

Development of an iOS, Augmented Reality for disaster management

Nikolaos Barmpas Zachariadis

2020
Department of
Physical Geography and Ecosystem Science
Centre for Geographical Information Systems
Lund University
Sölvegatan 12
S-223 62 Lund
Sweden



Nikolaos Barmpas Zachariadis (2020). Development of an iOS, Augmented Reality for disaster management.
Master Degree Thesis, 30/ credits in Master in Geographical Information Science
Department of Physical Geography and Ecosystem Science, Lund University

Development of an iOS, Augmented Reality for disaster management

Nikolaos Barmpas Zachariadis

Master Thesis, 30 credits, in Geographical Information Sciences

Supervisor Dr. Ali Mansourian

Department of Physical Geography and Ecosystem Science, Lund University

ACKNOWLEDGEMENTS

I would like to thank Lund University, Sweden and most specifically the Department of Physical Geography and Ecosystems Analysis, for allowing me to participate and complete the Master's Programme in Geographical Information Systems (G.I.S.). I would also like to thank my supervisor Ali Mansourian for his help, guidance, and comments on my manuscript. Without him, I would have never accomplished the completion of my Thesis. I would also like to thank Mahdi Farnaghi and Dan Metcalfe for reviewing my Thesis and providing me fruitful comments. I would also like to thank David Tenenbaum for his support through my studies. Finally, I would like to give special thanks to my partner Adriana for her patience and her support while pursuing my studies.

ABSTRACT

Disaster management is a field, where any kind of information is essential for planning and rescue operations. The availability of real-time data on a portable device, such as a smartphone, can be very useful for emergency workers, during rescue operations and citizens as well. Augmented mobile reality is a system where the users can see overlaid data and visuals in the real environment through their smartphones or tablet screens.

In this study, we have developed and tested a mobile application on the iOS platform, with disaster specific software functionalities. The mobile application has been designed to serve disaster management teams during the rescue operations for better coordination and information exchange. The system uses a combination of Geolocation services and Augmented Reality to present in real-world information, such as Points of Interests, citizen's health status or previous building's states so that will ease the work of the rescue teams and help citizens find shelter.

One of the primary aspects of this study is to find a suitable method to measure the accuracy of the iPhone 8 Plus render augmented information in reality. For that, we conducted 144 tests for different distances and azimuths to verify the accuracy of the augmented information on the real world.

The results of this study show that the iPhone 8 Plus can render objects with very high accuracy and can provide accurate information to the user. Finally, the app has been presented for two scenarios, during a disaster response phase, to have an overall picture, whether the application could prove satisfying results or not, and how useful it could be for future use.

Keywords: Geography, GIS, Accuracy, Augmented Reality, App, Geolocation, Mobile, iOS, iPhone

Table of Contents

ACKNOWLEDGEMENTS	iv
ABSTRACT	v
Table of Contents	vi
List of Tables	vii
List of Figures	vii
List of Graphs	viii
1. Introduction	1
1.1. Aims and objectives	2
1.2. Research questions.....	3
1.3. Thesis structure	3
2. Literature review	5
2.1. Virtual and Augmented Reality	5
2.2. Mixed Reality.....	6
2.3. History.....	8
2.4. Techniques	9
2.5. MAR –Augmented Reality on mobile devices	11
2.6. AR Applications.....	12
2.6.1. Military	12
2.6.2. Medical and health.....	12
2.6.3. Entertainment.....	13
2.6.4. Archaeology and tourism	13
2.6.5. Disaster management and related work	14
2.7. Development of MAR in iOS devices	18
2.7.1. Programming language	18
2.7.2. ARKit.....	19
2.7.3. SceneKit.....	20
2.7.4. 3D Modelling.....	20
3. Methods	21
3.1. Study Area	21
3.2. Data.....	21
3.3. Methodology	22
3.4. Step 1 - Define data and functionalities needed.....	24
3.5. Step 2 – Technology Selection.....	24
3.6. Step 3 - Design Application’s Architecture	25
3.7. Step 4 - Development and Analysis.....	27

3.7.1. Implementation	27
3.7.2. Core location, Haversine formula, and distance calculation	28
3.7.3. Position implementation	29
3.7.4. Setup the AR environment	31
3.7.5. App Main Interface	33
3.8. Step 5 - Debugging and accuracy tests	33
3.9. Demonstration of the utilization process	35
4. Results	37
4.1. iPhone 8 Plus Rendering Accuracy results	37
4.2. Disaster Response Scenarios.....	46
4.2.1. Geolocation Visualization (AR) Scenario.....	46
4.2.2. Infrastructure Visualization scenario	48
5. Discussion	51
5.1. Challenges.....	51
5.2. Advantages – Disadvantages	51
5.3. Future work.....	52
6. Conclusion	55
References	57
Appendix	61
Series from Lund University	65

List of Tables

Table 1 GIS apps in different phases (Source: Cova Thomas, 1999. “GIS in Emergency Management.”).....	17
Table 2 Data category, type, sources, target	22
Table 3 Technologies	25
Table 4 Aggregated Accuracy results from all tests (meters) for every POI.....	37
Table 5 Rendering results for eight different POIs – 5 different distances (meters) – 2 different azimuths (degrees)	39
Table 6 Rendering results for eight different POIs – 3 different distances (meters) – 2 different azimuths (degrees)	41
Table 7 Rendering results for eight different POIs – 3 different distances (meters) – 2 different azimuths (degrees)	42
Table 8 Average Positional Error and Standard Deviation for eight different POIs –2 different azimuths (degrees)	43

List of Figures

Figure 1 Milgram and Kishino's Mixed Reality on the Reality-Virtuality Continuum (Parveau and Adda 2018).....	7
Figure 2 Study Area - Rizari Park.....	21
Figure 3 Methodology flowchart	23

Figure 4 App Architecture	26
Figure 5 Gravity and heading alignment: a real-world reference frame has fixed directions (Apple 2019b)	32
Figure 6 App’s main interface	33
Figure 7 Rendering example from the app’s prototype interface	34
Figure 8 Render Sphere Object at 8th POI	43
Figure 9 Render Health Status of User in Real world	47
Figure 10 MapView of POIs.....	47
Figure 11 Render POIs.....	47
Figure 12 Render building Info / Damage %	48
Figure 13 Render Previous Infrastructure State.....	48
Figure 14 Render Previous Infrastructure State Different Angle	48
Figure 15 Setup the location manager	61
Figure 16 Calculate bearing	61
Figure 17 Convert radians to degrees	61
Figure 18 Matrices functions	62
Figure 19 Transformed location.....	62
Figure 20 Setup AR view.....	62

List of Graphs

Graph 1 Rendering results for 8 POIs – Azimuth 0° Degrees	40
Graph 2 Rendering results for 8 POIs – Azimuth 90° Degrees	41
Graph 3 Rendering results for 8 POIs – 1m-5m-10m meters – 45° & 90° degrees Azimuth ..	42
Graph 4 Boxplot of Different Meters.....	45
Graph 5 Boxplot of Different Azimuths	45
Graph 6 String Graph of Different Azimuths	45
Graph 7 String Graph of Different Meters.....	45

1. Introduction

Disaster can be any event of natural hazards such as earthquakes, hurricanes, or tsunami. During an emergency, people get injured, and some die whereas infrastructures shutter, and predominantly, there is a significant economic impact on society. Disaster management is the process of handling a disastrous situation before, throughout, and after its occurrence. It includes 4 phases: 1) mitigation, 2) preparedness, 3) response, and 4) recovery. Disaster management has been used to avoid the occurrence of hazards and their impact. It can help to reduce human losses and achieve an adequate response. Furthermore, it can also help to provide a rapid and durable recovery after the event.

During a disaster, accurate decisions need to be taken to avoid human and material losses. Hence, decision-making is a critical task in disaster management because it arises in all management activities before, during, and after a catastrophic event. On the contrary, indecisions can lead to significant losses, above human life's or infrastructures. Consequently, decision-making strives to reduce the impact of a disaster.

Presently, data describing the extent of damage from disaster events are available from sources, suchlike governmental departments, or organizations. The role of this data is vital in the decision-making process and several operations during disaster handling. It can also help managers make precise decisions during and right off a disaster.

Augmented reality is a technology, that can influence disaster management and, nowadays, is accessible right through our mobile devices. AR solutions let users visualize POIs (e.g. supermarkets or pharmacies) on reality through their mobile devices and navigate the real world without the need to carry a map. Furthermore, rescue teams can also use AR to navigate through destroyed buildings and see information on their device's screen. This information could be: a) number of rescued citizens in a location, b) people's geolocation, c) status of infrastructures, d) previous states of the foundations before the disaster, e) gas or water pipelines locations, f) interior or exterior of a building, etc. Present mobile devices accommodate better positional and orientational accuracies, due to their sensors, and that makes the technology suitable for such applications.

Augmented Reality is a unique piece of technology that we could use to handle a disaster situation and help disaster management. However, there are a lot of prospective problems regarding this technology. Such issues include: 1) gathering the data and 2) measuring the accuracy of the augmented information. The first one is one of the most fundamental problems in a disaster situation, and until today, there is not a perfect solution for it. In this study, we focus on measuring the accuracy of the augmented information of the iPhone 8 Plus and provide the results of those tests.

1.1. Aims and objectives

Aims

The aim of the Thesis is to develop an augmented reality application for iOS mobile devices to use it in the field of disaster management. The application target's group are emergency workers, such as search and rescue teams, along with citizens who get involved in a disaster situation. The emergency workers will use the app in an earthquake emergency to efficiently improve their mission by integrating augmented reality and mobile devices. Whereas the citizens will use the app to find shelters or submit their location and status for the rescue teams to be able to help them much faster. The app's main idea is that in an earthquake, the infrastructures could be damaged or destroyed. Accurate information such as the location and use (e.g., residential, school, chemical storage) of the building are necessary for emergency workers to be able to provide better services. For example, If an emergency worker knows that a building is a chemical factory, the first thing he should do is to secure the building in order not to be any chemical dispersion in the atmosphere, that can harm the citizens. The rescue teams can also render 3D infrastructures in reality on top of the destroyed ones. Apart from the above, the citizens can use the app to navigate to Points of Interests (POIs) for supplies. Finally, they can access their geolocation and submit it along with their health status to the app, so the emergency workers can render this information in reality later on.

Objectives

The aim of this Thesis can be achieved, following the targets below:

- Design and develop the architecture of a system, supporting data for AR for disaster management.

- Design and implement a mobile app, to render 3D data on iOS mobile devices.
- Measure the positional accuracy of rendering 3D data in the AR environment.
- Discuss challenges for the practical implementation of the system, based on the experience of this study and suggest solutions.

1.2. Research questions

This research is carried out to propose a new way of handling disastrous situations. The idea is to use Augmented Reality (AR) technology, to create a link between “before” and “post” disaster situations, to provide emergency field workers with better information about the disaster situation, victims, possibilities, etc. The research questions that will be answered in this study are:

- Research Question 1: “How to render 2D and 3D models in mobile AR?”
- Research Question 2: “How to use real-world coordinates to render 2D and 3D models in the real world using mobile AR?”
- Research Question 3: “How to measure the accuracy of the rendering points?”
- Research Question 4: “Where this application can be used in Disaster Management?”

1.3. Thesis structure

This Thesis has six chapters. 1st chapter is the introductory chapter. Here we describe the background, aim, objectives, and research questions of the Thesis. Section 2 discusses the history of augmented reality; it’s applications and its connection to GIS through findings from the literature review. Then, chapter 3 apprise of the selecting study area and data along with the methodology that followed, to implement the aim of the Thesis. Besides, section 3 clarifies the steps of the methods in order, pioneer steps one to five, to develop a geolocation – AR system as complicated as, the one presenting in this study. Chapter 4, presents the results of this study, conducting the measurements from the rendering tests. There is also a demonstration of the system for two different disaster response scenarios. In Chapter 5, we discuss the challenges, advantages and disadvantages, and future work regarding the application. Finally, chapter 6 contains the conclusion of the Thesis.

2. Literature review

2.1. Virtual and Augmented Reality

Virtual reality (VR) is the milestone of the concept and idea of Augmented Reality (AR). Tönnis (2010) have supported this broaden opinion by stating that AR is the logical progression of VR. Virtual Reality (VR) is a technology that has been used by many applications such as video games, space, and medical training for many years. With the use of VR, the user is surrounded by a virtual environment. In other words, a VR system transfers the user from the real world to an interactive computer-generated one, where he can navigate the artificial world while interacting with virtual features or items (Aukstakalnis et al. 1992).

Augmented reality is a complex structural system that integrates the real world with computer-generated virtual objects, where the user remains in the real world and augment his experience with virtual elements. Although augmented reality generally is about visual augmentations, other means of development are thinkable, such as sound or tangible devices. Azuma et al. (2001) proclaimed that AR systems would be able to include a complete set of augmentation applied to exploit all people's senses; such as touch, hearing, smell, and taste, as well as balance. Researcher Azuma (1997) was the first to explain to academic society what AR systems are. According to him, AR systems should include the following three characteristics:

1. "Combine physical and virtual reality"; this characteristic is fundamental because the physical world can display digital information. User experience a material world, where real and virtual objects co-occurred together.
2. "Interactive in real-time"; it allows the user distinctively to interact in real-time with the real physical world, observe information and change them or create new ones. For example, if the user changes the display information in the device, then his viewpoint in the environment will change as well.
3. "Registered in 3-D"; AR virtual objects are using the tracking system to allocate the real-world location and visualize 3D in the real world using the three-axis coordinate system. The three-axis coordinate system includes latitude (x), longitude (y), and altitude (z).

Although Azuma's AR definition is eminent, researchers Milgram and Kishino (1994) gave a new term for augmented reality (AR) in 1994. They characterized AR as being a fusion of real and virtual data within a real-world environment. The real world and a virtual environment are at the two ends of this continuum with the middle region, called Mixed Reality. Augmented reality lies near the real-world purpose of the spectrum with the predominant perception, being the real world augmented by computer-generated data. Besides the above, Milgram also created a new scientific term, the one of Augmented Virtuality. Milgram stated that Augmented Virtuality points out systems, that are mostly synthetic with some real-world imagery added, such as texture mapping video onto virtual objects (Milgram and Kishino 1994).

Klinker, in his work "Teleoperations and Virtual Environments, Special Issue on Augmented Reality" in 1997, expressed in words, that users can work individually or collectively, experiment with computer-generated information and interact with a mixed environment, like AR, in a natural way. Mahoney, two years later, added that augmented information is usually virtual, digital, or synthetic information. The real world, he explained, must be matched with the virtual in position and context to provide an understandable and meaningful view (Liarokapis 2007). The present-day, we have the option to add sounds and images to the environment by using AR. These applications, called haptic, are dependent on touched AR and have been used in medical fields (Khademi et al. 2012). To conclude, Van Krevelen and Poelman (2010) stated that, AR could be implemented through various other types of wearable or handheld technology, apart from Head-Mounted displays (HMD). Such devices can be smartphones, GPS devices, smartwatches, and large screens.

2.2. Mixed Reality

According to Paul Milgram and Hirohisa Kishino "Mixed Reality" is the universe between the real world and the virtual world (Fig. 1). It is the blend of real and virtual worlds along the reality virtuality continuum comprising of Augmented Reality (AR) and Augmented Virtuality (AV) technology. In Mixed Reality, virtual objects are being overlaid into the real world with the use of AR, while real objects are being captured by AV and they are being rendered into the virtual environment. In Mixed Reality the "real world" is the real-world environment surrounding the user, usually referred to as the "outside world". The virtual world represents a virtual environment that can be

computer-generated. In this world, the user can be immersed without any outside interaction and that is part of Virtual Reality. The augmented content becomes very limited when this world gets very close to the "real environment" and the user's experience tends to be less immersive (Parveau and Adda 2018).

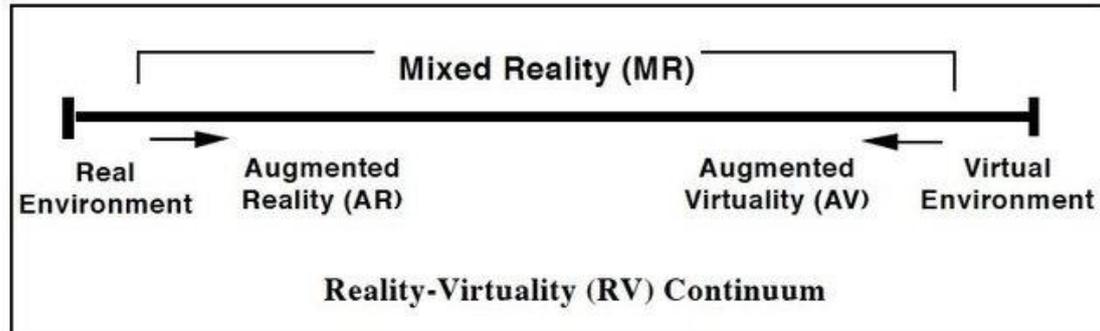


Figure 1 Milgram and Kishino's Mixed Reality on the Reality-Virtuality Continuum (Parveau and Adda 2018)

In Mixed Reality, the user can see the real objects through a display device. The objects are first being observed from the user and then they are being transferred into a digital model and resynthesized. In simple terms, the virtual objects are being simulated to create a viewable representation on the display device. MR allows the user to observe the interaction of the real objects along with the virtual ones in the mixed reality environment (Juraschek et al. 2018).

