeVideoFrame ends here
```

The next step for the program is to search every marker that possibly exists in the image. When a marker is found, its transformation matrix changes according to a 3x4 matrix, named trackingTrans and a new pose is estimated. The VirtualEnvironmentHandleARMarkerAppeared and VirtualEnvironmentHandleARMarkerDisappeared functions are responsible of making known when a marker becomes visible or invisible.

The trackingTrans matrix has been filled out with the pose when the ar2SetInitTrans()

have been called, setting the initial tracking transform. The initial transform is obtained from NFT KPM tracking, but it could be also obtained via a fiducial marker embedded in the NFT image.

➤ Model's rendering

For the rendering, the `VirtualEnvironmentInit()` function, written in C language, is used. As it would take a lot of pages to describe all the details of the function, only the most important parts will be presented.

```
if (!VirtualEnvironment_AROSG) {
    VirtualEnvironment_AROSG = arOSGInit();
    if (!VirtualEnvironment_AROSG) {
        ARLOGe("Error: unable to init arOSG library.\n");
        return (0);
    }
}
```

The function is supposed to load an OSG model, so an OSG initialization is the first important part of the code. In case the initialization fails an error appears to inform the user.

```
fp = fopen(objectListFile, "r");
if (fp == NULL) {
    goto bail1;
}

numObjects = 0;
get_buff(buf, MAXPATHLEN, fp, 1);
if (sscanf(buf, "%d", &numObjects) != 1) {
    goto bail2;
}

if (objects) {
    free(objects); objects = NULL;
    objectCount = 0;
}

objects = (VEObject *)calloc(numObjects, sizeof(VEObject));
if (!objects) {
    goto bail2;
}

// where
bail1:
    if (VirtualEnvironment_AROSG) {
        arOSGFinal(VirtualEnvironment_AROSG);
        VirtualEnvironment_AROSG = NULL;
    }

// and
bail2:
    fclose(fp);
```

It then locates and opens the objects' description file. In case the system is unable to open it, the virtual environment is declared null and the procedure stops. Otherwise, the code

continues with the `get_buff` function. This function is associated with a `fgets()` function which returns the length of the file, without its not important entries such as comments and blank lines, meaning the actual number of objects. The next step is to allocate space for the objects. In any case an error occurs during the loading of the objects, the function closes the file.

```
for (i = 0; i < numObjects; i++) {
    if (!get_buff(buf, MAXPATHLEN, fp, 1)) {
        ARLOGe("Error: unable to read model file name from object data
               file.\n");
        goto bail3;
    }
    if (!arUtilGetDirectoryNameFromPath(objectFullpath, objectListFile,
        sizeof(objectFullpath), 1)) {
        goto bail3;
    }
    strncat(objectFullpath, buf, sizeof(objectFullpath) -
        strlen(objectFullpath) - 1);

    // Translation.
    get_buff(buf, MAXPATHLEN, fp, 1);
#ifdef ARDOUBLE_IS_FLOAT
    if (sscanf(buf, "%f %f %f", &translation[0], &translation[1],
        &translation[2]) != 3)
#else
    if (sscanf(buf, "%lf %lf %lf", &translation[0], &translation[1],
        &translation[2]) != 3)
#endif
    {
        goto bail3;
    }
    // Rotation.
    get_buff(buf, MAXPATHLEN, fp, 1);
#ifdef ARDOUBLE_IS_FLOAT
    if (sscanf(buf, "%f %f %f %f", &rotation[0], &rotation[1],
        &rotation[2], &rotation[3]) != 4)
#else
    if (sscanf(buf, "%lf %lf %lf %lf", &rotation[0], &rotation[1],
        &rotation[2], &rotation[3]) != 4)
#endif
    {
        goto bail3;
    }
    // Scale.
    get_buff(buf, MAXPATHLEN, fp, 1);
#ifdef ARDOUBLE_IS_FLOAT
    if (sscanf(buf, "%f %f %f", &scale[0], &scale[1], &scale[2]) != 3)
#else
    if (sscanf(buf, "%lf %lf %lf", &scale[0], &scale[1], &scale[2]) != 3)
#endif
    {
        goto bail3;
    }

    // where
bail3:
    free(objects);
    objects = NULL;
```



```
objectCount = 0;
```

Following, all the info related to the object are read and a path separator is used in order to read the directory name of the model's file. The name of the file is then added to the end of the object's description file. From this moment the model can be translated and its rotation and scale can be computed. In any case the procedure fails, either because of an error before the appendix of the model, or during the translation, rotation and scaling the value of the object is being set null, after the memory they consumed is getting available again using the free() function.

```
objects[i].modelIndex = arOSGLoadModel2(VirtualEnvironment_AROSG,
    objectFullpath, translation, rotation, scale);
if (objects[i].modelIndex < 0) {
    ARLOGE("Error attempting to read object data file %s.\n",
        objectFullpath);
    goto bail4;
}

objectCount++;
} // end of for loop

fclose(fp);
return (objectCount);
bail4:
for (i--; i >= 0; i--) {
    arOSGUnloadModel(VirtualEnvironment_AROSG, i);
}
```

Finally, after some modifications concerning the lighting and the markers which have not been included in the part of the code presented above, the model, together with its path and its translation, rotation and scale information, is loaded to the object's modelIndex. If the modelIndex's value remains negative the object data file cannot be read and a function unloads the model. Otherwise, the objectCount's value raises and the next object can be examined.

When all the objects have been loaded, the objectListFile closes and the final value of the objectCount is returned to the VirtualEnvironmentInit() function.

➤ Closing the application

When it is time to close the application, the onStop() method is called.

```
public void onStop() {
    super.onStop();
    nftActivity.nativeStop();
}
```

To ensure that everything will close correctly, the onStop() method calls the nativeStop() function.

```
JNIEXPORT jboolean JNICALL JNIFUNCTION_NATIVE(nativeStop(JNIEnv* env, jobject
    object))
{
#ifdef DEBUG
    LOGI("nativeStop\n");
#endif
}
```

```
int i, j;

if (trackingThreadHandle) {
#ifdef DEBUG
    LOGI("Stopping NFT2 tracking thread.");
#endif
    trackingInitQuit(&trackingThreadHandle);
    detectedPage = -2;
}
j = 0;
for (i = 0; i < surfaceSetCount; i++) {
    if (surfaceSet[i]) {
#ifdef DEBUG
        if (j == 0) LOGI("Unloading NFT tracking surfaces.");
#endif
        ar2FreeSurfaceSet(&surfaceSet[i]);
        j++;
    }
}
#ifdef DEBUG
    if (j > 0) LOGI("Unloaded %d NFT tracking surfaces.", j);
#endif
    surfaceSetCount = 0;
    nftDataLoaded = false;
#ifdef DEBUG
    LOGI("Cleaning up ARToolkit NFT handles.");
#endif
    ar2DeleteHandle(&ar2Handle);
    kpmDeleteHandle(&kpmHandle);
    arParamLTFree(&gCparamLT);