Speicher et al. (2019) tried to give a specific definition for Mixed Reality in their work "What is Mixed Reality?". According to them, experts had a lot of difficulties to specify what constitutes MR, with several contradicting statements. Overall, they described Mixed Reality with terms such as 1) "strong" AR (i.e. AR with more capabilities), 2) marketing/buzzword, 3) AR plus full immersion, i.e. the possibility to combine AR and VR in the same app or on the same device, 4) "MR is the continuum", the combination of real and virtual, 5) MR is bound to specific hardware (e.g. HoloLens), 6) MR as "the poor man's version of AR.". Many experts argued that the term Mixed Reality has been used for marketing purposes, whereas others supported that "only academics understand the MR spectrum" and stated that there is not enough information about MR conceptually (Speicher et al. 2019). Still, today is very challenging to give an exact term for Mixed Reality. Each individual's perception is different but the majority understand that Mixed Reality is a blend of Augmented and Virtual Reality.

2.3. History

Augmented reality is considered the technology of the 21st century, but its roots go back to the 1950s, when cinematographer Morton Heilig built the first AR concept Sensorama in 1962, bringing the viewer into the movie by engaging all his senses. Sensorama was mechanical, rather than digital. The user could not interact with the display, but he could see a film with added sound, scent, wind, and vibration effects (Mandal 2013; Carmigniani and Furht 2011). Ivan Sutherland, in 1968, constructed the first virtual and augmented reality system; a computer-based Head Mounted Display (HMD). HMD uses an optical see-through head-mounted display to visualize 3D-graphics, which is tracked by either a mechanical tracker or an ultrasonic tracker. Because computers didn't have enough processing power those times, the only things the system could display in real-time, were simple wireframe drawings (Arth et al. 2015; Sutherland 1969; Carmigniani and Furht 2011).

In 1975, Myron Krueger invented VIDEO PLACE; a computer system that would coordinate the movement of a graphic object with the user's actions. Cameras caught users' silhouettes and projected them onto a screen, where they were able to interact with the virtual objects (Mandal 2013). At the same time, Bell Helicopter experimented with analog systems that would augment the vision of helicopter pilots, to be able to land in the dark using infrared cameras. During the 1970s and 1980s, virtual reality research developed with the aid of military funding, whereas in the early 1990s, Boeing coined the term "Augmented Reality" to describe his research on mounting cables in airplanes (Janin et al. 1993). During the late 1990s, students at Columbia University developed MARS3, which took AR out of the lab, to the outdoor environment, and after that, more advanced applications appeared, and research widened, into various areas of studies (Feiner et al. 1997). In the early 2000s, AR research got a lot of attention, and the development of the ARToolKit library supervised the creation of new custom hardware and commercial products. That allowed developers to be able to implement the AR technology in their applications (Palmarini et al. 2018; Maad 2010). To conclude, since the technology necessary for AR experience, became available for masses, through iOS and Android platforms, there has been a massive growth in AR commercial market.

2.4. Techniques

There are two AR techniques called optical see-through (OST) and video see-through (VST). In the first one (OST), we see in real-time the reality augmented with virtual objects through some transparent screen. Whereas in the second, (VST), the truth is captured by a camera and is combined with the virtual objects by a computer to be presentable on a screen. Both techniques require; a) a computer, that uses a camera to capture, or just read reality, b) GPS to determine the location of the user and c) some screen that can display the virtual layer. Nowadays, we can find all these necessary components on our smartphones and tablets. That's the reason those devices are ideal for commercial uses of augmented reality.

Video-see-through

Video see-through (VST), is the most common technique in the AR systems and has been used mainly by hand-held portable screens, which are equipped with a camera. Video see-through is used to capture the view of the real world, by the device's embedded camera or an external camera. To be able to implement that technique, the user must use a head-mounted camera in conjunction with a head-mounted tracker, to capture the real-world scene and gather all the necessary scene input. Then, a computer digitally combines the video stream from the real-world view, that user collected, with generated virtual objects (augmentations) and displays both a virtual layer and a layer of reality on the user's screen. So, the user doesn't see the truth but a recorded version of it. The viewpoint position is given to the graphics system to render the virtual objects in their proper place.

One of the significant advantages of such a system is that it has access to both; real and synthetic images, so it can manipulate them in space or time to register them. This capability can assist the process of tracking the head position and thus leading to a more accurate registration. Another advantage is that the video display of this system's type is typically high-resolution, so it is possible, to render highly detailed virtual objects (Roland et al. 1994; Geng et al. 2014).

On the other hand, the video see-through technique has disadvantages. The main drawback of the video see-through technique is that the merging of a real scene and graphics output is necessary to display the augmented view to the user. This merging process adds significant delays to the system. This delay translates into; a) time lag, b) low update rate, and c) poor resolution seen by the user, which reduces his feeling of

presence. In other words, it creates a delay between users' eyes and other senses. As a result, it can cause multiple complications, including physiological ones for the observer and create all kinds of discomfort, such as headaches and nausea. Unfortunately, this disadvantage can not be avoided but only get minimized. Finally, the video see-through is depending on what the monitor displays. If the monitor cut off, then the user will not be able to see anything (Takagi et al. 2000; Lee et al. 2015b).

Optical-see-through

Optical-see-through (OST) technique, in contrast to video-see-through, projects digital content on the screen, while the whole surrounding area is still visible, allowing the user to see the real world in real-time. The view of the real world remains nearly intact while being augmented by a computer-generated image. OST requires a head tracker that uses accelerometers, gyro meters, and GPS to determine the location of the user's head. The computer processes the data and creates a digital layer, which is then displayed on a transparent screen. These screens, called Head-Up Displays (HUD), are half-mirror displays, which provide a direct view of the surrounding physical world. This feature is especially useful, in some AR applications, that the user may need to have a picture of the physical world, even if the device is off. In systems using this display, the augmented content is projected on the half-mirror screen, to present both; augmented and real-world view to the user. Head-Up Displays combine the images the user is viewing in the actual environment, along with graphical depictions of data or imagery, the HMD receives from the AR system (Rolland and Fuchs 2000).

The advantage of this technique is that the observer has the chance to see through the display, directly to the world surrounding. So it is not required to film the environment before that. Because the optical-see-through technique allows the user to see the real world, augmented with virtual objects, there is an inevitable delay between the real world and the displayed virtual objects. Especially when accuracy is required, this can be problematic. In contrast to video-see-through, optical-see-through techniques cannot adjust this delay, because they don't capture reality through a camera. Furthermore, virtual objects might have problems with the brightness, because OST lets the real-world shines through it. As a result, video see-through techniques can blend virtual objects more accurately into the image of the real world (Azuma et al. 2001; Milgram and Kishino 1994).

Finally, the OST technique is broadly being used to our devices today, such as smart-phones and tablets. Those are equipped with powerful processing units, high-resolution

screens, and cameras, making them capable of optical see-through AR. A modern application of OST AR is IKEA Place, which lets the user add a piece of virtual furniture to the real-world desired space, whereas Google and Microsoft have used this technique to their products; Google Glass¹ and Microsoft HoloLens², respectively.

2.5. MAR –Augmented Reality on mobile devices

Mobile Augmented Reality (MAR), is a new technology based on Augmented Reality, that extends and enhances the user experience of the mobile device. It can be used in mobile devices such as smartphones, tablets, gaming consoles, and military Head-Up Display (HUD) (Junwei et al. 2016). Mobile augmented reality became very popular in mobile devices in 2017 because Apple announced its support to advanced Augmented Reality with its A11 bionic neural engine and Apple's ARKit (Apple 2019a). In the meantime, Google brought its new ARCore framework to the Android ecosystem by partnering with Samsung.

In general, augmented mobile reality is a system where the user can see overlaid data and visuals in the real environment. This overlay can be implemented through smartphones or head-wearable devices because they are equipped with camera and position or orientation tracking systems. Nowadays, mobile phones are so powerful that they can be compared to our laptops or desktop computers from past decades and include core pieces of technology, that are mandatory to deliver an MCAR experience.

Those are:

- a) Fast mobile Processor (CPUs) with graphics hardware (GPU unit); the processor needs to process user input, video images, and simulate applications, and the GPU needs to generate generating virtual images.
- b) Large screens that create the AR view by combining virtual images with pictures of the real world.
- c) High-Resolution cameras that we use to capture live video images. Those are being used for AR tracking and for overlaying virtual imagery onto the video images as well.
- d) Powerful communication hardware, cellular, and wireless networking for collaboration and sensors such as GPS, compass, and gyroscopes for tracking techniques (Alem and Huang 2011).

2.6. AR Applications

A wide range of fields such as tourism, entertainment, marketing, logistics, archaeology, medical and health, manufacturing, GPS mapping systems, and mobile phone applications have used AR technology. Below, we introduce some of them highlighting the field where they have used augmented reality.

2.6.1. Military

The military has used augmented reality for many years to provide soldiers with information about targets and dangerous areas on the battlefield and outside of it. On the other hand, Head-Up displays have been broadly used in training to augment the real world's view to pilots, navigate troops and vehicles and prepare soldiers for combat situations (Behringer et al. 2002; Hicks et al. 2003). BARS stands for a battlefield augmented reality system, that has been developed to recognize difficulties in processing and handling map data for the dismounted soldier in urban combat environments. The system uses a wearable computer, a wireless network system, and a tracked see-through HMD. Then computer-generated data is accurately correct overlaid and aligned with the environment on that display and may also report known hostile positions, such as snipers or artillery (Livingston et al. 2008).

2.6.2. Medical and health

Augmented reality has extraordinary application possibilities in medicine. Surgeons analyze and plan operations, using image data of patients. The image data arrives from various medical sensors, such as magnetic resonance, ultrasound, or computed tomography imaging. Therefore, operation support systems have been developed, using these sensors, to assist surgeons in the surgery through live overlays and real-time x-ray visions of internal anatomy. As a result, the use of AR reduces the risk of operations and make them less time consuming and reduces the cognitive load of doctors in the operation room (Tano et al. 2012; State et al. 1996). Roberts et al. (1986), executed the first medical augmented reality system, superimposing preoperative computerized tomography (CT) data, in the proper position, scale, and orientation.

Medical students used AR technology to practice surgery in a controlled environment. Medical AR provided a useful tool for medical guidance, training, education,

procedure, and workflow. AR helps to project anatomical information or image-guided surgical landmarks onto the patient, which provides image guidance during surgical procedures decreasing risks associated with long procedure times (Cheung et al. 2010; Lee et al. 2015a). At UNC-Chapel Hill, a research group worked on a system, that lets a physician see directly into a patient, by using ultrasound echography imaging, were, on the other hand, MIT University ran a project on image-guided surgery. The results of the last one led to the development of a navigation system, that surgeons are using to sort the average length of surgeries in hospitals, where have been applied, such as the Brigham and Women's Hospital in Boston, United States (Grimson et al. 1996).

2.6.3. Entertainment

Entertainment is often found to be the most potent force, to push technology forward, and this is likely to happen in the AR field as well. Games, using augmented reality, have appeared in several forms. An example of an AR game is an AR chess, where each user runs an instance of the shared chess application and uses the tracking data communicated via the two multicast groups, to render the users' actions correctly. The application migration features described in the section, allow the mobile user to enter the workspace and join the running chess game anytime (Reitmayer and Schmalstieg 2001). Another mobile game, called "Pokémon Go", introduced in 2016, is a mobile location-based social game and is by far the most popular AR game, involving the physical activity of the gamer in the real world with potential and documented health benefits (Paavilainen et al. 2017).

2.6.4. Archaeology and tourism

AR can be applied to visualize architects' designs of bridges or buildings, that are about to be constructed on a particular site, historic buildings, long torn down, in their original location, or reconstructions of archaeological sites. The European sponsored project ARCHEOGUIDE is the first Augmented Reality based Cultural Heritage on-site guide developed by Vlahakis et al. (2002). ARCHEOGUIDE provides an augmented tour of ancient Greece and visualizes how buildings initially looked before and after the war period. Its aim is also to reconstruct-visualize a cultural heritage site in augmented reality and let visitors view, obtain information, and learn about ancient architecture and customs. So, tourists can see the actual location along with reconstructions of both

buildings and people. Similarly, Kim et al. (2009) used AR to provide an immersive experience of the historical scene, reflecting the needs of tourists improving the quality of the cultural tour.

2.6.5. Disaster management and related work

Disaster management is an experimental environment, where the user can test new methods for disaster prevention and reaction planning. Mobile augmented reality (MAR) can give a solution to support the first responder by supporting and guiding them, through superimposed information over the real world, to initiate the relief operation. Several systems are already addressing the use of AR, in support of emergency response such as the Augmented reality system for earthquake disaster response that Leebmann (2019) describes in his work “An augmented reality system for earthquake disaster response.” Its purpose is to overlay different invisible disaster-relevant information, e.g., people trapped in the rubble of collapsed buildings and overlay it with the real environment. This kind of information can help emergency workers, to plan their actions on-site using the augmented information to find and rescue these humans. Experts from different areas can also use this system to share their data to work together. The system consists of a backpack rack that holds two Trimble 4800 GPS, a pole for the GPS-antenna, a tripod, and an aluminum suitcase mounted on the backpack. The briefcase contains all connectors, rechargeable batteries, and a portable computer. A tripod that is mounted on the backpack carries the camera and an IMU measuring the orientation of the camera. A second IMU is attached to the retina display. Leebmann (2019) conclude that the design of such a system has many challenges since the system integrates different methods such as mapping, photogrammetry, inertial navigation, and differential GPS and he stated, that “the ARS might be applied in other phases of the disaster cycle like disaster impact, relief, rehabilitation, reconstruction, mitigation, and preparedness”.

Brunetti et al. (2015) presented a wearable AR collaborative system, called “Smart Augmented Field for Emergency (SAFE)”. This system’s purpose is to deliver information and skills to first responders and operators to face health, security, and managerial issues during an emergency response. That way, the system empowers rescue missions on emergency scenarios and train the rescuers, providing an innovative tool based on the idea of the augmented field. SAFE consists of mission-oriented AF

middleware; a kind of operating system, and the domain-specific applications to expand involved entities such as rescuers and victims. Each rescuer is carrying a mobile AR-oriented device, that allows the system to bring up all messages and information related to the rescuer's current real field of view. When a rescuer finds out a victim involved in the disaster, he has to identify and evaluate patient health status to perform first aid treatments. For the identification process, SAFE uses an active mobile device that is associated with each patient. This device consists of a memory unit dedicated to the real-time storage of patient health status and a smart sensor management layer, in which some software agents continuously sense the patient's vital signs and update the local state. Rescuer's intelligent assistant agents use the data provided by the tag, to help rescuers to reduce action's performing time and complete them successfully. In conclusion, this system can provide the best outcome actions and operations for rescuers, only if the right health severity code assigned to patients and their health status keep monitored (Brunetti et al. 2015).

Erra et al. (2018) developed a MAR application, to visualize geo-referenced sensors measurements and integrate simulation data of hydraulic and hydrological models on mobile devices. This application, developed in the Metaio framework and runs on mobile devices equipped with a webcam and sensors such as a gyroscope, a compass, and a GPS. This platform: (a) increases the visual perception of the real world merging additional information with the natural scene in real-time, (b) allows mobile workforces to reach easier the most critical areas subject to flooding, and (c) rapidly decides on the level of flood protection. The mobile application can access a wide range of measurement sensors, deploy various visualizations, consult different data in real-time, and integrate simulation results explicitly developed to create a shared awareness in the field teams. The user of this application can capture a scene on their mobile device and receive details for a point of interest (POI). The user can also import new POIs into the database, and export data and metadata to other programs and software. Thanks to this application, the users are aware of the potential danger around them in emergencies. This allows speeding up both the decision-making processes and the crisis responses. Erra et al. (2018) research conclude that augmented information on various vulnerability scenarios during the emergency phases, helps both technical and non-technical operators to quickly intervene, containing or preventing secondary disasters, thus reducing deaths and injuries, and limiting the resulting economic losses and social disruption. From the above, we can see that as long as the hardware technology

advancement increases; such platforms can grow in the environmental monitoring sector and possibly trigger a broader range of opportunities to improve the existing toolset and explore new potential uses.

Apart from MAR, Geographic Information Systems (GIS) are also crucial for the disaster management process because we can use them in all phases of the procedure; from prevention to immediate reaction. With GIS we can create maps, including information such as the population's distribution demographic data in an affected area or analyze the geospatial aspects of this area. The problem with GIS is when the spatial and non-spatial data are geographically dispersed and stored in heterogeneous databases. To resolve this issue, Stoimenov et al. (2007) used a platform called GeoNis. GeoNis is responsible for creating a common interoperability framework for the data sources. In that way, we can store the geo-data in databases that have the same structure format, and they can become available to both public and emergency workers. As a result, emergency management becomes more intuitive and efficient because everybody can participate in the process and can help decision making.

Tiwari (2015), developed a 3D-GIS enable disaster management architecture to simulate disasters and help overall the disaster management process. The use of such a system is essential because it covers all the different stages of the disaster management process and can help the user to create different scenarios. We need 3-dimensional geoinformation from various sources and sensor systems that have to be integrated, to: a) improve monitoring, b) reduce damage and c) facilitate better estimates of damages and losses. For that purpose, the 3D-GIS engine; the main component for the development of this system, has been used to collect all the above information and make all the computations, to produce the 3D model. Once the 3D model is ready, then it is applying to every phase of disaster management. In this stage, GIS analysis tools are used to overly, define mapped attribute information to calculate expected losses, and a 3D modeling engine is modeling results continuously. This engine then generates various disaster scenarios than can be simulated by the emergency workers to plan their actions. Finally, the system becomes available in all possible formats, e.g., desktop or web, including relevant standards and services.

We can see that GIS is critical in the field of disaster management. While studying Cova (1999) scientific paper about GIS role in disaster management, we point out that there are many different applications developed based on GIS for each phase of the disaster response process. The table below visualizes those (Table 1):

<i>Spatial question</i>	<i>Application arena</i>	<i>GIS examples</i>
What is the spatial extent of the impacted area, and what are the expected losses?	Disaster preparedness and response systems	Earthquake (Routh 1990) Oil spill at sea (Friel et al. 1993) Dam failure (Kragt et al. 1993) Forest fire (Goulstone-Grunland et al. 1994) All hazards (Linz and Bryant 1994) Chemical spill (Lu and Xiang 1995) Flood (Fulcher et al. 1995) Radiological accident (Guber et al. 1995) Hurricane (Corbley 1995)
Which are the best routes available for emergency vehicle deployment?	Emergency vehicle routing	Automatic vehicle location (Zografos and Douligeris 1991) Fleet management (Karimi 1993)
How can we identify good evacuation routes and develop evacuation plans?	Evacuation planning	Path analysis (Dunn 1992) Simulation modeling (Silva et al. 1993) Evacuation vulnerability (Cova and Church 1997)
Which populations need to be evacuated?	Evacuation zoning	Chemical spill (Cartwright 1990) Hurricane (Corbley 1995)
How can we immediately locate an emergency caller on a map?	Enhanced 911 (E-911 emergency calls)	All emergencies (Fitzgibbon 1993)

Table 1 GIS apps in different phases (Source: Cova Thomas, 1999. "GIS in Emergency Management.")

Scientific research related to the combination of AR and GIS in disaster management is limited, but below, we present two significant experimental projects regarding this new technology. The first one is an AR-based geovisualization app that is used to simulate the movement of people evacuating a building. The simulation shows all the possible paths the evacuees can follow inside the building to come out of it. The simulation contains a 3D terrain, derived from a digital elevation model (DEM). The 3D ground provides the walkable path and allows the emergency workers to navigate virtually, by using AR, the exterior of the building along with the evacuees. In that way, the application enables the user to analyze the simulated movement of people as they evacuate the building, and this makes them capable of implementing a solution to find the best path the evacuee can follow. So, this prototype proves that AR can be used to visualize the mass evacuation of people from a building and can provide training to emergency management workers utilizing the application for a real evacuation event (Lochhead 2019).

The other project has been carried out by Luchetti et al. (2017), where they developed an Augmented Reality Crowd-Mapping System for Civil Protection and Emergency Management. The system integrates augmented reality (AR), crowd-mapping (CM),

social networks, the Internet of Things (IoT), and wireless sensor networks (WSN) by exploiting technologies and frameworks of Web 2.0 and GIS 2.0. Its purpose is to store real-time data, analyze them, and produce georeferenced information data to a real-time online geodatabase, so they are available online. These data can then be used from civil protection authorities to manage the operations related to disaster management. The AR technology is crucial to make the information accessible for quick interpretation of geo-data to emergency workers and the public on the scene of an emergency. The AR capability allows users to visualize unique assets useful for civil protection scenarios. The system is composed of a geo-server, a mobile application with AR and an analytics dashboard, and its main architecture components are a) The GeoData Collector; a server responsible for data collection from social networks and data analysis. It uses geodata sources such as geo-databases (e.g., PostGIS or Esri Spatial Database Engine), geo-files (e.g., Geography Markup Language or Shapefile) or web mapping services (e.g., Web Map Service WMS or Web Processing Service WPS), to improve the quality of the information, b) The Mobile Application; responsible to visualize AR-relevant details about the environment or event and c) The Analytics Dashboard; enables the analysis of the games using both plain text and custom visualizations on maps.

While the marriage of AR and GIS is quite a new implementation, a few attempts to create useful apps using this technology have been carried out. This combination is the future of where the world is leading, and soon, this technology will be able to help not only the disaster management process but also will find applications in health, education, and many other sections of life.

2.7. Development of MAR in iOS devices

2.7.1. Programming language

Swift is a programming language, developed from Objective-C and created by Apple; it is the programming language that we used for this study. Objective-C developed based on C programming language in the 1980s, and its main focus is object-oriented programming style. That's the reason Apple used it to build macOS as well as iOS applications. However, since 2014, Apple decided that Swift should be the primary programming language of the Apple ecosystem.

Apple developed Swift by improving and amending the old-outdated Objective-C. Swift has the compiler named Low-Level Virtual Machine, which generates a

completed product without losing flexibility. It reduces the needed files and code for the development of Swift. Swift also has both the power of low-level programming languages like C or C++ and the smoothness of high-level languages like C# or JavaScript. Moreover, Swift is light and packs a pre-defined library. Swift has a runtime debugger which can work spontaneously with developers to track and alert all errors. As a result, Swift is more natural, easy to read, and maintain and user-friendly (Apple 2019d; Hillyer 2016).

The Swift standard library comprehends the base layer of functionality for writing Swift programs and includes a) data types, b) data structures, c) functions and methods, d) protocols (Reboucas et al. 2016). The Swift Core Libraries have been written in such a way so they can provide a rich set of APIs. Those APIs are consistent across the various platforms that Swift supports. That's how developers can write code that can be portable from smartphones to tablets and vice versa, which Swift helps.

These libraries provide a higher level of functionality as compared to the Swift standard library. Finally, Swift provides a side feature called Playground; a simulator which offers a test environment where developers can write and test their code without building the entire program, and this finds a prominent use in education where the students can be trained to develop apps since a young age (Hoffman 2017; Solt 2015).

2.7.2. ARKit

ARKit is a framework developed and released by Apple in September 2017 for iOS 11. ARKit utilizes the camera together with motion sensors to help developers create AR content within mobile applications. By combining device motion tracking, camera scene captures, and advanced scene processing, ARKit has the excellent capability of creating augmented reality experiences for iOS devices. Due to having consistent performance, it is required to use Apple A9 and later processors to perform ARKit.

For world tracking, ARKit uses Visual-Inertial Odometry (VIO) technology, which uses a combination of the input video and the motion sensor of the device. As with Feature Points in ARCore, ARKit finds notable features in the image of the scene and traces them based on changes in their positions and obtained information from motion sensors. ARKit can recognize flat horizontal planes in the captured scene, where user can add and place virtual objects on, but vertical surfaces as well. ARKit can also

recognize images such as paintings or brochures and use this as an anchor point to render AR content.

ARKit uses a camera sensor to estimate the intensity of light in the device's surrounding environment and apply the correct view. Some limitations and weaknesses that ARKit is suffering from are that is depending on featured points that detected with analyzing the image. That means that the low quality of the input image may affect the accuracy of the tracking, whereas low light intensity and darkness in the scene may affect tracking feature points (Apple 2019a).

2.7.3. SceneKit

SceneKit is a 3D framework developed by Apple, which “combines a high-performance rendering engine with a descriptive API for import, manipulation, and rendering of 3D assets”. The API available in SceneKit is relatively high-level compared to similar frameworks such as Metal and OpenGL that require the user to implement in precise detail the rendering algorithms that display a scene. Finally, the only thing, SceneKit needs from the user is: a) to describe the content of the scene he wants to have in a view and b) define the actions or animations he wants the content to perform (Apple 2019d).

2.7.4. 3D Modelling

For this study, we have created 3D models. To do that, we used Google Sketchup because is an easy software to use. Google Sketchup can render and manipulate 3D models by transforming, translating, scaling, rotating them, and covers five depended on API areas: 1) scenery, 2) camera, 3) landscape, 4) animation, and 5) user model. SketchUp has almost no preliminary setting windows or analogs, and the user may keep dimensions and properties during projecting or later. That's the main reason why SketchUp is handy to first create our idea and then edit it. It is best for the first stages of the project when it is necessary to see and explain the primary purpose, where there is not enough time. Finally, SketchUp also can export the project to Collada (.dae) files for using those in the development of AR applications.

3. Methods

3.1. Study Area

The application developed for this research should be working globally, because it uses the iPhone's GPS sensor to receive the location of the user through satellites and render the objects in the real world, using real-world coordinates. However, the selected study area for testing and evaluating the results of the application, is Athens, Greece. Athens is the capital of Greece, covered by the Attica region and has the largest population in the country. Most specifically, our tests for rendering the sphere objects to test the iPhone's sensors' accuracy took place in Rizari park.

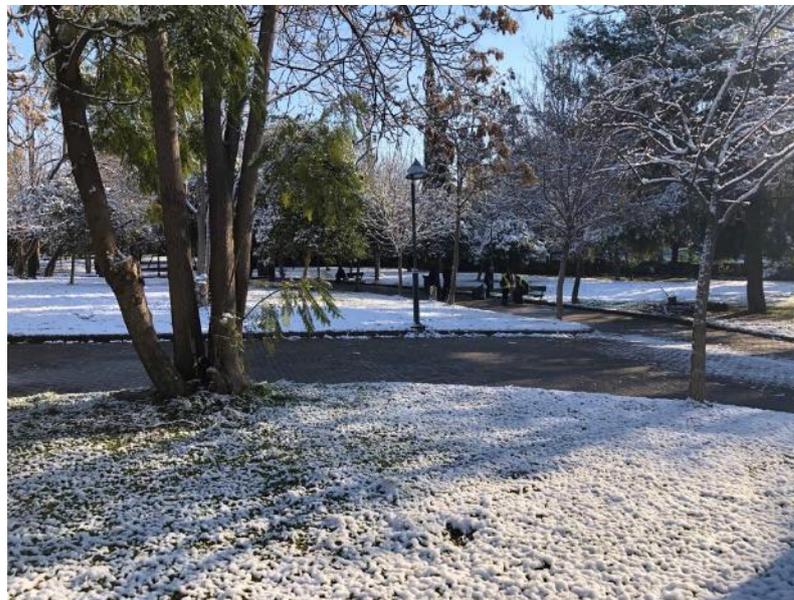


Figure 2 Study Area - Rizari Park

Rizari Park has beautiful sculptures that can we can use as POI (Points of Interest). We have used 8 different sculptures-POIs to get their coordinates from and use them as references for our tests. Finally, due to its size, it is possible to navigate through it and utilize different locations with different world coordinates where we carry our checks.

3.2. Data

The table below (Table 2) illustrates the data categories, types, and sources that have been used in the development of the application. Additionally, the target list describes the use of the data for this project. All the data mentioned below are needed to achieve the aim of the Thesis and indicates the origins of these data acquisitions. Generally, the data acquisition method is one of the most critical issues when it comes to disaster

management. Therefore, we must take into account that emergency response organizations' role is vital. They are responsible for collecting and providing data, such as building attributes or rescued people, even though this study doesn't state any information related to this matter.

Category	Data	Data type	Data source	Target
Geometry object	SCNNode	3D object in geometries of spheres and lines	-Swift programming language - Scene Kit	-Rendering tests to measure the accuracy of the rendering points
Building geometry	3D building(exterior) 3D building(frame)	.dae	-To be designed by the author in Google SketchUp by measuring buildings dimensions and their attributes in the real world	-Render the previous state of infrastructures
Building attribute	Building damage status Building type Building attributes Address	.txt (Text)	-Hellenic cadastre and user's input	-Render building attributes to help the rescue teams
List of POIs	Name and address	.txt (Text) - UILabel	-Google's Places API	-Render POIs to help people navigating the real-world to find shelters e.g. hospitals

Table 2 Data category, type, sources, target

3.3. Methodology

Figure 3 shows the process of the application implementation and the way all the technological elements are connected. The following sub-sections describe the application development process in 5 steps. Step 1 is to list and describe the app's functionalities and the data we need for its implementation. In Step 2, we define the technologies along with the frameworks that we used to achieve the aim of this Thesis.

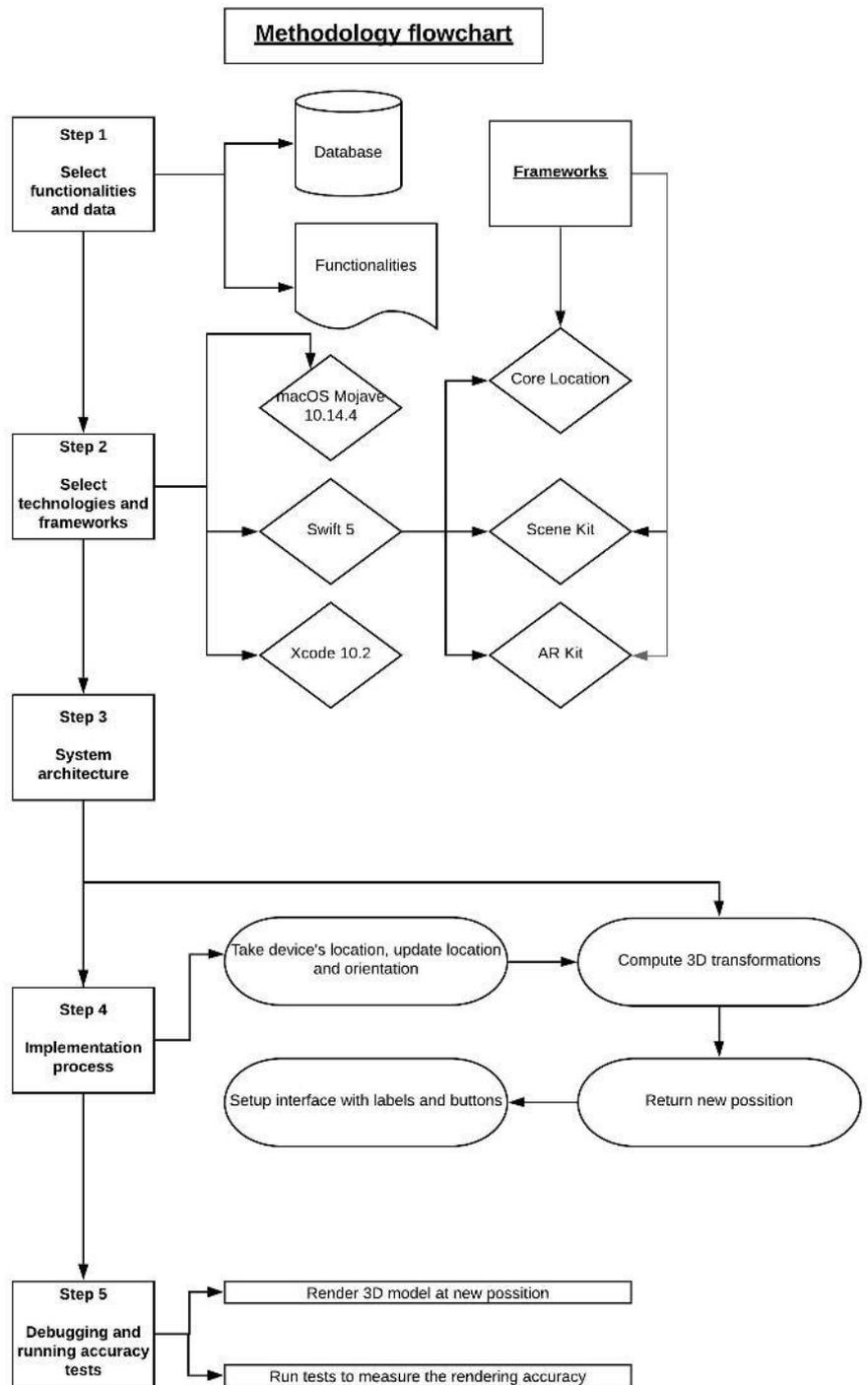


Figure 3 Methodology flowchart

Step 3 illustrates the design of the system's architecture. Step 4 describes the implementation process to perform all the necessary operations for rendering a 3D model or any information in the real world using real-world coordinates. Finally, Step

5, includes the tests to measure the accuracy of the iPhone's sensors in the study area and make sure the app is functioning correctly.