    if (gVideoFrame) {
        free(gVideoFrame);
        gVideoFrame = NULL;
        gVideoFrameSize = 0;
    }
    ar2VideoClose(gVid);
    gVid = NULL;
    videoInitied = false;
    return (true);
}
```

As it can be seen, this function is responsible of cleaning up all the parts of the application that are still open. First of all, it finishes the tracking procedure, giving to the detectedPage the value -2, which indicates that the tracking has ended. It also unloads the NFT tracking surfaces and it sets the surfaceSet to null. Then, it cleans up all the libraries that have been used to handle the tracking procedure. Finally, it allocates the space occupied by the video frames and it stops the video recording.

It is important to mention again that the functions presented above are not quoted as a whole; only parts essential to the explanation of the aforementioned steps have been included. That means that in some cases a sense of discontinuity may exists, but it was inevitable in order to avoid an extended and exhausting code quotation.

3.5.2. Running the application for the first time

Once the building had been concluded, there is one more issue to be taken into consideration before the application is ready to be tested. This issue concerns the camera's calibration.

3.5.2.1. Camera's Calibration

The existing file "camera_para.dat" of the ARToolKit software, which is being read each time the application is initialized, contains default camera properties that are sufficient for basic tracking for a wide range of different cameras. The tracking accuracy, though, is really important, especially when it comes to real-time projects, and in order to achieve the best results, a cautious calibration is crucial.

ARToolKit offers a utility program, named `calib_camera`, which can generate a separate parameter file for the specific camera that is being used. It works by capturing images of a pre-prepared calibration pattern of black and white squares surrounded by a white border, the image file of which is provided with the SDK. It locates the corners of the squares and then it measures the spacing between the corners. This information is used to calculate the lens distortion measurement, because of which the straight lines naturally blend when they are viewed through the camera lens.

When `calib_camera` is executed using the command prompt, the default calibration pattern settings appear on the screen. These are the number of intermediate corners in horizontal and vertical directions (7 for X axis and 5 for Y axis), the width of every square (30 cm), the number of the images that were going to be captured for the calibration procedure (10) and the video configuration image size (640,480). As the device that is used does not support (640,480) images to be captured, the image size was modified to (2048,1152) using the `-vconf` parameter, maintaining the 16:9 aspect ratio. The rest of the settings were fine, so there was no need for further modifications.

When the chessboard grid is viewed by the camera, the inner corners of the squares appear highlighted with 'X' marks and numbered. As long as the view is poor, these marks are green and it is not possible for the calibration image to be captured. When all the corners are clearly seen by the camera the marks turn red and the capture is enabled. This procedure has to be repeated 10 times, as the default settings indicated. The 10 images have to be obtained at a variety of angles to the camera lens, to ensure a good calibration for the camera.

For each one of the shots the relative positions of the 'X' indicators are calculated and so is the distortion factor associated to the camera device. To consider the calibration data good, the estimated error in each image has to be low. The estimated error was less than one for every image captured, it has thus been accepted.

3.5.2.2. First results

The application's rendering code has been based to the NFTBook example of the ARToolKit framework, where an animated propeller plane along with a 3D axis has to be superimposed upon an image, using Natural Feature Tracking. The model .osg file has been also used for the first test, in order to ensure that the rendering code was not affected by the modifications that have been made to the rest of the code. The result was successful.

As a next step, the initial project's image has been replaced with the image of the floor plan that has to be tracked, and the essential modifications to the code have been

realized. Once again, the result was successful, as the 3D axis has been superimposed on the origins of the trackable image, where it was supposed to appear, and the model of the plane has also been superimposed on a random position.

Subsequent to the image's replacement was the replacement of the model. As the original model was of .osg type, an attempt to load the 3D model of Jesus's Tomb in .osg type has been made. However, even after the model's size reduction that have been described, the model was too big to be supported. The jnlibrary has included a plugin in order to enable the rendering to be realized with an .obj model.

Unfortunately, as it was revealed during the effort to use it, the gnu C++ runtime library, which was essential for the completion of this procedure, has been abandoned. As a result, it is not any more compatible with the rest of the libraries and models with types different from the original cannot be supported [92]. It is a known issue for the developers of ARToolKit, but as a new release of the software is being built at the moment, they are not trying to face it.

3.5.3. OBJ loader's creation

Since it was not possible to produce the desirable result with the current rendering code under the specific circumstances, an attempt for the creation of a completely new renderer code, designed specifically for the loading of .obj models, has been made. Its basic parts are the next thing to be presented and explained.

```
public class OBJLoader extends Object {
    public static RawModel LoadObjModel (String fileName, Loader loader) {
        FileReader fr = null;
        try {
            fr = new FileReader(new File("res/Tomb.obj"));
        } catch (FileNotFoundException e) {
            System.err.println("Couldn't load file!");
            e.printStackTrace();
        }
        BufferedReader reader = new BufferedReader(fr);
        String line;
        List<Vector3f> vertices = new ArrayList<Vector3f>();
        List<Vector2f> textures = new ArrayList<Vector2f>();
        List<Vector3f> normals = new ArrayList<Vector3f>();
        List<Integer> indices = new ArrayList<Integer>();

        float[] verticesArray = null;
        float[] normalsArray = null;
        float[] textureArray = null;
        int[] indicesArray = null;
    }
}
```

The OBJLoader class uses the LoadOBJModel() method to return the .obj model as a raw file. A FileReader is created and the .obj model is added in it. A BufferReader is then created to handle the attributes' reading. The next elements to be created are four lists; one for the vertices, one of the textures and one for the normals of the .obj file and the last one to provide information concerning the file's indices. Their values will be assigned to the four relevant arrays that the class can handle.

```
try {
    while(true) {
```

```
line = reader.readLine();
String[] currentLine = line.split(" ");
if (line.startsWith("v ")) {
    Vector3f vertex = new Vector3f(Float.parseFloat(
        currentLine[1]),Float.parseFloat(currentLine[2]),
        Float.parseFloat(currentLine[3]));
    vertices.add(vertex);
}else if (line.startsWith("vt ")) {
    Vector2f texture = new Vector2f(Float.parseFloat(
        currentLine[1]),Float.parseFloat(currentLine[2]));
    textures.add(texture);
}else if (line.startsWith("vn ")) {
    Vector3f normal = new Vector3f(Float.parseFloat(
        currentLine[1]),Float.parseFloat(currentLine[2]),
        Float.parseFloat(currentLine[3]));
    normals.add(normal);
}else if (line.startsWith("f ")) {
    textureArray = new float[vertices.size()*2];
    normalsArray = new float[vertices.size()*3];
    break;
}
}
```

The try block is used to check possible invalid formats from the .obj file. Inside it, the while loop reads each one of its lines, until either the last line or the break point has been reached. Each line is splitted to parts with the line.split() method. The lines starting with v contain vertices, vt is for textures, vn for normals and f for faces. Every registration is added to the relevant list. In .obj models the faces' registrations begin after the ones for vertices, textures and normals have ended. Because of that, when the line starts with an f it means that there are no more elements to be added to the lists. The size of the texture and the normal arrays can thus be computed. The break statement used stops the reading of the file to the first face's registration.

```
while(line!=null) {
    if (!line.startsWith("f ")) {
        line = reader.readLine();
        continue; //
    }
    String[] currentLine = line.split(" ");
    String[] vertex1 = currentLine[1].split("/");
    String[] vertex2 = currentLine[2].split("/");
    String[] vertex3 = currentLine[3].split("/");

    processVertex(vertex1,indices,textures,normals,textureArray,normalsArray);
    processVertex(vertex2,indices,textures,normals,textureArray,normalsArray);
    processVertex(vertex3,indices,textures,normals,textureArray,normalsArray);
    line = reader.readLine();
}
reader.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

A second while loop starts, which is going to be repeated until a null line is found,

which means until the end of the file, as there are no other empty lines in an .obj file. The reading is continued from the first face line.