3.4. Step 1 - Define data and functionalities needed

Step 1 is vital because it is the starting point for developing the app and design the workflow along with the targets to implement through our development of the application. The main idea of the app is to superimpose virtual (digital) data and information in the real environment, to improve user's immersive in a disaster situation. In this first step, we have to decide:

- a) which kind of data, we are going to use in our application,
- b) the type of this data, e.g., text (.txt) or 3D model data (.dae) and the source of this data; where we are going to take this data from, either we will create it ourselves,
- c) what kind of functionalities to develop that can help both rescue teams and citizens on the field.

Below are summarized the functionalities of the app that have been implemented:

- Render the location and health status of a person.
- Render Points of Interests (POIs), through the camera's phone in the real world.
- Render infrastructure's status using a text form (e.g., the damage percent of a building).
- Render the use of infrastructures (e.g. chemical factory, company, etc.).
- Render previous infrastructure states in reality.

3.5. Step 2 – Technology Selection

At Step 2, we describe the technologies that have been used for the implementation of this application (Table 3). To develop this application, we chose mobile platform iOS and iPhone 8 Plus as the target device. They were several reasons behind this choice. First, iPhones have one of the most accurate GPS sensors and gyroscopes in the mobile market, and that's the primary reason, that makes them perfect devices, to render 3D objects with proper accuracy in reality. Second, Apple has put a lot of effort into

implementing AR kit; augmented reality’s framework, and Core Location; the frame of the location, in contrast to the Android ecosystem that has fewer resources and is not so advanced as Apple’s one.

Mac OS Development Environment		
OS	macOS Mojave 10.14.4	
Programing Language	Swift 5	
IDE	Xcode 10.2	Frameworks: <ul style="list-style-type: none"> ▪ AR Kit ▪ Core Location ▪ SceneKit
Mobile device		
iOS Device	iPhone 8 Plus	Ram 2 GB
3D Model		
Colada	.dae	SketchUp

Table 3 Technologies

Apart from the above, another critical reason to support the choice of iOS is that there is a higher number of potential users and a higher potential for economic benefit, in case the author wants to publish the app in the App store. Additionally, it is also essential to mention that the ownership of a MacBook Pro and an iPhone 8 Plus, made the choice of using iOS and iPhone more native, compared to the Android ecosystem.

3.6. Step 3 - Design Application’s Architecture

The mobile application’s architecture consists of three main layers (Fig. 3), and its services. The View layer is the main screen, where the interactions with the different users take place. Together with the Background Services and the Handling layer, they act as the primary input points to the system. The Handling layer, which consists of three modules, receives the events that take place in the Main Storyboard. The View Controller is responsible for interaction with the Background services and the communication between the different layers.

The Main Storyboard is hosting necessary user interface components, facilitating the possible interactions with the users. It displays the:

- AR View,

- the user's location as obtained by the Background Services,
- 3D models,
- and the POIs presented as labels on the real world.

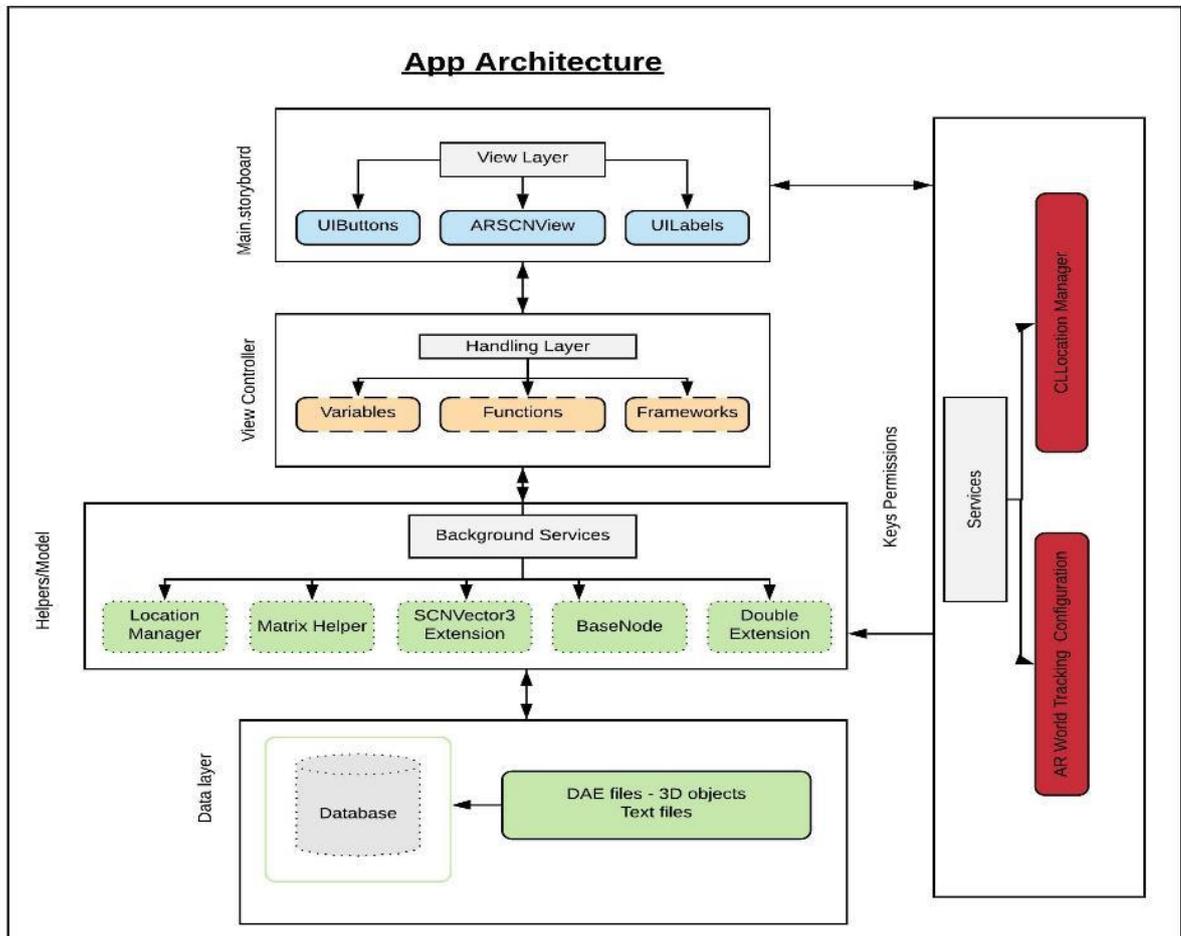


Figure 4 App Architecture

The Handling layer is the most important one because interactions, exchange of information, and synchronization pass through this layer. Background Services contain modules that are responsible for:

- a. gaining the GPS coordinates and parse them to the matrix helper to transfer them,
- b. creating 3D objects,
- c. interacting with the database.

The action starts on the View layer and flows to the lower components. Background Services receive and send information to the database as required by the other layers. The Data Layer is responsible for storing the data needed for the application. The Location Manager module is the component responsible for obtaining the user's locations. It uses all the available phone's sensors to collect the most accurate position. The View Layer requests location updates to the Location Manager and receives sites containing latitude and longitude. The Matrix Helper is the component responsible for serving the other Background components for rendering the information on the proper location in the real world. Finally, when the application is in the background and aims to provide control over the location service, and the camera will use the Key Permissions.

3.7. Step 4 - Development and Analysis

3.7.1. Implementation

Step 4 describes the implementation of the application. First, to develop the app, we had to learn Swift -Apple's programming language. For that, we spent a significant amount of time. Furthermore, we learn how to use Core Location and AR Kit in the development of the application to successfully achieve the aim of the Thesis. Apple Inc. offers its tool to develop iOS applications, called Xcode, and we have used this one for the development of the application. Finally, the 3D model for 3D objects, e.g., building exterior, has been created using 3D design software, SketchUp. The following factors should be defined to implement the system:

- user's current location (geographical coordinates) from iPhone's GPS and user's input,
- gyro sensor (yaw, pitch, roll) to get the direction of mobile device,
- 3D object's rotation, scale, and translation.

First, the application obtains the user's location, through the phone's GPS and illustrates it to the app interface on a text label. Then the user can input the coordinates of the place he wants to render information for; this place called destination location. Once the app identifies the user's location, then it will use those coordinates for the mathematical calculations to compute the new coordinates to render the information or

the object on the real world. Those new coordinates must be calculated every time because we are changing direction or moving away from an object.

The application is also capable of providing the distance from destination location in meters. Like that, we know how far we are from the Augmented information/object in the real world. Afterward, the app performs the bearing calculation, which will help us create a rotation transformation to position the model in the right direction and distance from the renderer. Then there is a lot of mathematics taking place, described in section 3.7.3, to render the objects or information in the right place on reality. In a brief description, we first use matrices to rotate the 3D objects, scale and transform them. Finally, the transformation moves objects from the world origin (0, 0, 0) to the destination location, to render the object in the proper position in the real world.

3.7.2. Core location, Haversine formula, and distance calculation

First, we must calculate the distance between the coordinates of the user's location and the destination's world coordinates. The first ones can be accessed from the location manager, using the Core Location framework. In general, a framework in the Swift programming language is a dynamic library or a package of libraries that multiple applications can use. Core Location framework provides services that determine a device's geographic location, altitude, and orientation, or position relative to a nearby iBeacon device. The structure gathers data using all available components on the device, including the Wi-Fi, GPS, Bluetooth, magnetometer, barometer, and cellular hardware.

In this study, we used the `CLLocationManager` class to configure, start, and stop the Core Location services. In our app, the "Start" button starts the Core Location services and receives the coordinates, whereas the "Stop" button terminates them. `CLLocation` is a class of the `CLLocationManager` that we used to track the user's real-world coordinates position. Being able to monitor that location is fundamental to our project. To get the best possible position, we used the `kCLLocationAccuracyBestForNavigation`, as the accuracy property. This property provides the highest possible accuracy that uses additional sensor data along with all the available onboard hardware, including Wi-Fi, GPS, Bluetooth, magnetometer, barometer, and cellular equipment to gather a device's geographic location, altitude, orientation, or position. This option takes the most power and takes longer to set the

place, but it is the most accurate method we can use, and this is crucial for our accuracy tests. However, in this study, we used only the 3G cellular network of the phone and not the Wi-Fi connection.

It is also essential to consider location services as sensitive data due to Apple's Privacy policy. For that reason, the app gives the user the ability to choose whether the app can use his location or not. However, if the user doesn't allow the app to use his site, then the app cannot work, because the last one will not receive any coordinates. The coordinates are mandatory to render the information on real-world locations. To warn the user for the use of his location data, we have used a delegate for checking whether the user allows the use of his location or not. This delegate option is called "When-in-use authorization." The commissioner is working only while the application is running. If the app is not running, then the location manager's delegate will never execute.

The next step is to calculate the distance. To calculate this distance, we will use: a) the user's position and b) a method called "distance", that is part of the class "CLLocation.". The above formula uses the Haversine Formula, that uses two different latitude/longitude pairs of values and calculates the distance between them. So, it measures the distance between the location in the current object and the value in the location parameter. The distance is calculated by tracing a line between the two points, that follows the curvature of the Earth and measuring the length of the resulting arc. The arc is a smooth curve that does not take into account altitude changes between the two locations.

3.7.3. Position implementation

Now we need to calculate the bearing (the angle) between the two different latitude/longitude pairs of values using the mathematical equation and setting the coordinates as bellow:

$$bearing = atan2(x, y)$$

where:

lat1 = latitude of users location

lon1 = longitude of users location

lat2 = latitude of destination location

lon2 = longitude of destination location

$$x = \sin(dLon) * \cos(lat2)$$

$$y = \cos(lat1) * \sin(lat2) - \sin(lat1) * \cos(lat2) * \cos(dLON)$$

Figure 16 of the Appendix visualizes the implementation of calculating the bearing between the two world coordinates. Another thing to consider is that for the matrix transformation, we have to use radians instead of degrees as angle units, implementing the following equation:

$$radians^\circ = \frac{180^\circ}{\pi(\pi)^\circ \sim 3.14} \text{ (Webb 2019)}$$

After we calculate the bearing, we must use a $[4 \times 4]$ matrix: $[x, y, z, w]$ to calculate the orientation (rotation) of the 3D model in the best possible way. The fourth dimension (w), called “projective space,” and the coordinates in the projective space are called “homogeneous coordinates.” When w equals 1, it does not affect x , y , or z because the vector is a position in the area. When $w = 0$, the coordinate represents a point at infinity (a vector with infinite length), which is used to express a direction.

To get objects in the right direction, we need to implement a rotation transformation. The vector around the origin (0,0,0) is rotating through the rotation transformation, using a given axis and angle. In this application, we are interested in placing the 3D object to our current location and rotate it based on the compass. So, we always need to calculate the orientation as if it was the first time to switch the model. That’s because once we turn the model a certain angle, then the following rotations will be done relative to that angle. So, to render information or an object in another place, we must recalculate the angle (ARKit and CoreLocation, 2019). To implement the rotation transformation, we used the following mathematical equation:

$$\begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ (Webb 2019)}$$

To calculate the transformation matrix, first, we must calculate; scaling, rotation, and translation matrices. Then, to estimate the transformation matrix, we must combine the other three models as below:

$$\text{Transform} = \text{Scaling} * \text{Rotation} * \text{Translation}$$

For that calculation, we used SIMD (Single Instruction Multiple Data - simd_mul) operations. Apple tacked on a library called “simd.” We used this library to implement

SIMD (single-instruction, multiple-data) arithmetic for scalars, vectors, and matrices. This built-in library gives us the ability to do 2D, 3D, and 4D vector and matrix operations across various processors on OS X and iOS. Finally, we calculated the `positionFromTransform` (Appendix Fig. 16) where that takes as input the location transform pattern and gives us back the x, y, z values from the 3rd column. Those are the values that we have to use to place the marker of the destination location. Like that, we can place/render our object in the real world using its real-world coordinates.

3.7.4. Setup the AR environment

To experience AR through our phone, we need to build a single session object using ARKit. ARKit can track a device's position and motion relative to the real world. To run an ARSession, we need to configure it. For that, we used `WorldTrackingConfiguration`, a subclass of the abstract `ARConfiguration` class, for tracking the device's orientation, position, and detecting feature points like surfaces that are recorded by the camera. The `ARWorldTrackingConfiguration` can augment the user's view of the world around them through the device's back camera.

The `ARWorldTrackingConfiguration` class tracks the device's movement with six degrees of freedom (6DOF). Those are the three rotation axes (roll, pitch, and yaw), and three translation axes (movement in x, y, and z). Due to this kind of tracking, a virtual object can appear to stay in the same place relative to the real world. Even if the user tilts the device to look above or below the purpose or moves the device around to see the object's sides and back, the object remains in the same position. In general, `ARWorldTrackingConfiguration` (Appendix Figure 10) is responsible for connecting the physical world which the user and the phone exist in, with the virtual coordinate space generated by the phone based on the camera and motion data (Apple 2019c).

Apart from the above, to render any object or information in real-world coordinates through the camera of the iPhone, we must first track the device's orientation and position. For that purpose, we need to config the world alignment property on the AR session. This property defines how the AR session interprets the `ARFrame`'s motion data on a 3D coordinate mapping system that keeps track of the world and build the Augmented Reality experience.

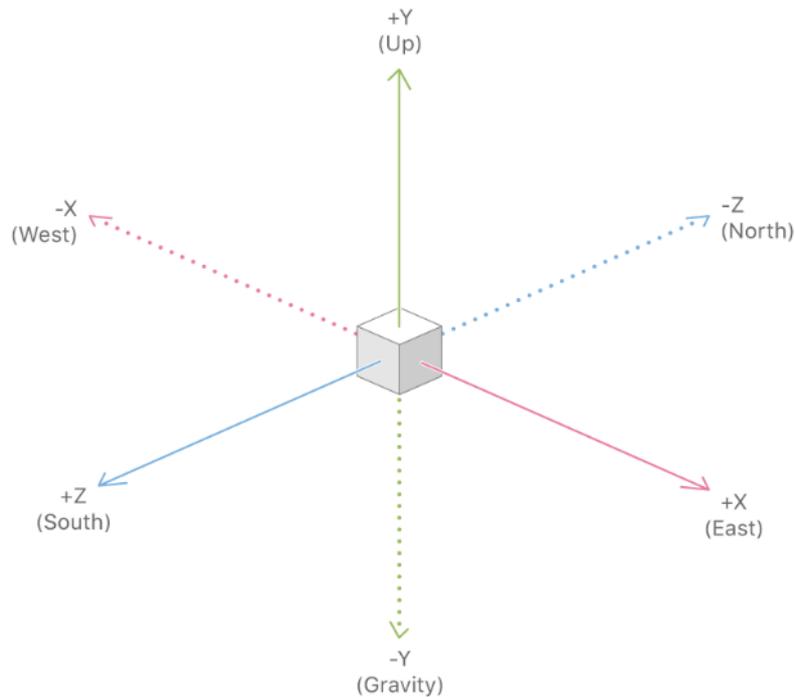


Figure 5 Gravity and heading alignment: a real-world reference frame has fixed directions (Apple 2019b)

By setting the adjustment to gravity and heading (Figure 5), ARKit aligns the y-axis parallel to gravity, and the z and x axes orient towards compass heading. The initial location of the device contains the origin. In gravity and heading alignment, the real-world reference frame has a fixed direction. The y-axis matches the direction of gravity as detected by the device's motion-sensing hardware. The x- and z-axes match the longitude and latitude directions as measured by Location Services. While this is accurate most of the time and fixes the instructions of the three coordinate axes to real-world routes, its precision is not incredibly high, because the location of the coordinate system's origin is still relative to the device, matching the device's position as of when the session configuration is first to run (Apple 2019b).

The last thing to do is to create our node class and render the object in the real world. We'll use subclass SCNNode and give it a title property, which is a string. We will add to it as well an anchor property, which is an optional ARAnchor that offers the ability to track positions and orientations of models relative to the camera, so it updates the status of the SCNNode. Additionally, we'll give our SCNNode class, a location property which is a CLLocation. Then we create a SCNNode of geometry sphere with radius 0.1 and the color blue, which we will use for our accuracy tests. In the same way, we created our sphere object; we will draw lines that are going to be gas pipelines and text items that will visualize information regarding the number of

inhabitants, location of a room in a building, etc. Finally, we will import the 3D models in our project to render them on reality through the app.

3.7.5. App Main Interface

The interface is what the user sees and interacts with, so this part is very important. In this application, the main interface will be as simple as possible because we will use the app during the disaster response phase. For doing that, it is essential to implement methods for inputting the coordinates, shows the location and status of inhabitants or infrastructure and buildings and render 3D objects or text. Finally, the main view accommodates 4 different buttons that link to different views/functionalities:

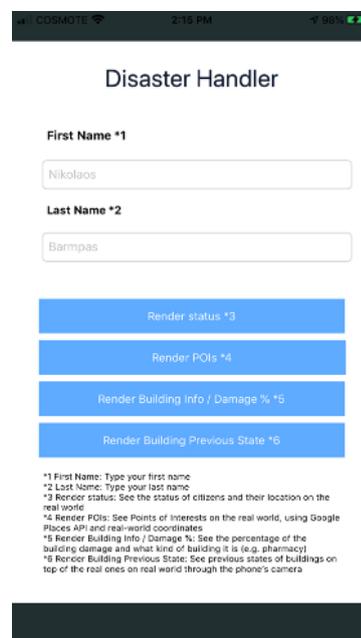


Figure 6 App's main interface

- “Render Status”; the user can input his status and his geolocation in a text format. Then the rescue worker can render this in the real world,
- “Render POIs”; the user can render POIs in the real world,
- “Render Building Info / Damage %”; the user can render the stored information about damage percentage of building and info about the kind of building,
- “Render Building Previous State”; the user can the previous state of a building.

3.8. Step 5 - Debugging and accuracy tests

In Step 5, we will test the accuracy and functionality of mobile AR, by running tests, using a 3D sphere object at the study area. Since this study uses sensor-based AR, the accuracy of data registration depends on the efficiency of mobile sensors. Therefore, it

is essential to analyze that mobile AR has enough capacity to improve safety assessment in the field by running the app.

The target of those tests is to measure the error (distance in meters) from the rendering point to the real one, so we get an overall picture about the capability and the accuracy of the iPhone 8 Plus to render information in the real world. In general, the app has been implemented, in such a way to visualize the data on reality, with the highest accuracy.



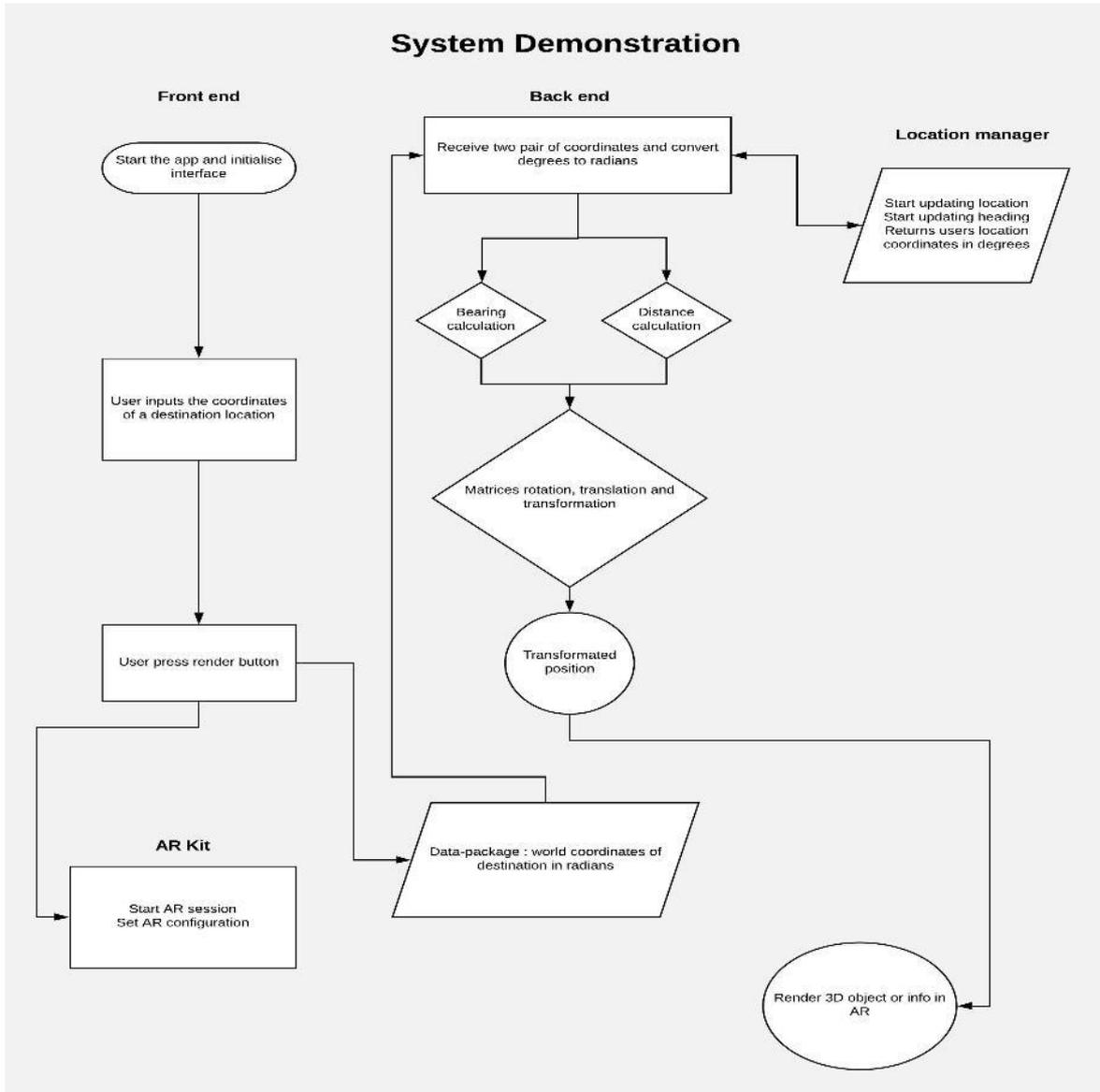
Figure 7 Rendering example from the app's prototype interface

The first step was to create the sphere object (Fig. 7), in the app prototype, and then create the buttons, where we were able to submit the x and y coordinates from the GPS. Further, we manage to render the sphere object on those coordinates, by typing the last ones in the text fields “x” and “y” coordinates. When we presented the sphere object, in reality, we used an application, called Tape Measure, to measure the distance between the real-world object and the rendering sphere object. Tape Measure uses Augmented Reality as well, to measure the distance between two objects, through the camera of the phone. Finally, the overall evaluation of mobile AR performed, based upon these tests.

3.9. Demonstration of the utilization process

The app has 2 target groups; the people on the field that can be injured or trapped and the emergency workers. When the first ones start the app, a pop-up window asks them for permitting the app to use their current location and their device's camera. Then they can tap into the "Render Status" button and see the coordinates of their location on the labels "Latitude" and "Longitude" and on top, they can see the real world through the camera. The users can submit their real-world coordinates along with their status to the app (in a different text field next to the "Submit Health Status" button). To do that, they have to copy-paste the latitude to the text field called "x coordinate" and similarly, do the same with the longitude to the field called "y coordinate". Once they do that, they should type on the text field next to the "Submit Health Status" button, their condition, e.g. injured, trapped, safe and submit it along with their geolocation. The ones that are on the field can render POIs using the button "Render POIs" in the app's main interface. This button will bring on the main camera view all the POIs that are surrounded in the area and the user can navigate towards them using his phone.

On the other side, the rescue teams can use the app to render the information that citizens have submitted earlier. By tapping at the button "Render Status", they can see the location of the people in need along with their status. So they can find them and help them immediately. Finally, rescue teams can input the coordinates of a building to take information regarding its kind/damage % and see its previous state, by tapping the buttons "Render Building Info / Damage%" and "Render Building Previous State" respectively.



Flowchart 1 System Demonstration

4. Results

4.1. iPhone 8 Plus Rendering Accuracy results

In this study, we carried out two types of tests:

1. During the first one, we tested the app for 0m, 1m, 5m, 10m, and 15m (meters) rendering distance from the POIs and direction to the north; 0° and 90° degrees for each distance separately.
2. Whereas in the second one, we tested the app for 1m, 5m, 10m (meters) rendering distance from the POIs and direction to the north; 45° and 180° degrees for each distance separately.

The rationale behind those tests was to determine the horizontal accuracy of the GPS sensor of the iPhone 8 Plus, to see whether or not the device is capable to provide useful and accurate results to the users in a disaster situation. To take appropriate results, we had to test each rendering object in different azimuths-locations for the user. During the tests, we discovered that the longer we stood in a place and kept the phone stable, the better accuracy we get. This factor is the most common that affects the iPhone's GPS accuracy. It is called occupation time and is required to achieve the best and most efficient results. For some POIs, we noticed that after 1-2 minutes, the GPS stopped receiving updated coordinates. That means that the GPS reached the highest possible efficiency and the most accurate results.

POIs	Aggregated Accuracy results from all tests (meters)
1	6.1
2	1.7
3	4.9
4	5.3
5	4.7
6	3.2
7	4.7
8	4.5

Table 4 Aggregated Accuracy results from all tests (meters) for every POI

In general, the overall accuracy was about 5.23 meters, which is very close to the professional GPS. 1st POI presented the worst result of 6.1 meters whereas 2nd POI had the best accuracy at 1,7 meters. The second best in terms of accuracy was the 6th POI

with 3.2 meters. 3rd, 5th, 7th, and 8th POIs provided an estimated accuracy in the range of 4.4 - 4.9 meters.

Below Table 5 visualizes:

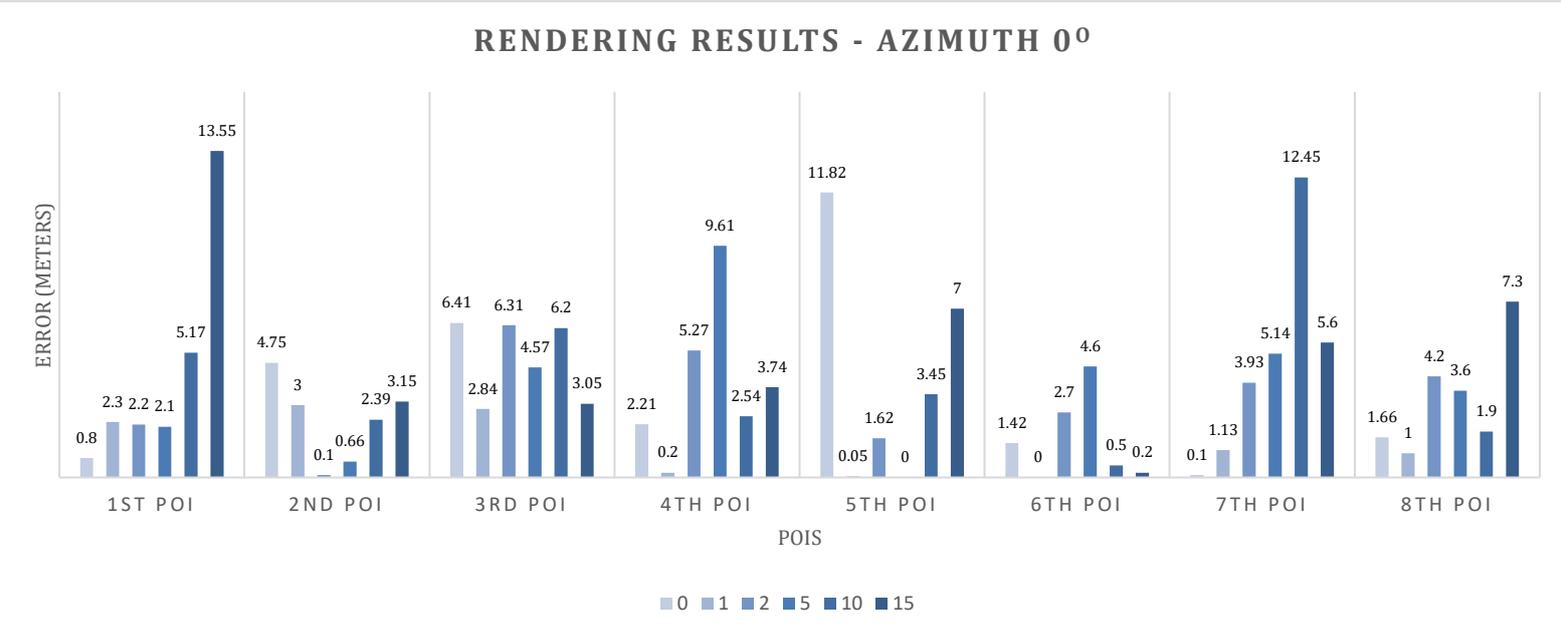
1. the different POIs,
2. the coordinates of each rendering point,
3. the distance of the renderer from the rendering point,
4. direction to the north in degrees and
5. under the azimuth columns, we can see the error in meters. The error is the distance of the rendering object to the real world one.