The faces are defined using lists of vertex, texture and normal indices. Polygons such as quadrilaterals can be defined by using more than three vertex/texture/normal indices. Optionally, normal indices can be used to specify normal vectors for vertices when defining a face. To add a normal index to a vertex index when defining a face, a second slash must be put after the texture coordinate index, followed by the normal index. A valid normal index starts with 1 and matches the corresponding element in the previously defined list of normals. Each face can contain three or more elements. A face line in the .obj file has thus the form “f v1/vt1/vn1 v2/vt2/vn2 v3/vt3/vn3 etc. For this reason, the code splits each line in sections and the slash symbols divide these sections in vertices. The processVertex() method is used to associate every vertex's information with the corresponding arrays.

```
verticesArray = new float[vertices.size()*3];
indicesArray = new int[indices.size()];
int vertexPointer = 0;
for(Vector3f vertex:vertices) {
    verticesArray[vertexPointer++] = vertex.x;
    verticesArray[vertexPointer++] = vertex.y;
    verticesArray[vertexPointer++] = vertex.z;
}
for(int i=0;i<indices.size();i++) {
    indicesArray[i] = indices.get(i);
}
return loader.loadToVAO(verticesArray, textureArray, indicesArray);
}
```

Since the vertices and the indices' sizes are now known, the values to the corresponding arrays can be set, using the ones in the relevant lists. The arrays containing the vertices, the textures and the indices can be returned to the class.

The connection of the current class with the rest of the code could possibly solve the rendering problem that has been created. However hard it has been tried, though, the developer's experience was not enough to permit this connection to be realized. The attempt to create the application, thus, has stopped for the moment.

4. Results and Commentation

The last chapter's purpose is to conclude and sum up the important parts of the thesis and to present the application's results. Furthermore, future work that can be done to expand the application will be discussed here.

4.1. Conclusions

The current thesis' purpose was to explain the term of Augmented Reality, framing its historical context and describing in detail its technology, and to describe its role in the field of preservation and presentation of the cultural heritage. In second place, the development of an application has been attempted, to present a monument of great historical and theological significance. The first purpose has been achieved, while the second has been failed.

The application has been proved to be extremely complicated and time consuming. Internet, which is the source of every type of information, does not contain sufficient material for the familiarization with the particular subject of augmenting the reality using a smartphone device, as it does for other domains. The problems that have been emerged were strenuous to be faced in a logical timeframe. For all these reasons, the desired final result has not been accomplished.

Having said that, though, it is significant to point out the knowledge, understanding, capabilities and attitudes that have been developed during this incomplete application production. At the beginning of the procedure, the computer familiarity level was low, understanding code written in Java was impossible and handling 3D models' skills were at basic degree. At the present moment, the comprehension level is high for all of the above. Taking into consideration that this is the actual goal of a thesis, the overall experience can be considered successful.

4.2. Application's results

Even though the application is not complete, it has already been said that the problems start when the new 3D model has to be rendered. Until this point the code runs perfectly fine, tracking the image of Aedicule's floor-plan based on its natural features. To show how it was supposed to work, thus, the results acquired with the right image and the wrong object can be described and visualized.

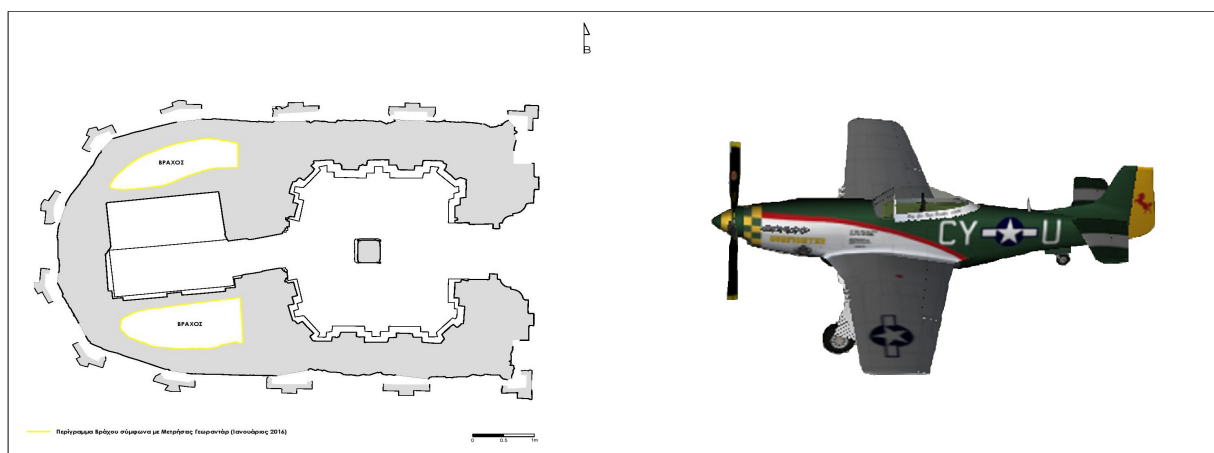


Figure 27: Right image to be tracked (left) and wrong model (plane) to appear (right)

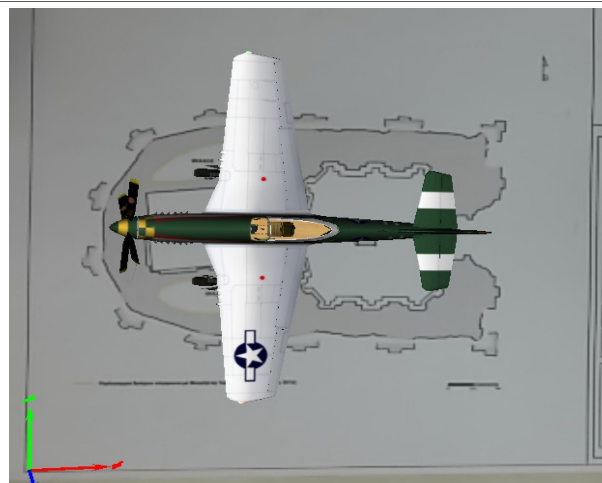
The way the program is designed, when the image is tracked the model is at first placed at a pose indicated by a file that contains information about all the digital objects that are going to be superimposed on the image. Among the data provided there, the coordinates, the rotations and the dimensions of every item can be found and manually corrected.

The information provided are of the type

$$\begin{bmatrix} x & y & z \\ \omega & \varphi & \kappa \\ s1 & s2 & s3 \end{bmatrix}$$

where x, y and z are the coordinates, ω , φ , and κ are the rotations around the X, Y and Z axis and s1, s2 and s3 are the model's three dimensions.

It has been decided for the visualization of the results the trirectangular system of coordinates to be placed at the origins of the tracked image, meaning at the down left corner of its borderline and the plane model to be located at the image's center, as it appears at the first image of figure 28. At this figure the table containing the values that have be given to the application for the placement of the plane model appear next to the relevant view that displays on the camera. The coordinates of the point considered as the origins of the image (where the 3 axis are encountered) is (-12.0, -37.0, 0.0)



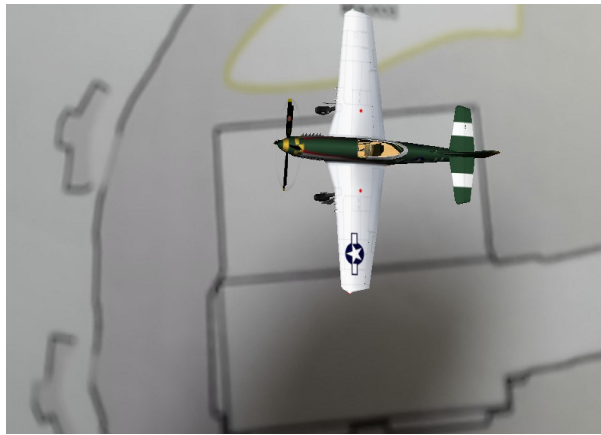
$$\begin{bmatrix} 15.0 & 100.0 & 100.0 \\ 0.0 & 0.0 & 0.0 \\ 15.0 & 15.0 & 15.0 \end{bmatrix}$$

As it can be seen, when the smartphone's camera's view is vertical the tracking procedure is very effective, as the trirectangular system of coordinates appears exactly at the place where it should be located. There is no way to check the plane's coordinates but this view will be used as a reference for the evaluation of the others.



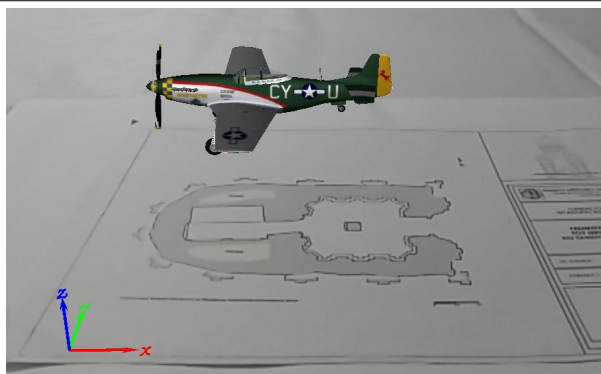
$$\begin{bmatrix} 15.0 & 100.0 & 100.0 \\ 0.0 & 0.0 & 0.0 \\ 15.0 & 15.0 & 15.0 \end{bmatrix}$$

The result when the distance from the trackable image is increased is indicated in this snapshot. The distance is about 1 meter, which is the maximum distance from which the model was superimposed. It looks like there are errors on the coordinates but as the axes are located where they should be, it is probably an illusion created by the model's z coordinate's value.



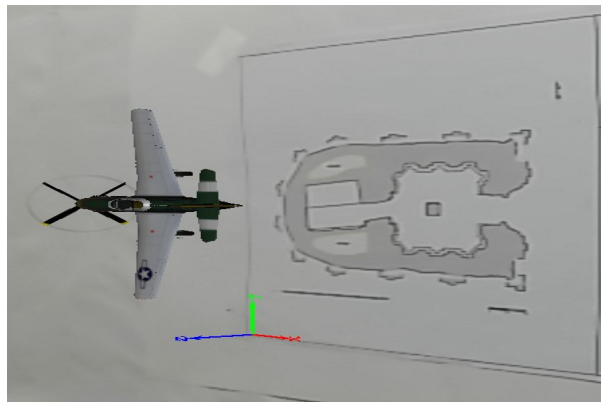
$$\begin{bmatrix} 15.0 & 100.0 & 100.0 \\ 0.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 1.0 \end{bmatrix}$$

An image taken by the minimum distance in which the superimposition is possible appears here. The distance is about 5 cm. The size of the object in this case has been changed, so as the whole model to appear on the display. The results in this case were very unstable, the object was constantly trembling and the coordinates' error very important.



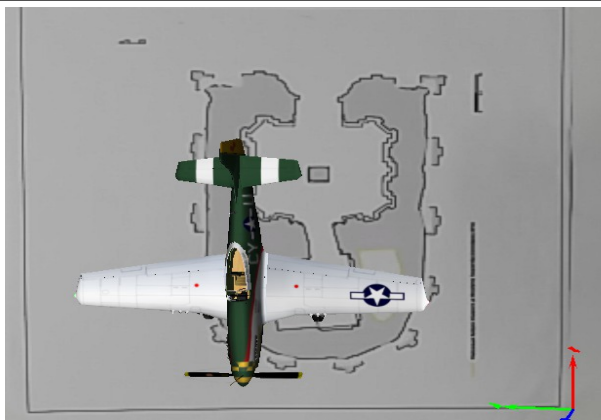
$$\begin{bmatrix} 15.0 & 100.0 & 100.0 \\ 0.0 & 0.0 & 0.0 \\ 15.0 & 15.0 & 15.0 \end{bmatrix}$$

In this case the maximum ω viewing angle is examined. As it can be seen, as the angle raises the coordinate's error raises too, especially in x axis, which has been significantly moved. The maximum value before the model's disappearance has been proved to be around 60 degrees.



$$\begin{bmatrix} 15.0 & 100.0 & 100.0 \\ 0.0 & 0.0 & 0.0 \\ 15.0 & 15.0 & 15.0 \end{bmatrix}$$

On the same note, the maximum ϕ viewing angle is the next to be inspected. The paradox here is that no error appears, instead of what happens in the previous case. The maximum rotation possible around the Y axis is also around 60 degrees.



$$\begin{bmatrix} 15.0 & 100.0 & 100.0 \\ 0.0 & 0.0 & 0.0 \\ 15.0 & 15.0 & 15.0 \end{bmatrix}$$

The rotation around the z axis seems to be a problem rather bigger than what it was expected. At first it was considered as an error arising from the z axis not being on the perpendicular, but no matter how carefully the procedure has been repeated, the result would not become any better than the one seen in the snapshot.