Results of Testing					
POI	Coordinates	Rendering Distance from point (m)	Error (meters)		POI
			Direction to the north (Azimuth in degrees)		
			0°	90°	
1 st POI	X: 37.9754889 Y: 23.7464209	0	0.80	6.75	
		1	2.3	5.23	
		2	2.20	5.07	
		5	2.10	6.63	
		10	5.17	0.80	
		15	13.55	8.20	
2 nd POI	X: 37.9755478 Y: 23.7468358	0	4.75	2.48	
		1	3	0.58	
		2	0.10	1.03	
		5	0.66	1.14	
		10	2.39	0.20	
		15	3.15	0.50	
3 rd POI	X 37.9746368 Y 23.7477968	0	6.41	7.08	
		1	2.84	6.29	
		2	6.31	4.8	
		5	4.57	9.05	
		10	6.2	6	
		15	3.05	4.85	
4 th POI	X 37.9745671 Y 23.7473878	0	2.21	2.08	
		1	0.2	1	
		2	5.27	2.79	
		5	9.61	8.28	
		10	2.54	7	

		15	3.74	11.66			
5 th POI	X 37.9749764 Y 23.7464510	0	11.82	2.85			
		1	0.05	4.45			
		2	1.62	11.20			
		5	0	3.75			
		10	3.45	5.89			
		15	7	15.07			
6 th POI	X 37.9753930 Y 23.7471675	0	1.42	2			
		1	0	0.1			
		2	2.7	0.05			
		5	4.6	3.58			
		10	0.5	1.1			
		15	0.2	1.82			
7 th POI	X 37.9745409 Y 23.7475053	0	0.1	7.62			
		1	1.13	6.23			
		2	3.93	7.73			
		5	5.14	2.41			
		10	12.45	4.06			
		15	5.6	3.4			
8 th POI	X 37.9744354 Y 23.7471283	0	1.66	6			
		1	1	0.30			
		2	4.2	0.15			
		5	3.6	0.70			
		10	1.9	3.4			
		15	7.3	1.9			

Table 5 Rendering results for eight different POIs – 5 different distances (meters) – 2 different azimuths (degrees)

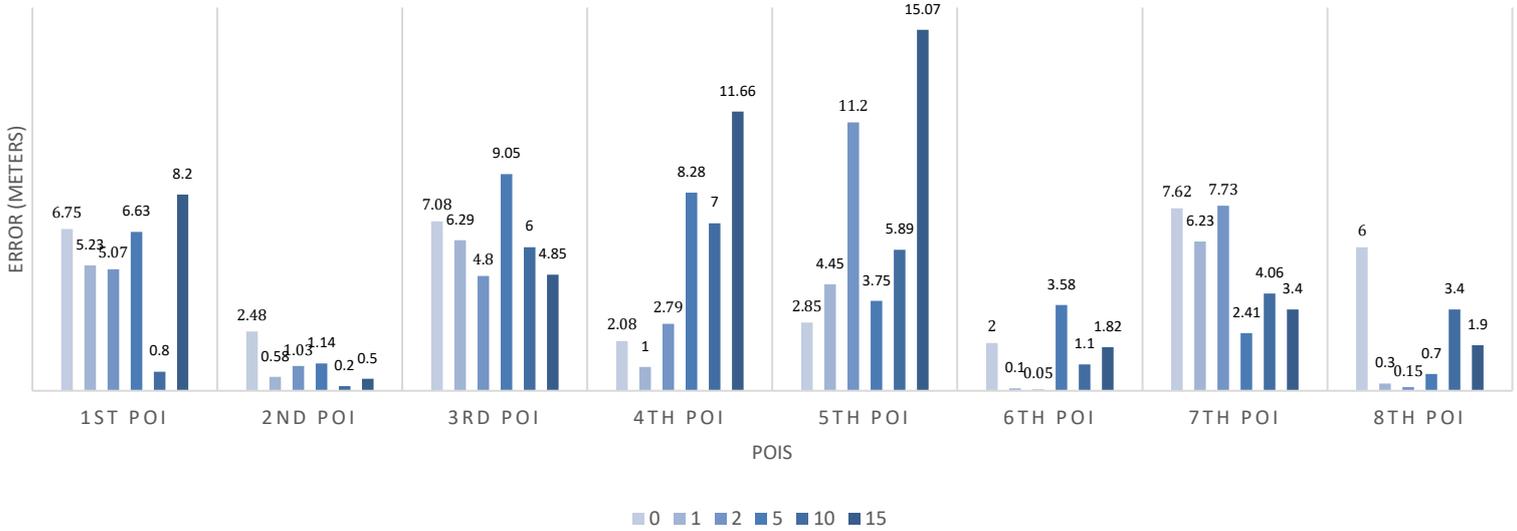
Graph 1 illustrates the rendering results for azimuth 0° degrees across the different POIs. The 1st POI produces the highest error of 13.55 meters (Graph 1), whereas the 5th POI shows the minimum error, 0 meters. Overall, the results for 0° degrees direction to the north, show that the typical error is between 0-5 meters. Whereas in some cases, it is over 6 and below 8 meters. Among all 2nd, 3rd, 4th, and 8th POI illustrated the best results on this test. Finally, the 5th POI shows a failure of 11.82 meters for 0 meters rendering distance, where the 7th POI provides a similar result for 10 meters rendering distance.



Graph 1 Rendering results for 8 POIs – Azimuth 0° Degrees

Graph 2 results are different from Graph 1. Graph 2 provides information about the rendering results with direction to the north 90° degrees. According to the graph, the range of error is 0-9 meters, whereas 3 POIs have errors higher than this range. Again, accuracy is good enough. If we examine carefully the graph, we notice that most errors are between 0-5 meters. The 5th POI shows the highest error again; 15.07 meters, while the rendering distance is 15 meters and a failure of 11.20 meters for rendering distance 2 meters. Here 2nd, 6th, and 8th POI provide the best results. Finally, the lowest error is 0.05 meters for the 6th POI at distance 2 meters. From both graphs 1 and 2, we conclude that the highest errors appear when the rendering distance is 5, 10 or 15 meters. For smaller distances, the errors are much lower.

RENDERING RESULTS - AZIMUTH 90°



Graph 2 Rendering results for 8 POIs – Azimuth 90° Degrees

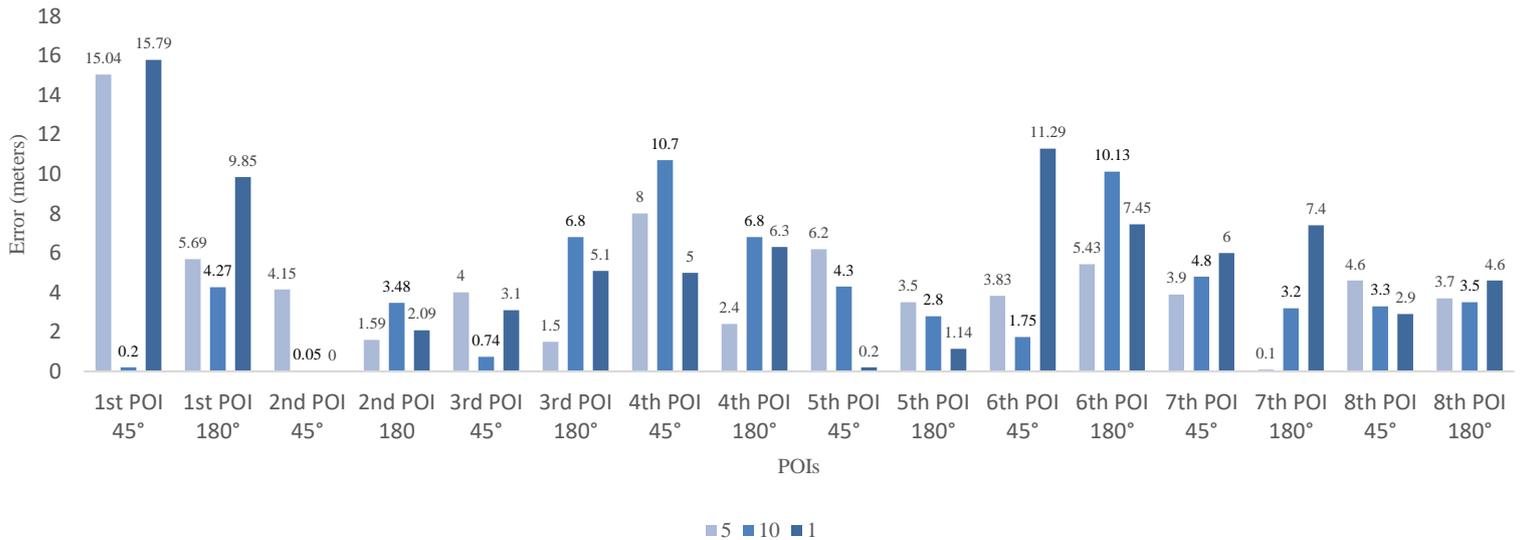
Below Table 6 is alike Table 5, only this time the results are referring to rendering distances: a) 1 meter, b) 5 meters and c) 10 meters for two different azimuths: 45° and 180° degrees.

Rendering Results						
POI	Coordinates	Azimuth (Degrees °)	Distance (meters)	Error (meters)		
				1	5	10
1 st POI	X: 37.9754889 Y: 23.7464209	Azimuth (Degrees °)	45	15.79	15.04	0.20
			180	9.85	5.69	4.27
2 nd POI	X: 37.9755478 Y: 23.7468358		45	0	4.15	0.05
			180	2.09	1.59	3.48
3 rd POI	X 37.9746368 Y 23.7477968		45	3.1	4	0.74
			180	5.10	1.5	6.8
4 th POI	X 37.9745671 Y 23.7473878		45	5	8	10.7
			180	6.3	2.4	6.8
5 th POI	X 37.9749764 Y 23.7464510		45	0.20	6.2	4.3
			180	1.14	3.5	2.8
6 th POI	X 37.9753930 Y 23.7471675		45	11.29	3.83	1.75
			180	7.45	5.43	10.13
7 th POI	X 37.9745409 Y 23.7475053		45	6	3.9	4.8
			180	7.4	0.10	3.2
8 th POI	X 37.9744354 Y 23.7471283		45	2.9	4.6	3.3
			180	4.6	3.7	3.5

Table 6 Rendering results for eight different POIs – 3 different distances (meters) – 2 different azimuths (degrees)

Graph 3 visualizes two types of tests for 45° and 180° degrees, respectively, for 1,5,10 meters distance from the rendering point. The results here we can say that are more satisfied than the previous tests. Here for both azimuths, the 1st POI shows the highest error of 15.79- and 15.04- meters for 1- and 5-meters rendering distance, respectively. 2nd, 3rd, 5th, 7th, and 8th POI provide the best results for both kinds of tests. Whereas 2nd POI produced the lowest error of 0 m for 1 m rendering distance and 45° azimuth.

Rendering Results Azimuth 45° & 180° Degrees



Graph 3 Rendering results for 8 POIs – 1m-5m-10m meters – 45° & 90° degrees Azimuth

Distance (meters)	1m	5m	10m	Error (meters)	Distance (meters)	1m	5m	10m	Error (meters)
1 st POI 45°	15.8	15	0.2		5 th POI 45°	0.2	6.2	4.3	
1 st POI 180°	9.85	5.69	4.27		5 th POI 180°	1.14	3.5	2.8	
2 nd POI 45°	0	4.15	0.05		6 th POI 45°	11.3	3.83	1.75	
2 nd POI 180°	2.09	1.59	3.48		6 th POI 180°	7.45	5.43	10.1	
3 rd POI 45°	3.1	4	0.74		7 th POI 45°	6	3.9	4.8	
3 rd POI 180°	5.1	1.5	6.8		7 th POI 180°	7.4	0.1	3.2	
4 th POI 45°	5	8	10.7		8 th POI 45°	2.9	4.6	3.3	
4 th POI 180°	6.3	2.4	6.8		8 th POI 180°	4.6	3.7	3.5	

Table 7 Rendering results for eight different POIs – 3 different distances (meters) – 2 different azimuths (degrees)

The reason that the 1st and 5th POIs visualize the highest errors in our tests are: a) the high human traffic at those places and b) the open terrain that allows the continuous flow of coordinates from satellites. Another reason for the low accuracy is the weak network signal that we have in Greece. This has affected to a great extent our test results.



Figure 8 Render Sphere Object at 8th POI

In the table below we have calculated:

- Average Positional Error for 0° direction to the north.
- Average Positional Error for 90° direction to the north.
- Standard Deviation for 0° direction to the north.
- Standard Deviation for 90° direction to the north.

Distance	Average Positional Error 0° (m)	Average Positional Error 90° (m)	Std. Deviation (m) 0°	Std. Deviation (m) 90°
1 st POI	4.3533	5.4467	2.2856	1.4478
2 nd POI	2.3417	0.9883	1.6763	0.9336
3 rd POI	4.8967	6.3450	2.4240	1.6263
4 th POI	3.9283	5.4683	2.1712	2.3038
5 th POI	3.9900	7.2017	2.1881	2.4874
6 th POI	1.5700	1.4417	1.3726	1.2322
7 th POI	4.7250	5.2417	2.3812	1.8413
8 th POI	3.2767	2.0750	1.9829	1.4018

Table 8 Average Positional Error and Standard Deviation for eight different POIs –2 different azimuths (degrees)

To calculate the Average Positional Error (1) and Standard Deviation (2), we used the following equations:

$$\bar{D} = \frac{1}{n} \sum_{i=1}^n D_i \quad (1)$$

$$SD = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (D_i - \bar{D})^2} \quad (2)$$

(Drosos and Malesios 2012)

And to calculate the accuracy, we used the method of Root Mean Square Error (3) using the following equation:

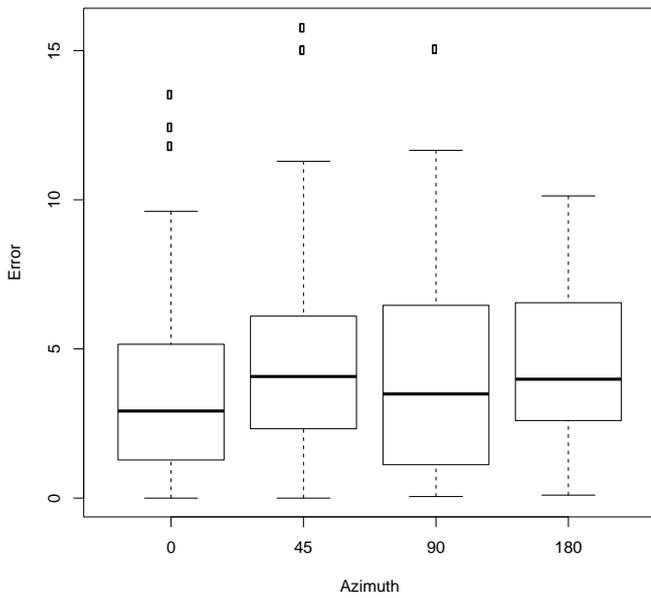
$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - a_i)^2} \quad (3)$$

(Deakin and Kildea 1999)

Below are the overall accuracy results we get from the above calculations:

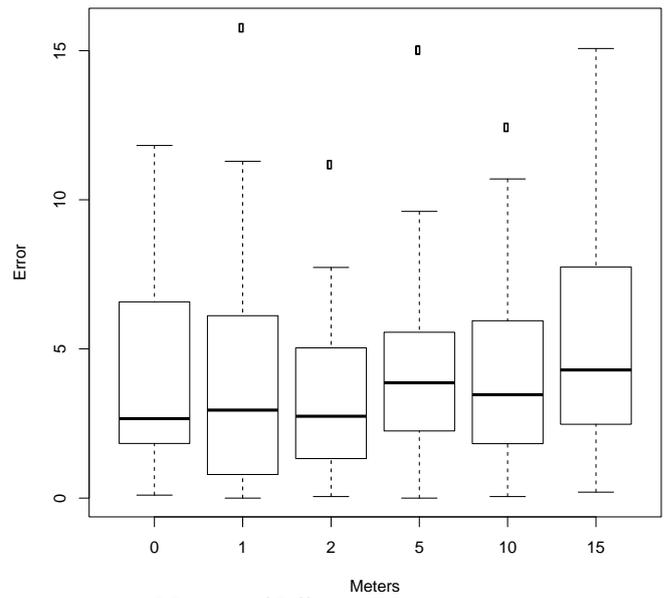
1. 0° degrees -> 5.06 meters
2. 90° degrees -> 5.40 meters
3. The average of those is 5.23 meters

Boxplot of Different Azimuths

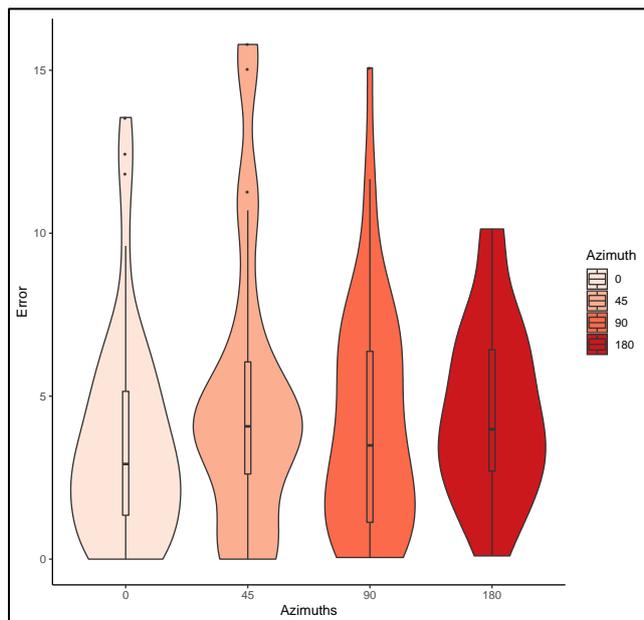


Graph 6 Boxplot of Different Meters

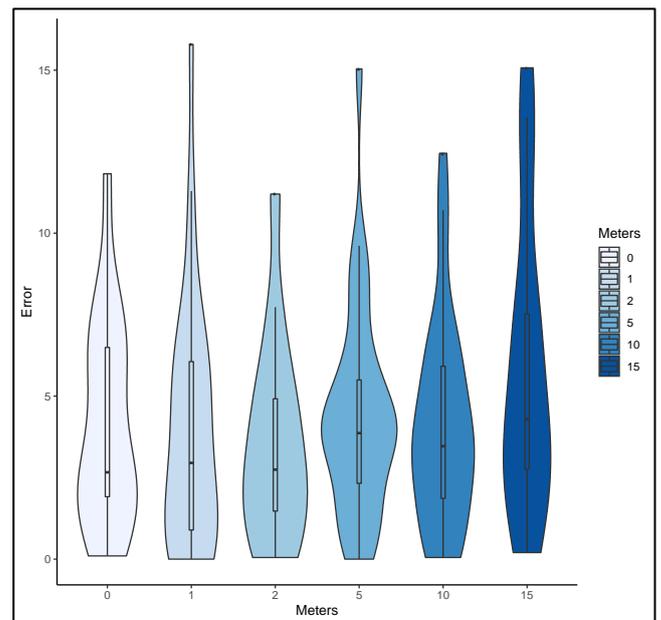
Boxplot of Different Meters



Graph 5 Boxplot of Different Azimuths



Graph 4 String Graph of Different Azimuths



Graph 7 String Graph of Different Meters

Above, we have created Boxplots and String Graphs using R programming language to show summarized the different results of the rendering tests and the distribution of error. Looking at Graphs 4 and 5, we can notice that the average error for different distances and azimuths is below 5 meters. There are some outliers at 0°, 45°, 90° degrees and 1, 2, 5, 10 meters. Apart from that, we can see that in any azimuth the errors are relatively the same. The rendering point with distance 15 meters in Graph 5 has the highest error whereas the rendering points with 0, 1, and 2 meters produce the lowest error.