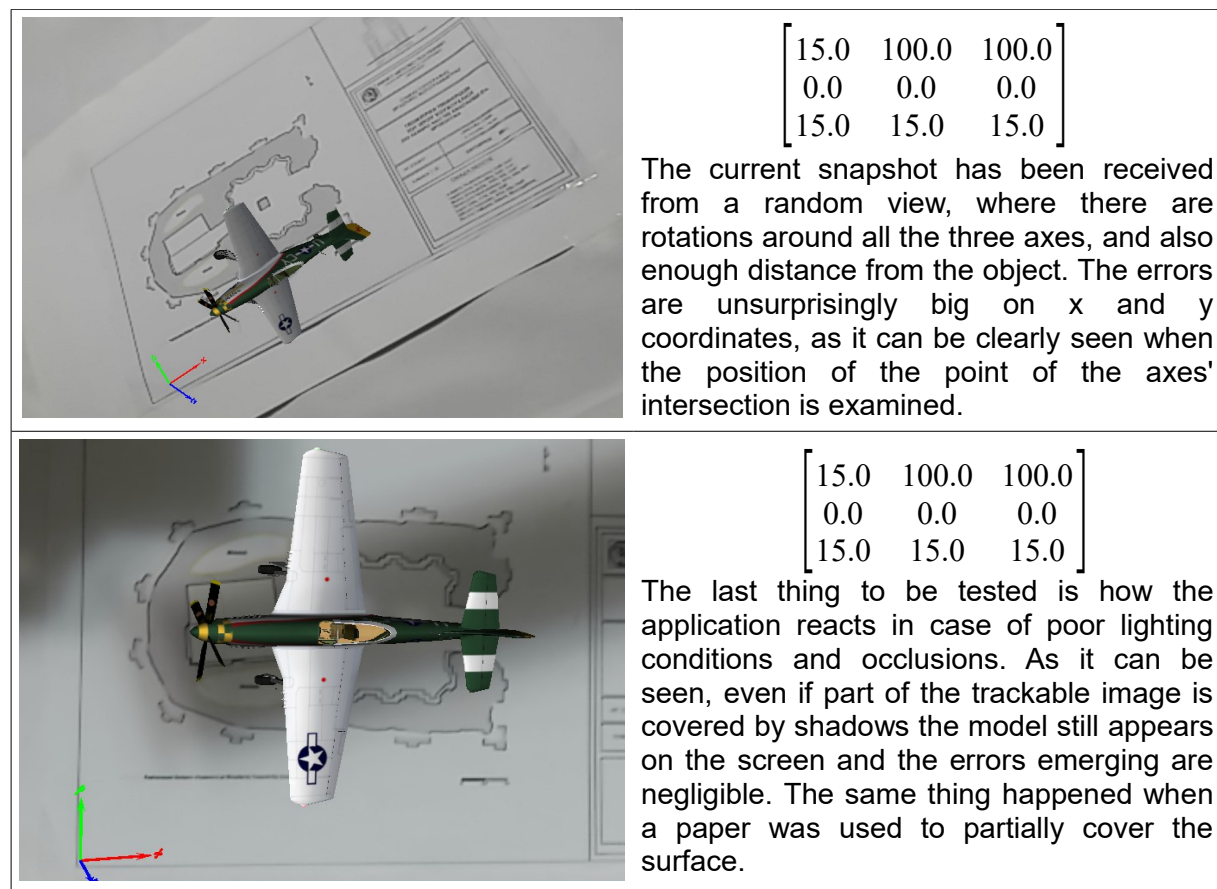


Figure 28: Examples of the application's results

In all the cases described above, for the model's first appearance the camera had to vertically view the trackable image the distance between them had to be around 30 cm. It was also necessary not to have occlusion on the image. However, once the tracking had been realized, it was possible to change all of these factors without having the object's disappearance as a result.

Summing up the information learned from the tests, the model totally vanishes from the view in three cases. When the distance is less than 5 cm or more than 1 meter, when the viewing angle is more than 60 degrees or when the illumination is not adequate. However, even when these limits are kept, the closest the value is to them the lower the accuracy is. Moreover, in such an occasion the result becomes very unstable and the model cannot stay in one place.

4.3. Future work

First of all, for the application to run correctly two possible solutions exist. The first one depends on the modifications that the developers of ARToolKit are eager to realize. If a new library, compatible to the rest of the code, takes the place of the problematic one, then the project is expected to well fine. The second solution is a brand-new renderer module to be created, which is what has been tried with the construction of the OBJloader. Unfortunately, in this attempt the previous renderer kept on causing problems, as in the way the original system has been built, it is really hard to totally delete a whole part of the code and replace it with another. The methods and functions are mixed up, demanding expertise

to both Java language and native code, as well as to the way the development environment, in this case Android Studio, is designed. To acquire this type of expertise, a lot of time and occupation is needed. The best technique for someone who is willing to dedicate his time would be to start with the creation of simple projects and to progressively delve into it.

Once the application runs as it should, a lot of features could be added. A first idea is the code to offer the possibility to stop the image on a specific shot and to zoom in and out. The user could use this operation to observe details on the model that are not possible to be seen from the original distance. Another facility could be the environment to be changed from augmented to digital when the user decides it. The digital environment would be composed by the model situated to its original place, in this case the museum of the Holy Sepulchre. Rotating the 3D model, the view of the museum would also change. In a virtual environment it would be also interesting to be able to change the illumination and observe the object under different conditions. One more possibility could be to adjust a timeline with which the user could interact, in order to admire the monument through different time periods, in real-time. Furthermore, hotspots could be added on significant parts of the Aedicule, which would correspond to information panels with textual and media content that the user could access by touching them.

To bring all these ideas to life simultaneously, the computational power should be way too big, especially when compared with the current systems. However, as the technology evolves, it is quite likely the hand-held devices like tablets or even smartphones to be powerful enough to support such features in some years from now.

Reference List

- (1) Abernathy, MF, Houchard, J, Puccetti, MG, and Lambert, JV (1993) 'Debris Correlation Using the Rockwell WorldView Simulation System', *MIT Lincoln Laboratory Space Surveillance Workshop Proceedings*, vol. 1, pp. 189-195.
- (2) Alahi, A, Ortiz, R, Vandergheynst, P (2012) 'FREAK: Fast Retina Keypoint', in *Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 510-517.
- (3) Azuma, R (1997) 'A Survey of Augmented Reality', in *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355-385.
- (4) Banterle, F, Cardillo, FA, Malomo, L (2015) '*LecceAR: an augmented reality App for cultural heritage*', In *ERCIM News*, vol. 2015, no. 103, pp. 16 - 17.
- (5) Barrilleaux, J (1999) 'Experiences and Observations in Applying Augmented Reality to Live Training'.
- (6) Bederson, B (1995) 'Audio augmented reality: a prototype automated tour guide', *Conference Companion on Human Factors in Computing Systems*, pp. 210-211.
- (7) Bell, B, Feiner, S, Höllerer, T (2001) 'View Management for Virtual and Augmented Reality', *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology, New York, USA*, pp. 101–110.
- (8) Billinghamurst, M, Clark, A, Lee, G (2015) 'A Survey of Augmented Reality', *The Human Interface Technology Laboratory New Zealand*, vol. 8, No 2-3, pp. 73-272.
- (9) Bimber, O, Raskar, R, (2005) 'Spatial Augmented Reality Merging Real and Virtual Worlds', *Ltd. Natick, MA, USA*, pp. 71-90.
- (10) Bimber, O, Raskar, R, (2006) 'Modern Approaches to Augmented Reality', *Mitsubishi Electric Research Laboratories*, pp. 1-11.
- (11) Bostanci, E, Kanwal, N, Ehsan, S, Clark, AF (2013) 'User tracking methods for augmented reality', in *International Journal of Computer Theory and Engineering*, vol. 1, no. 5, pp. 93-98.
- (12) Caudell, TP, Mizell, DW (1992) 'Augmented Reality: an application of heads-up display technology to manual manufacturing processes', *System Sciences, Proceedings of the Twenty-Fifth Hawaii International Conference on*, vol. 2, pp. 659-669.
- (13) Chen, J, Pinz, A (2004) 'Structure and motion by fusion of inertial and vision-based tracking', in *Proceedings of the 28th ÖAGM/AAPR Conference*, vol. 179, pp. 55–62.
- (14) Falk, J, Redström, J, Björk, S (1999) 'Amplifying Reality', *1st International Symposium of Handheld and Ubiquitous Computing, Springer, Karlsruhe, Germany*, pp. 274-280.