In Graph 6 most values are lower than 10, while for 45° and 90° degrees there are errors high as 15 meters followed by 0° degrees that have lower than 15 meters high but higher than 10, with the latter three to have some outliers since most of the values for all three are below 10 meters error.

In Graph 7 we can see that all of the errors for all the distances are below 10. Rendering points at 1, 5, and 15 meters had the highest errors, but in general, for all the meters the error was less than 10 meters with average error to be 5 meters.

To sum up, from the above tests and calculations we can say that the overall GPS accuracy for 0° and 90° can be found in professional GPS only, so we can say that Apple has done a great job here, and iPhone 8 Plus can present accurately augmented objects on reality; that is the purpose of this project. However, there is plenty of room to improve the results regarding the long rendering distances, especially for the ones of 10 and 15 meters.

4.2. Disaster Response Scenarios

In this section, we discuss two different scenarios, where the application developed in this study can be useful during the disaster response phase. In the place of the disaster, the emergency workers want to see the status and position of people left behind, the damage percentage and the status of infrastructures during a rescue operation. We assume that before these scenarios, the citizens and the rescue teams have been informed about the app and know how to use it.

4.2.1. Geolocation Visualization (AR) Scenario

Suppose that an earthquake, that had a preliminary rating of 7.2 magnitudes, hit the city of Athens in Greece. Many buildings suffer big damages whereas others have been collapsed completely. Hundreds of people have been injured and many of them are trapped under the damaged buildings. The survivors are looking for supplies and shelters, while the ones that are trapped are scouting for help. The rescue teams are getting ready to help the people on the field. Once they arrive at the affected area, they begin immediately the Search and Rescue operation.

The use of information and on the spot examination is one of the most effective methods the rescue teams can use to find the exact position of the trapped persons and provide help to the victims. In this method, information is assessed and collected by non-trapped

tenants, relatives or neighbors. The information required refers to the number of trapped persons and their position in the area. Once the emergency workers have that information, they have to decide the most efficient way to get them out of the ruins in a short time. They have to act fast because the later they react, the more consequences there would be.

In this case scenario, the app can be proved extremely helpful not only for the rescue teams but for the citizens as well. The survivors can access the application and use the “Render POIs” feature (Fig. 11) to render the location of places such as hospitals, pharmacies, help centers, etc. on the real world through their device’s camera.



Figure 9 Render Health Status of User in Real world

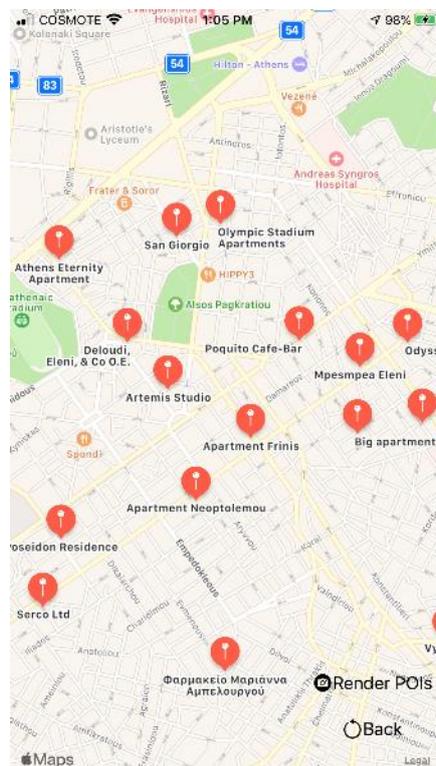


Figure 10 MapView of POIs

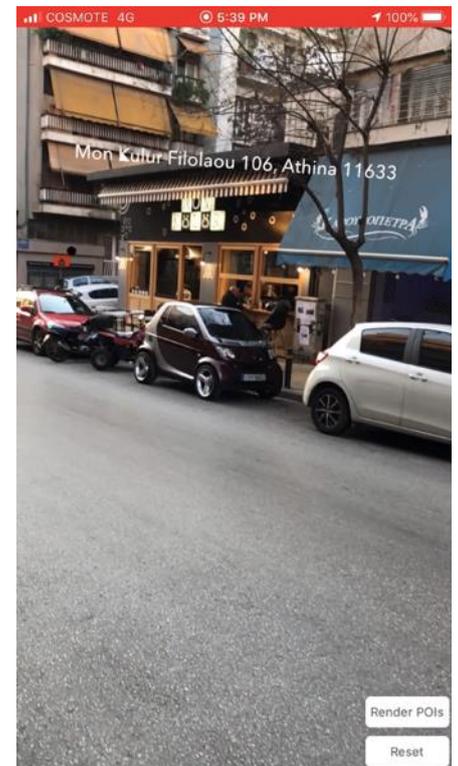


Figure 11 Render POIs

Apart from that, the people that are trapped but able to reach their devices can report their health status and location to the rescue teams. To do that, all they must do is to start the app, tap in the button “Render Status”. Then copy and paste their location from the Latitude and Longitude fields and paste them to the x and y coordinate fields respectively. Further, they have to type their health status in the relevant field and submit it along with their position on the app. After that, the emergency workers can go to the field and render this information by tapping at the “Render Status” button and know exactly how far they are from the victims and their health conditions. Finally, people around the affected area that could have useful information regarding victims or

reporting people that they were in the buildings that collapse, can as well send this information using the app.

4.2.2. Infrastructure Visualization scenario

Once the Search and Rescue operation be completed, the next step is for the emergency workers to evaluate the buildings' condition and decide which ones are safe to use. The app can provide the inspectors' accurate information about the location (e.g., building coordinates), attributes (e.g., kind of building and status) and the extent (e.g., 3-D building frame) of totally or partially collapsed buildings.



Figure 12 Render building Info / Damage %



Figure 13 Render Previous Infrastructure State



Figure 14 Render Previous Infrastructure State Different Angle

After the earthquake, the teams in charge of evaluating the damages will visit the affected areas and they can submit to the app the location of the buildings along with their status. They can do that by tapping into the “Render Building Info / Damage %” button and there they can copy and paste the coordinate from fields latitude and longitude type the damage of the building (Fig. 12) and then submit it. Once this information is stored, any following evaluator such as civil engineers or surveyors, etc,

can access these first reports by tapping the “Render Building Info” button and use them to classify the structure and evaluate it. Figure 12 visualizes such an example, where the user can see that the place in the corner is a bakery and has 4% damage.

The application has also the possibility to show the people how the building was before collapsing. So, the user can tap into the option “Render Previous State Building” and he can see the real world through the phone’s camera and render the building in the position where it was before the disaster. He can move around the building from different angles (Fig. 13 & 14) and examine it to take the information he needs.

5. Discussion

5.1. Challenges

The challenges met in this study were:

1. study the needs that can be covered by a Mobile AR application in Disaster Management,
2. render 3D objects in reality,
3. render the information in their most accurate positions on real-world, using real-world coordinates and
4. create a method to measure the accuracy of the rendering points using the app and the mobile device.

First, we had to study the field of disaster management, understand it and research deeply how this kind of app can be useful for the people during a disaster. Second, there are not many methods available to render the information in the real world using real-world coordinates. The one we have applied here is using the Haversine formula and matrices calculations, to render the information on the real world, using the user's real geolocation. It took a lot of time to implement and develop the back end of the application, using the mathematical equations presented in this study. The application has been debugging and tested several times, to achieve the appropriate results.

The third obstacle was to manage to render the 3D objects, with the proper scale, draw them and modify them in a way they are presentable in reality. To achieve that, we first had to learn how to use the Google SketchUp Program. Then we imported those in our app, and through several tests in the field, we managed to render the different 3D objects, e.g., building exterior, using the right dimensions, such as scale. Finally, was very challenging the fact that there were not many available resources that could be used to earn knowledge and use them in our study.

5.2. Advantages – Disadvantages

Below we are discussing the advantages and disadvantages of the app. First, the main advantage of this application is that the emergency worker can get an instant insight into the place, where the earthquake or any disaster occurred. He can superimpose

information on reality, on the most accurate position, and this lets him make fast decisions, regarding the plan he must follow and the actions that he has to take. Citizens can also navigate in the real world and find shelter, while they use the app to render POIs in the nearby areas.

On the other hand, the application has some disadvantages as well. To be able to calculate the exact location and render the information with the highest possible accuracy, the app consumed a significant amount of battery. While the app was running, the battery consumption was about 65% for 2 hours of usage. This battery loss is substantial, especially if we think that there will not be electricity or plugs in such a place, so the rescuers must carry many phones or power banks with them.

Another issue is that to take the highest accuracy of the GPS locations to render the information in the proper places, we had to stay still for 1-3 minutes. This occupational time has occurred because the app has been developed to take the signal from the highest number of available satellites. Waiting to take the proper position, though, is a crucial issue because while on a disaster response phase, people need to act fast and cannot wait too long to react. We could change this fact if we change the way we develop the app to search for an accurate position but not as precise as it is now.

The overall accuracy of the app is good, but there is plenty of room for improvement. Especially when we wanted to render information from long distances from the object, the rendered information was too far from the real object. For example, when we tried to render the sphere object from a 10-meter distance from the real statue in our tests, then the first one instead of appearing on the right location, it appears much further from it. In general, the longer the distance, the worst accuracy will be. This the major problem of the application. This can be improved in two different ways. First, we must study and created a different method, than the one presented here to render the information. This choice must be evaluated by debugging the app and running accuracy tests all over to check if the results improved or satisfied our expectations. Second, the choice of a more advanced smartphone such as iPhone XS could provide more accurate results due to its latest technological components, such as GPS, gyroscope, etc.

5.3. Future work

As of now, the application is in the testing phase and has been developed, working only outdoors. We should run further tests, to investigate the accuracy indoors as well. Also,

it would be nice to run tests using a Wi-Fi connection. The development of a database, that is needed for the implementation of the application and will be the host to the different kinds of data is a future step, as well. The database will organize the information associated with real-world coordinates, such as infrastructure addresses and 3D models created in Google SketchUp. We can extend the architecture of the application, to include other functionalities as well. The app is currently showing on reality specific POIs near the user's location, and this can be extended to show other amenities as well. A significant requirement could be indoor navigation and a second interface, including a map view along with the augmented reality view. Ultimately, it would be nice to have the capability to edit or update address data or any other information linked to the address infrastructure.

Apart from the above, it would be nice to develop the app for Android devices as well, because not everybody holds an iPhone. Most of the world's population is using Android devices, so it is important to provide the app to all target groups. Furthermore, the creation of a button that will allow the user to submit his geolocation without the need of copying, paste the coordinates is mandatory, because it will help the user to submit his location if he is trapped. One important addition to the app could be to test the app for distances longer than 15 meters to see how much accurate results we can get and try to find other ways of implementing the geolocation feature. Also, we could run tests taking into account weather variables such as air temperature (°F), relative humidity (%), barometric pressure (inches), wind speed (mph), and weather various weather conditions such as dark, clear, partly cloudy, overcast, etc.

The rendering tests show that the iPhone 8 Plus has a fantastic GPS accuracy for a handling device and can be an excellent tool for anybody out there, that needs a GPS. In the future, it would be great to test newer iPhones as well, such as iPhone 11 Pro and Pro Max, that carry more advanced hardware compare to iPhone 8 Plus and powerful Android phones as well, such as Samsung Galaxy Note 10.

6. Conclusion

The primary objective of this Thesis was to build an iOS application prototype, to serve the needs of the disaster management teams as well as the citizens one's. This application was conceptualized, designed, partially developed, and implemented in Athens, Greece. A prototype of this application used to test the iPhone's eight-Plus accuracy, regarding its ability to render augmented information on the real world, using real-world coordinates. The app can extend to accommodate more functionalities and serve further needs of the emergency workers. Apart from the application implemented in such a way, to be as user-friendly as possible. There is not any other application of this kind, that can render information on the real world, using real-world coordinates and not the ones that apps like Google Maps provide.

The results from the rendering tests show that the accuracy of the iPhone's eight-plus sensors is as close as the one the professional GPS has, and due to that, we can render with confidence the information correctly in the place where they belong in the real world. Users can use this application to navigate through reality and see the: a) damage status and kind of infrastructure, b) the exterior of it as well as the locations of the several rooms, c) location and state of the people that need help, d) points of interests in the area nearby. Furthermore, the user can see on the app interface, how far it is from a rendering point.

The rendering tests raised some issues concerning the application. The accuracy of the coordinates demands the user to stay approximately in a place standing 1-3 minutes, and many factors are affecting this one, such as the wind, crowd traffic, etc. The proposed system is a comprehensive system for disaster management on iOS phones that can host AR technology. The realized prototype confirms that the system works very well outdoors. Future work and research could be done in indoor navigation and test the app's results there, and more functionalities could be built to host further emergency worker's needs on the field.

To conclude, the accuracy of the iPhone 8 Plus is excellent and can be compared to professional GPS, that surveyors, civil engineers, etc. are using. Mobile GPS is the future of a handling GPS, where the user doesn't have to carry any more multiple devices. Having a phone like this and an application that can take advantage of the device's hardware, the user can have a powerful tool in his hand to do any job that demands GPS.