- (15) Feiner, S, MacIntyre, B, Höllerer, T, Webster, A (1997) 'A touring machine: prototyping 3d mobile augmented reality systems for exploring the urban environment', *IEEE International Symposium on Wearable Computers*, pp. 74–81.
- (16) Fritz, F, Susperregui, A, Linaza, MT (2005) 'Enhancing Cultural Tourism experiences with Augmented Reality Technologies', in *VAST05: The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*.
- (17) Georgopoulos, A, Karras, G, Stergiopoulos, J (2001) 'Inverse Photogrammetry “anticipates” the future', in *CIPA International Symposium, Potsdam*.
- (18) Heilig, M (1955) 'The Cinema of the Future', *Presence, I* (3), pp. 279-294.
- (19) Helle, S, Salmi, H, Turunen, M, Woodward, C, Hehtonen, T (2017) 'MIRACLE Handbook: Guidelines for Mixed Reality Applications for Culture and Learning Experiences'.
- (20) Holloway, RL (1995), 'Registration Errors in Augmented Reality Systems', *University of North Carolina at Chapel Hill*, pp. 199-205.
- (21) Jokilehto, J (2005) 'Definition of Cultural Heritage, References to Documents in History', Originally for ICCROM, 1990.
- (22) Keil, J, Pujol, L, Roussou, M, Engelke, T, Schmitt, M, Eleftheratou, S (2013) 'A digital look at physical museum exhibits Designing personalized stories with handheld Augmented Reality in museums', in *Conference: Digital Heritage International Congress, Marseille*, Vol. 2
- (23) Keil, J, Zollner, M, Becker, M, Wientapper, F, Engelke, T, Wuest, H (2011) 'The house of Olbrich - an augmented reality tour through architectural history', in *Proceedings of the 2011 IEEE International Symposium on Mixed and Augmented Reality – Arts, Media and Humanities*, pp. 15-18, IEEE.
- (24) Klein, G (2006) 'Visual Tracking for Augmented Reality - PhD Thesis', in *University of Cambridge*.
- (25) Kooper, R, MacIntyre, B (2003) 'Browsing the Real-World Wide Web: Maintaining Awareness of Virtual Information in an AR Information Space', *International Journal of Human-Computer Interaction*, vol. 16, no. 3, pp. 425–446.
- (26) Lepetit, V, Berger, M-O (2001) 'An Intuitive Tool For Outlining Objects in Video Sequences: Applications to Augmented and Diminished Reality', *Proceedings of International Symposium on Mixed Reality (ISMAR)*, pp. 1-2.
- (27) Low, K, Welch, G, Lastra, A, Fuchs, H (2001) 'Life-Sized Projector-Based Dioramas', in *ACM Symposium on Virtual Reality Software and Technology, Banff, Alberta, Canada, November 2001*, pp. 161–168.
- (28) Magnenat-Thalmann, N, Papagiannakis, G. (2010) 'Recreating Daily life in Pompeii', in *Virtual Archeology Review*, vol. 1, no. 2, pp. 19-23.
- (29) Mann, S (2002) 'Mediated Reality with implementations for everyday life', *Presence*:

Teleoperators and Virtual Environments, vol. 11, no. 2, pp. 158-175.

- (30) Mehling, M (2006) 'Implementation of a Low Cost Marker Based Infrared Optical Tracking System - PhD Thesis', in *Fachhochschule Stuttgart*.
- (31) Milgram, P, Kishino, F (1994) 'A Taxonomy of Mixed Reality Visual Displays', *IEICE Transactions on Information Systems*, vol. E77-D, no. 12, pp. 1321-1329.
- (32) Möhring, M, Lessig, L, Bimber, O (2004) 'Optical Tracking and Video see-Through AR on Consumer Cell-Phones', *Proceedings of International Symposium on Augmented and Mixed Reality (ISMAR '04)*, pp. 252-253.
- (33) Moropoulou, A, Korres, Em, Georgopoulos, A, Spyarakos, C, Mouzakis, Ch (2017) 'Faithful Rehabilitation of the Holy Tomb of Christ', in *Journal of ASCE*, November 2017, pp. 54-78
- (34) Neumann, U, You, S, Cho, Y, Lee, J, Park, J (1999) 'Augmented reality tracking in natural environments', in *International Symposium on Mixed Reality*.
- (35) Papagiannakis, G, Ponder, M, Molet, T, Kshirsagar, S, Cordier, F, Magnenat-Thalmann, N, Thalmann, D (2002) 'LIFEPLUS: Revival of life in ancient Pompeii', in *Proceedings of the 8th International Conference on Virtual Systems and Multimedia (VSMM'02)*, pp. 25-27.
- (36) Pujol, L, Roussou, M, Poulou, S, Balet, O, Vayanou, M, Ioannidis, Y (2012) 'Personalizing interactive digital storyteeling in archaeological museums: the CHESS project', in *40th Annual Conference of Computer Applications and Quantitative Methods in Archaeology, Amsterdam University Press*.
- (37) Rambach, J, Pagani, A, Schneider, M, Artemenko, O, Stricker, D (2017), '6DoF Object Tracking based on 3D Scans for Augmented Reality Remote Live Support', in *Basel, Switzerland*.
- (38) Raskar, R, Welch, G, Cutts, M, Lake, A, Stesin, L, Fuchs, H (1998) 'The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays', *Computer Graphics Proceedings, Annual Conference Series*, pp. 179-188.
- (39) Reitmayr, G, Drummond, TW (2006) 'Going out: Robust Tracking for OutDoor Augmented Reality', *Proceedings of the 5th IEEE and ACM International Symposium on*, pp. 109-118.
- (40) Rosenberg, LB (1993) 'Virtual fixtures: Perceptual tools for telerobotic manipulation', *Proceedings of the IEEE Annual International Symposium on Virtual Reality*, pp. 76-82.
- (41) Rekimoto, J, Ayatsuka, Y (2000) 'CyberCode: designing augmented reality environments with visual tags', *Proceedings of DARE 2000 on designing Augmented Reality environments*, pp. 1-10.
- (42) Rekimoto, J, Nagao, K (1995) 'The world through the computer: Computer augmented interaction with real world environments' *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology, New York, USA*, pp. 29-36.
- (43) Ruiz, R (2004) 'Augmented reality used to recreate a Mayan funeral chamber', in Valencia, Spain.

- (44) Siltanen, S (2012) 'Theory and applications of marker-based augmented reality'.
- (45) Starner, T, Mann, S, Rhodes, BJ, Levine, J, Healey, J, Kirsch, D, Picard, RW, Pentland, A (1997) 'Augmented reality through wearable computing', *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 386–398.
- (46) Strand, TER (2008) 'Tracking for Outdoor Mobile Augmented Reality - Further development of the Zion Augmented Reality Application', in *Norwegian University of Science and Technology*.
- (47) Sutherland, I (1965) 'The Ultimate Display', in Proceedings of the IFIP Congress.
- (48) Sutherland, I (1968) 'A Head-Mounted three-dimensional Display', *Proceedings of the Fall Joint Computer Conference, San Francisco, California*, pp. 757-764.
- (49) Thomas, B, Close, B, Donoghue, J, Squires, J, De Bondi, P, Piekarski, W (2002) 'First Person Indoor/Outdoor Augmented Reality Application: ARQuake', *Personal and Ubiquitous Computing*, vol. 6, no. 1, pp. 75-86.
- (50) Thomas, B, Demczuc, V, Piekarski, W, Hepworth, D, Gunther, B (1998) 'A Wearable Computer System with Augmented Reality to Support Terrestrial Navigation', in *Proceedings 2nd International Symposium on Wearable Computers*, pp. 168-171.
- (51) Vlahakis, V, Karigiannis, J, Tsotros, M, Gounaris, M, Almeida, L, Stricker, D, Gleue, T, Christou, IT, Carlucci, R and Ioannidis, N (2001) 'Archeoguide: First results of an augmented reality, mobile computing system in cultural heritage sites', in *Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage, VAST '01, (New York, NY, USA)*, pp. 131–140.
- (52) Welch, G, Bishop, G, Vicci, L, Brumback, S, Keller, K, Colucci, D (2001) 'High-performance wide-area optical tracking: The hiball tracking system', in *Presence: Teleoperators and Virtual Environments*, pp. 1–22, 2001.
- (53) Zoellner, M, Stricker, D, Bleser, G, Pastarmov, Y (2007) 'iTACITUS – Novel Interaction and Tracking Paradigms for Mobile AR'.
- (54) Zöllner, M, Keil, J, Wuest, H, Pletinckx, D (2009) 'An Augmented Reality Presentation System for Remote Cultural Heritage Sites', in *Short and Project Papers Proceedings, University of Malta*, pp. 112-116. sel 46
- (55) A tutorial on binary descriptors – part 4 – The BRISK descriptor (Last Modified: 08/09/2013), Available at: <https://gilscvblog.com/2013/11/08/a-tutorial-on-binary-descriptors-part-4-the-brisk-descriptor/> (Last Accessed: 18/03/2018)
- (56) A tutorial on binary descriptors – part 5 – The FREAK descriptor (Last Modified: 09/12/2013), Available at: <https://gilscvblog.com/2013/12/09/a-tutorial-on-binary-descriptors-part-5-the-freak-descriptor/> (Last Accessed: 18/03/2018)
- (57) An Introduction to Simultaneous Localisation and Mapping (Last Modified: 13/05/2016),

- Available at: <https://www.kudan.eu/kudan-news/an-introduction-to-slam/> (Last Accessed: 14/03/2018)
- (58) Article “Eye Am a Camera” (2012), Available at: <http://techland.time.com/2012/11/02/eye-am-a-camera-surveillance-and-sousveillance-in-the-glassage/> (Last Accessed: 11/01/2017)
- (59) ARToolKit (Last modified: October 2016), Available at: <https://en.wikipedia.org/wiki/ARToolKit> (Last Accessed: 11/01/2017)
- (60) ARToolKit documentation (Last Modified:2016), Available at: <https://artoolkit.org/documentation/> (Last Accessed: 18/01/2017)
- (61) ARToolKit GitHub page (Last Modified: 2017), Available at: <https://github.com/artoolkit> (Last Accessed: 18/01/2017)
- (62) ARToolKit Home Page - Human Interface Technology Laboratory, Available at: <https://www.hitl.washington.edu/artoolkit/publications/> (Last Accessed: 20/03/2018)
- (63) Augmented Reality (Last modified: September 2012), Available at: http://wiki.csisdmsz.ul.ie/wiki/Augmented_reality (Last Accessed: 11/01/2017)
- (64) Augmented Reality (Last modified: January 2017), Available at: https://en.wikipedia.org/wiki/Augmented_reality (Last Accessed: 11/01/2017)
- (65) Augmented Reality in Education (Last Modified: 2018), Available at: <https://thinkmobiles.com/blog/augmented-reality-education/> (Last accessed: 14/03/2018)
- (66) Beit Emmett: Holy Sepulchre, Succot and Hezekiah's Tunnel (Last Modified: 05/10/2009), Available at: <http://beitemmett.blogspot.gr/2009/10/holy-sepulchre-succot-and-hezekiahs.html> (Last accessed: 20/03/2018)
- (67) Bionic contact lens (Last Modified: December 2016), Available at: https://en.wikipedia.org/wiki/Bionic_contact_lens (Last Accessed: 11/01/2017)
- (68) BRIEF (Binary Robust Independent Elementary Features) (Last modified: 2013), Available at: https://en.wikipedia.org/wiki/Oriented_FAST_and_rotated_BRIEF (Last Accessed: 18/03/2018)
- (69) Bringing the past and future to life with Digital Placemaking AR Binoculars (Last Modified: 26/10/2017), Available at: <http://zubr.co/augmented-reality-binoculars/> (Last Accessed: 14/03/2018)
- (70) Christian Classics Ethereal Library (Last Modified: 13/07/2005), Available at: <http://www.ccel.org/ccel/schaff/npnf202.ii.iv.xvii.html> (Last Accessed: 14/03/2018)
- (71) Church of the Holy Sepulchre (Last Modified: 14/03/2018), Available at: https://en.wikipedia.org/wiki/Church_of_the_Holy_Sepulchre (Last Accessed: 14/03/2018)
- (72) Church of the Holy Sepulchre, Jerusalem (Last Modified: 2018), Available at: <http://www.sacred-destinations.com/israel/jerusalem-church-of-holy-sepulchre>

(Last Accessed: 14/03/2018)

- (73) Dissecting the Camera Matrix, Part 2: The Extrinsic Matrix (Last modified: 22/08/2012), Available at: <http://ksimek.github.io/2012/08/22/extrinsic/> (Last Accessed: 20/03/2018)
- (74) Dissecting the Camera Matrix, Part 3: The Intrinsic Matrix (Last modified: 13/08/2013), Available at: <http://ksimek.github.io/2013/08/13/intrinsic/> (Last Accessed: 20/03/2018)
- (75) DRAKERZ : Augmented Reality Trading Card Game (Last modified: 19/05/2014), Available at: <http://www.f2pmmogs.com/drakerz-confrontation/> (Last Accessed: 20/03/2018)
- (76) Features of a Computer (Last modified: 2015), Available at: <http://www.agriinfo.in/default.aspx?page=topic&superid=8&topicid=1703> (Last Accessed: 18/03/2018)
- (77) Fudzila (Last Updated: January 2016), Available at: <http://www.fudzila.com/news/graphics/39762-nvidia-shows-off-its-light-field-vr-headset-at-vrla-2016> (Last Accessed: 11/01/2017)
- (78) History of Information (Last Updated: December 2016), Available at: <http://www.historyofinformation.com/expanded.php?id=4698> (Last Accessed: 11/01/2017)
- (79) Hololens (Last Modified: January 2017), Available at: https://en.wikipedia.org/wiki/Microsoft_HoloLens (Last Accessed: 11/01/2017)
- (80) International Symposium on Mixed and Augmented Reality (Last modified: 09/2016), Available at: https://en.wikipedia.org/wiki/International_Symposium_on_Mixed_and_Augmented_Reality (Last Accessed: 11/01/2017)
- (81) Hartley, R, Zisserman, A (1999) 'Multiple View Geometry', Available at: http://users.cecs.anu.edu.au/~hartley/Papers/CVPR99-tutorial/tut_4up.pdf (Last Accessed: 18/03/2018)
- (82) Konstruct (2011), Available at: <http://apps.augmatic.co.uk/konstruct> (Last Accessed: 11/01/2017)
- (83) Google Glass (Last Modified: December 2016), Available at: https://en.wikipedia.org/wiki/Google_Glass (Last Accessed: 11/01/2017)
- (84) Graphics Cards: Integrated vs. Dedicated (Last modified: 2018), Available at: <http://www.toptenreviews.com/computers/articles/graphics-cards-integrated-vs.-dedicated/> (Last Accessed: 18/03/2018)
- (85) Guide to smartphone hardware (1/7) Processors, Available at: <https://lingualeo.com/ru/jungle/guide-to-smartphone-hardware-1-7-processors-83954#/page/1> (Last Accessed: 18/03/2018)
- (86) Guide to smartphone hardware (2/7): Graphics (Last modified: 21/02/2012), Available at: <https://www.neowin.net/news/guide-to-smartphone-hardware-27-graphics-2> (Last Accessed: 18/03/2018)

18/03/2018)

- (87) Guide to smartphone hardware (3/7): Memory and Storage (Last modified: 12/03/2012), Available at: <https://www.neowin.net/news/guide-to-smartphone-hardware-37-memory-and-storage> (Last Accessed: 18/03/2018)
- (88) Guide to smartphone hardware (4/7): Displays (Last modified: 26/05/2012), Available at: <https://www.neowin.net/news/guide-to-smartphone-hardware-47-displays> (Last Accessed: 18/03/2018)
- (89) How to choose a digital camera (Last modified: 2015), Available at: <http://users.wfu.edu/matthews/misc/DigPhotog/start/camera.html> (Last Accessed: 18/03/2018)
- (90) IKEA catalog uses augmented reality to give a virtual preview of furniture in a room <https://newatlas.com/ikea-augmented-reality-catalog-app/28703/#gallery>
- (91) Industrial Augmented Reality (Last Modified: December 2016), Available at: https://en.wikipedia.org/wiki/Industrial_augmented_reality (Last Accessed: 11/01/2017)
- (92) Issues · artoolkit/artoolkit5 · GitHub (Last modified: -), Available at: <https://github.com/artoolkit/artoolkit5/issues/166> (Last Accessed: 18/03/2018)
- (93) ITPRO - IBM showcases augmented reality shopping app (2013), Available at: <http://www.itpro.co.uk/apps/19376/ibm-showcases-augmented-reality-shopping-app> (Last Accessed: 11/01/2017)
- (94) Java (programming language) (Last modified: 14/03/2018), Available at: [https://en.wikipedia.org/wiki/Java_\(programming_language\)#Android](https://en.wikipedia.org/wiki/Java_(programming_language)#Android) (Last Accessed: 18/03/2018)
- (95) Java Programming Tutorial - Java Native Interface (JNI) (Last modified: 03/2018), Available at: <https://www3.ntu.edu.sg/home/ehchua/programming/java/JavaNativeInterface.html#zz-3>. (Last Accessed: 18/03/2018)
- (96) KARMA, Available at: <http://monet.cs.columbia.edu/projects/karma/karma.html> (Last Accessed: 11/01/2017)
- (97) Kinect (Last Modified: January 2017), Available at: <https://en.wikipedia.org/wiki/Kinect> (Last Accessed: 11/01/2017)
- (98) Layar (Last Modified: November 2016), Available at: <https://en.wikipedia.org/wiki/Layar> (Last Accessed: 11/01/2017)
- (99) Lunettes See-through (in french), Available at: <http://www.see-through.narvaya.fr/limites.html> (Last Accessed: 11/01/2017)
- (100) Map and Virtual Tour - Holy Sepulchre, Available at: <http://www.sepulchre.custodia.org/default.asp?id=4098> (Last Accessed: 14/03/2018)

- (101) Microsoft Hololens, Available at: https://en.wikipedia.org/wiki/Microsoft_HoloLens (Last Accessed: 20/03/2018)
- (102) Mixed Reality – Open Source, Available at: <http://trc.utu.fi/ar/research/open-source/> (Last Accessed: 20/03/2018)
- (103) Most Popular and Influential Programming Languages of 2018 (Last modified: 18/12/2017), Available at: <https://stackify.com/popular-programming-languages-2018/> (Last Accessed: 18/03/2018)
- (104) Motherboard (Last modified: 14/03/2018), Available at: <https://en.wikipedia.org/wiki/Motherboard> (Last Accessed: 18/03/2018)
- (105) Oriented FAST and rotated BRIEF (Last modified: 20/11/2017), Available at: https://en.wikipedia.org/wiki/Oriented_FAST_and_rotated_BRIEF (Last Accessed: 18/03/2018)
- (106) Photoelectric Sensors | OMRON Industrial Automation (Last modified: 2018), Available at: <https://www.ia.omron.com/support/guide/43/introduction.html> (Last Accessed: 20/03/2018)
- (107) Pokémon Go (Last Modified: January 2017), Available at: https://en.wikipedia.org/wiki/Pok%C3%A9mon_Go (Last Accessed: 11/01/2017)
- (108) Processor (computing) (Last modified: January 2017), Available at: [https://en.wikipedia.org/wiki/Processor_\(computing\)](https://en.wikipedia.org/wiki/Processor_(computing)) (Last Accessed: 11/01/2017)
- (109) Real-time 3D flight guidance (1999), Available at: <http://adsabs.harvard.edu/abs/1999SPIE.3691..149D> (Last Accessed: 11/01/2017)
- (110) Resurrection of Jesus (Last modified: 12/03/2018), Available at: https://en.wikipedia.org/wiki/Resurrection_of_Jesus (Last Accessed: 14/03/2018)
- (111) Samsung Galaxy A5 - Full phone specifications (Last modified: 2014), Available at: https://www.gsmarena.com/samsung_galaxy_a5-6761.php (Last Accessed: 18/03/2018)
- (112) Scale-invariant feature transform (Last modified: 15/02/2018), Available at: https://en.wikipedia.org/wiki/Scale-invariant_feature_transform (Last Accessed: 18/03/2018)
- (113) Smartphones Unlocked: Understanding processors (Last modified: 08/08/2011), Available at: <https://www.cnet.com/news/smartphones-unlocked-understanding-processors/> (Last Accessed: 18/03/2018)
- (114) Social Compare – Augmented Reality SDK comparison (Last update: January 2017), Available at: <http://socialcompare.com/en/comparison/augmented-reality-sdks> (Last Accessed: 17/01/2017)
- (115) Speeded up robust features (Last modified: 20/08/2017), Available at: https://en.wikipedia.org/wiki/Speeded_up_robust_features (Last Accessed: 18/03/2018)
- (116) Tango (platform) (Last modified: 24/02/2018), Available at: [https://en.wikipedia.org/wiki/Tango_\(platform\)](https://en.wikipedia.org/wiki/Tango_(platform)) (Last Accessed: 20/03/2018)

- (117) Temple of Resurrection (in greek) (Last Modified: 06/06/2018), Available at: https://el.wikipedia.org/wiki/%CE%9D%CE%B1%CF%8C%CF%82_%CF%84%CE%B7%CF%82_%CE%91%CE%BD%CE%B1%CF%83%CF%84%CE%AC%CF%83%CE%B5%CF%89%CF%82 (Last Accessed: 14/03/2018)
- (118) Terence Eden's Blog (Last Modified: 2017), Available at: <https://shkspr.mobi/blog/2010/11/augmented-reality-games-how-far-have-we-come-in-7-years/> (Last Accessed: 11/01/2017)
- (119) The Holy Edicule – Jerusalem Patriarchate, Available at: <http://www.jerusalem-patriarchate.info/main/eng/page/%CE%B9%CE%B5%CF%81%CE%BF-%CE%BA%CE%BF%CF%85%CE%B2%CE%BF%CF%85%CE%BA%CE%BB%CE%B9%CE%BF> (Last Accessed: 14/03/2018)
- (120) The invisible train (2004-2005), Available at: https://studierstube.icg.tugraz.at/invisible_train/ (Last Accessed: 11/01/2017)
- (121) The Most Popular Programming Languages of 2016 (Last modified: 18/08/2016), Available at: <https://blog.newrelic.com/2016/08/18/popular-programming-languages-2016-go/> (Last Accessed: 18/03/2018)
- (122) Tinmith – mobile outdoor augmented reality (Last Updated: 2016), Available at: <http://www.tinmith.net/> (Last Accessed: 11/01/2017)
- (123) U.S. Naval research Laboratory, Available at: <https://www.nrl.navy.mil/itd/imda/research/5581/augmented-reality> (Last Accessed: 11/01/2017)
- (124) Venice Charter (Last Modified: January 2017), Available at: https://en.wikipedia.org/wiki/Venice_Charter (Last Accessed: 11/01/2017)
- (125) Videoplace (Last Updated: November 2014), Available at: <https://en.wikipedia.org/wiki/Videoplace> (Last Accessed: 11/01/2017)
- (126) Virtual Reality (Last Modified: January 2017), Available at: https://en.wikipedia.org/wiki/Virtual_reality (Last Accessed: 11/01/2017)
- (127) Wikitude (Last Modified: January 2017), Available at: <https://en.wikipedia.org/wiki/Wikitude> (Last Accessed: 11/01/2017)
- (128) Wikitude: Un logiciel de navigation en réalité augmentée (in french) , Available at: <http://sensebook.reaco.fr/blog/realite-augmentee/wikitude-un-logiciel-de-navigation-en-realite-augmentee/> (Last Accessed: 11/01/2017)