References

- Alem, L., and W. Huang. 2011. *Recent Trends of Mobile Collaborative Augmented Reality Systems*.
- Apple. 2019a. AR Kit. Apple Developer. Retrieved 2019-02-28 2019, from. <https://developer.apple.com/augmented-reality/>
- Apple. 2019b. ARConfiguration.WorldAlignment.gravityAndHeading. Retrieved 25/10/2019, from. <https://developer.apple.com/documentation/arkit/arconfiguration/worldalignment/gravityandheading>
- Apple. 2019c. ARWorldTrackingConfiguration. Retrieved 10/02/2019, from. <https://developer.apple.com/documentation/arkit/arworldtrackingconfiguration>
- Apple. 2019d. SceneKit. Apple Developer. Retrieved 28/02/2018 2019, from. <https://developer.apple.com/documentation/scenekit>
- Arth, C., R. Grasset, L. Gruber, T. Langlotz, A. Mulloni, and D. Wagner. 2015. The History of Mobile Augmented Reality.
- Aukstakalnis, S. Blatner, and David. 1992. Silicon Mirage: The Art and Science of Virtual Reality. *Peachpit Press*.
- Azuma, R., Y. Baillet, R. Behringer, S. Feiner, S. Julier, and B. Macintyre. 2001. Recent advances in augmented reality. *IEEE Comput Graphics Appl. Computer Graphics and Applications, IEEE*, 21: 34-47. DOI: 10.1109/38.963459
- Azuma, R. T. 1997. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6: 355-385. DOI: 10.1162/pres.1997.6.4.355
- Behringer, R., J. Park, and V. Sundareswaran. 2002. *Model-based visual tracking for outdoor augmented reality applications*.
- Brunetti, P., A. Croatti, A. Ricci, and M. Viroli. 2015. Smart Augmented Fields for Emergency Operations. *Procedia Computer Science*, 63: 392-399. DOI: 10.1016/j.procs.2015.08.358
- Carmigniani, J., and B. Furht. 2011. Augmented Reality: An Overview. 3-46 pp.
- Cheung, C., C. Wedlake, J. Moore, S. Pautler, and T. Peters. 2010. *Fused Video and Ultrasound Images for Minimally Invasive Partial Nephrectomy: A Phantom Study*.
- Cova, T. 1999. GIS in Emergency Management. 845-858 pp.
- Deakin, R., and D. Kildea. 1999. A note on standard deviation and RMS. *Australian Surveyor*, 44. DOI: 10.1080/00050326.1999.10441907
- Drosos, V., and C. Malesios. 2012. Measuring the Accuracy and Precision of the Garmin GPS Positioning in Forested Areas: A Case Study in Taxiarchis-Vrastama University Forest. *Journal of Environmental Science and Engineering B*, 1. DOI: 10.17265/2162-5263/2012.04.015
- Erra, U., D. Mirauda, R. Agatiello, and M. Cerverizzo. 2018. Mobile Augmented Reality For Flood Events Management. *International Journal of Sustainable Development and Planning*, 13: 418-424. DOI: 10.2495/SDP-V13-N3-418-424
- Feiner, S., B. Macintyre, T. Höllerer, and A. Webster. 1997. *A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment*.
- Geng, X., F. Yang, G. Pan, Z. Kong, and S. Wang. 2014. A New Design of Video Streaming Capture and Transmission System. *Journal of Multimedia*, 9. DOI: 10.4304/jmm.9.10.1160-1165
- Grimson, W. E. L., G. J. Ettinger, S. J. White, T. Lozano-Perez, W. Wells, and R. Kikinis. 1996. An automatic registration method for frameless stereotaxy,

- image guided surgery, and enhanced reality visualization. *Medical Imaging, IEEE Transactions on*, 15: 129-140. DOI: 10.1109/42.491415
- Hicks, J. D., R. A. Flanagan, P. V. Petrov, and A. D. Stoyen. 2003. *Eyekon: augmented reality for battlefield soldiers*.
- Hillyer, N. (2016). The Ultimate Guide to Choosing Objective-C or Swift for Your Project. <https://savvyapps.com/blog/ultimate-guide-choosing-objective-c-or-swift> (last accessed 22.02.2018).
- Hoffman, J. 2017. *Mastering Swift 4 - Fourth Edition*.
- Janin, A. L., D. Mizell, and T. Caudell. 1993. *Calibration of head-mounted display for augmented reality applications*.
- Junwei, Y., F. Lu, and L. Chuazheng. 2016. Key technology and application research on mobile augmented reality. In *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 547-550.
- Juraschek, M., L. Büth, F. Cerdas, A. Kaluza, S. Thiede, and C. Herrmann. 2018. Exploring the Potentials of Mixed Reality for Life Cycle Engineering. *Procedia CIRP*, 69: 638-643. DOI: <https://doi.org/10.1016/j.procir.2017.11.123>
- Khademi, M., H. Mousavi Hondori, C. Lopes, I. dodakian, and S. Cramer. 2012. *Haptic Augmented Reality to Monitor Human Arm's Stiffness in Rehabilitation*.
- Kim, K., B.-K. Seo, J.-H. Han, and J.-I. Park. 2009. *Augmented reality tour system for immersive experience of cultural heritage*.
- Lee, J.-D., H.-C. Lee, C.-H. Hsieh, C.-T. Wu, and S.-T. Lee. 2015a. *A Projection-based Medical Augmented Reality System*.
- Lee, J. D., H. C. Lee, C. H. Hsieh, C. T. Wu, and S. T. Lee. 2015b. A Projection-based Medical Augmented Reality System. *8th International Symposium on Visual Information Communication and Interaction (VINCI 2015)*: 164-165. DOI: 10.1145/2801040.2801065
- Leebmann, J. 2019. An Augmented Reality System for Earthquake Disaster Response.
- Liarokapis, F. 2007. An augmented reality interface for visualizing and interacting with virtual content. *Virtual Reality*, 11: 23-43. DOI: 10.1007/s10055-006-0055-1
- Livingston, M., S. Julier, D. Brown, J. Ii, J. Gabbard, and D. Hix. 2008. An augmented reality system for military operations in urban terrain.
- Lochhead, I. 2019. Mixed reality emergency management: bringing virtual evacuation simulations into real-world built environments. *International journal of digital earth*, v. 12: pp. 190-208-2019 v.2012 no.2012. DOI: 10.1080/17538947.2018.1425489
- Luchetti, G., A. Mancini, M. Sturari, E. Frontoni, and P. Zingaretti. 2017. Whistland: An Augmented Reality Crowd-Mapping System for Civil Protection and Emergency Management. *ISPRS International Journal of Geo-Information*, 6: 41. DOI: 10.3390/ijgi6020041
- Maad, S. 2010. *Augmented Reality*. Vukovar: Intech.
- Mandal, S. 2013. Brief Introduction of Virtual Reality & its Challenges.
- Milgram, P., and F. Kishino. 1994. A Taxonomy of Mixed Reality Visual Displays. *IEICE Trans. Information Systems*, vol. E77-D, no. 12: 1321-1329.
- Paavilainen, J., H. Korhonen, K. Alha, J. Stenros, E. Koskinen, and F. Mäyrä. 2017. *The Pokémon GO Experience: A Location-Based Augmented Reality Mobile Game Goes Mainstream*.
- Palmarini, R., J. Erkoyuncu, R. Roy, and H. Torabmostaedi. 2018. A systematic review of augmented reality applications in maintenance. *Robotics and Computer-Integrated Manufacturing*, 49: 215-228. DOI: 10.1016/j.rcim.2017.06.002

- Parveau, M., and M. Adda. 2018. 3iVClass: a new classification method for Virtual, Augmented and Mixed Realities. *Procedia Computer Science*, 141: 263-270. DOI: <https://doi.org/10.1016/j.procs.2018.10.180>
- Reboucas, M., G. Pinto, F. Ebert, W. Torres, A. Serebrenik, and F. Castor. 2016. *An Empirical Study on the Usage of the Swift Programming Language*.
- Reitmayr, G., and D. Schmalstieg. 2001. *Mobile Collaborative Augmented Reality*.
- Roberts, D., J. Strohbehn, J. Hatch, W. Murray, and H. Kettenberger. 1986. A Frameless Stereotaxic Integration of Computerized Tomographic Imaging and the Operating Microscope. *Journal of neurosurgery*, 65: 545-549. DOI: 10.3171/jns.1986.65.4.0545
- Roland, P. E., C. J. Graufelds, W. H. J, L. Ingelman, M. Andersson, A. Ledberg, J. Pedersen, S. Akerman, et al. 1994. Human brain atlas: For high-resolution functional and anatomical mapping. *Hum Brain Mapp*, 1: 173-184. DOI: 10.1002/hbm.460010303
- Rolland, J., and H. Fuchs. 2000. Optical Versus Video See-Through Head-Mounted Displays in Medical Visualization. *Presence*, 9: 287-309. DOI: 10.1162/105474600566808
- Solt, P. 2015. Swift vs. Objective-C: 10 reasons the future favors Swift. Retrieved 11/11/2016 2016, from. <https://www.infoworld.com/article/2920333/swift-vs-objective-c-10-reasons-the-future-favors-swift.html>
- Speicher, M., B. Hall, and M. Nebeling. 2019. *What is Mixed Reality?*
- State, A., M. Livingston, W. Garrett, G. Hirota, M. Whitton, E. Pisano, and H. Fuchs. 1996. *Technologies for Augmented Reality Systems: Realizing Ultrasound-Guided Needle Biopsies*.
- Stoimenov, L., A. Milosavljevic, and A. Stanimirović. 2007. GIS as a Tool in Emergency Management Process. *Lecture Notes in Engineering and Computer Science*, 2165.
- Sutherland, I. 1969. A head-mounted three dimensional display. *FJCC*, 33: 757-764. DOI: 10.1145/1476589.1476686
- Takagi, A., S. Yamazaki, Y. Saito, and N. Taniguchi. 2000. Development of a stereo video see-through HMD for AR systems. 68-77. Institute of Electrical and Electronics Engineers Inc.
- Tano, S. i., K. Suzuki, K. Miki, N. Watanabe, M. Iwata, T. Hashiyama, J. Ichino, and K. Nakayama. 2012. Simple augmented reality system for 3D ultrasonic image by see-through HMD and single camera and marker combination. DOI: 10.1109/BHI.2012.6211617
- Tiwari, A. 2015. A Detailed 3D GIS Architecture for Disaster Management. *International Journal of Advanced Remote Sensing and GIS*, 4: 980-989. DOI: 10.23953/cloud.ijarsg.132
- Tönnis, M. 2010. *Augmented Reality*. Springer-Verlag Berlin Heidelberg.
- Van Krevelen, R., and R. Poelman. 2010. A Survey of Augmented Reality Technologies, Applications and Limitations. *International Journal of Virtual Reality (ISSN 1081-1451)*, 9: 1. DOI: 10.20870/IJVR.2010.9.2.2767
- Vlahakis, V., M. Ioannidis, J. Karigiannis, M. Tsotros, M. Gounaris, D. Stricker, T. Gleue, P. Daehne, et al. 2002. Archeoguide: An augmented reality guide for archaeolog sites. *Computer Graphics and Applications, IEEE*, 22: 52-60. DOI: 10.1109/MCG.2002.1028726
- Webb, C. 2019. ARKit and CoreLocation. Retrieved 25/01/2019 2019, from. <https://medium.com/journey-of-one-thousand-apps/arkit-and-corelocation-part-two-7b045fb1d7a1>

Appendix

```
// locationManager()
locationManager.desiredAccuracy = kCLLocationAccuracyBestForNavigation
locationManager.distanceFilter = kCLDistanceFilterNone
locationManager.headingFilter = kCLHeadingFilterNone
locationManager.pausesLocationUpdatesAutomatically = false
locationManager.delegate = self
locationManager.requestWhenInUseAuthorization()
locationManager.startUpdatingLocation()
locationManager.startUpdatingHeading()

func locationManager(_ manager: CLLocationManager, didUpdateHeading newHeading: CLHeading) {
    if newHeading.headingAccuracy < 0 { return }
    let heading = newHeading.trueHeading > 0 ? newHeading.trueHeading : newHeading.magneticHeading
    userHeading = heading
    NotificationCenter.default.post(name: Notification.Name(rawValue:"myNotificationName"), object: self, userInfo:
        nil)
}
```

Figure 15 Setup the location manager

```
let distance_new = Float(stepLocation.distance(from: stepLocation1))
```

Figure 13 Distance calculation

```
static func bearingToLocationRadian(destinationLocation: CLLocation, start_location:
    CLLocation) -> Double {
    let lat1 = start_location.coordinate.latitude.toRadians()
    let lon1 = start_location.coordinate.longitude.toRadians()
    let lat2 = destinationLocation.coordinate.latitude.toRadians()
    let lon2 = destinationLocation.coordinate.longitude.toRadians()
    let dLon = lon2 - lon1
    let x = sin(dLon) * cos(lat2)
    let y = cos(lat1) * sin(lat2) - sin(lat1) * cos(lat2) * cos(dLon);
    let radiansBearing = atan2(x, y)
    return radiansBearing
}
```

Figure 16 Calculate bearing

```
func toDegrees() -> Double {
    return self * 180.0 / .pi
}
```

Figure 17 Convert radians to degrees

```

import Foundation
import CoreLocation
import GLKit

class MatrixHelper {
    static func translationMatrix(with matrix: matrix_float4x4, for translation : vector_float4) -> matrix_float4x4 {
        var matrix = matrix
        matrix.columns.3 = translation
        return matrix
    }
    static func rotateAroundY(with matrix: matrix_float4x4, for degrees: Float) -> matrix_float4x4 {
        var matrix : matrix_float4x4 = matrix
        matrix.columns.0.x = cos(degrees)
        matrix.columns.0.z = -sin(degrees)
        matrix.columns.2.x = sin(degrees)
        matrix.columns.2.z = cos(degrees)
        return matrix.inverse
    }
    static func transformMatrix(for matrix: simd_float4x4, originLocation: CLLocation, location: CLLocation) -> simd_float4x4
    {
        let distance = Float(location.distance(from: originLocation))
        let bearing = bearingToLocationRadian(destinationLocation: location, start_location: originLocation)
        let position = vector_float4(0.0, 0.0, -distance, 0.0)
        let translationMatrix = MatrixHelper.translationMatrix(with: matrix_identity_float4x4, for: position)
        let rotationMatrix = MatrixHelper.rotateAroundY(with: matrix_identity_float4x4, for: Float(bearing))
        let transformMatrix = simd_mul(rotationMatrix, translationMatrix)
        return simd_mul(matrix, transformMatrix)
    }
    static func bearingToLocationRadian(destinationLocation: CLLocation, start_location: CLLocation) -> Double {
        let lat1 = start_location.coordinate.latitude.toRadians()
        let lon1 = start_location.coordinate.longitude.toRadians()
        let lat2 = destinationLocation.coordinate.latitude.toRadians()
        let lon2 = destinationLocation.coordinate.longitude.toRadians()
        let dLon = lon2 - lon1
        let x = sin(dLon) * cos(lat2)
        let y = cos(lat1) * sin(lat2) - sin(lat1) * cos(lat2) * cos(dLon);
        let radiansBearing = atan2(x, y)
        return radiansBearing
    }
}

```

Figure 18 Matrices functions

```

static func positionFromTransform(_ transform: matrix_float4x4) -> SCNVector3 {
    return SCNVector3Make(transform.columns.3.x, transform.columns.3.y, transform.columns.3.z)
}

let locationTransform = MatrixHelper.transformMatrix(for: matrix_identity_float4x4, originLocation: stepLocation1,
    location: stepLocation)

let locationNew = SCNVector3.positionFromTransform(locationTransform)

```

Figure 19 Transformed location

```

let configuration = ARWorldTrackingConfiguration()

override func viewDidLoad() {
    // Setup the AR view
    super.viewDidLoad()
    configuration.worldAlignment = .gravityAndHeading
    self.sceneView.session.run(configuration)
}

```

Figure 20 Setup AR view

```

func addSphere(){
    let coordinate = CLLocationCoordinate2D(latitude: UserDefaults.standard.value(forKey: "LAT") as! CLLocationCoordinateDegrees,
        longitude: UserDefaults.standard.value(forKey: "LON") as! CLLocationCoordinateDegrees)
    let new1 = Double(xCoord.text!)
    let new2 = Double(yCoord.text!)
    let stepLocation1 = CLLocationCoordinate(latitude: coordinate.latitude, longitude: coordinate.longitude)
    let step = CLLocationCoordinate2D(latitude: new1!, longitude: new2!)
    let stepLocation = CLLocationCoordinate(latitude: step.latitude, longitude: step.longitude)
    let locationTransform = MatrixHelper.transformMatrix(for: matrix_identity_float4x4, originLocation: stepLocation1,
        location: stepLocation)
    let distance_new = Float(stepLocation.distance(from: stepLocation1))
    let s = NSString(format: "%.3f", distance_new)
    distance.text = "\(s) meters"
    let locationNew = SCNVector3.positionFromTransform(locationTransform)
    let stepAnchor = ARAnchor(transform: locationTransform)
    let sphere = BaseNode(title: "The sphere has coordinates x:\(step.latitude) y:\(step.longitude)")
    let scale = 100 / Float(distance_new)
    sphere.scale = SCNVector3(x: scale, y: scale, z: scale)
    sphere.position = locationNew
    sphere.location = stepLocation
    sphere.addNode(with: 0.1, and: .blue, and: "Coordinates")
    sphere.anchor = stepAnchor
    sceneView.session.add(anchor: stepAnchor)
    sceneView.scene.rootNode.addChildNode(sphere)
}

```

Listing 1 Render sphere in a real-world location

Series from Lund University

Department of Physical Geography and Ecosystem Science

Master Thesis in Geographical Information Science

1. *Anthony Lawther*: The application of GIS-based binary logistic regression for slope failure susceptibility mapping in the Western Grampian Mountains, Scotland (2008).
2. *Rickard Hansen*: Daily mobility in Grenoble Metropolitan Region, France. Applied GIS methods in time geographical research (2008).
3. *Emil Bayramov*: Environmental monitoring of bio-restoration activities using GIS and Remote Sensing (2009).
4. *Rafael Villarreal Pacheco*: Applications of Geographic Information Systems as an analytical and visualization tool for mass real estate valuation: a case study of Fontibon District, Bogota, Columbia (2009).
5. *Siri Oestreich Waage*: a case study of route solving for oversized transport: The use of GIS functionalities in transport of transformers, as part of maintaining a reliable power infrastructure (2010).
6. *Edgar Pimiento*: Shallow landslide susceptibility – Modelling and validation (2010).
7. *Martina Schäfer*: Near real-time mapping of floodwater mosquito breeding sites using aerial photographs (2010).
8. *August Pieter van Waarden-Nagel*: Land use evaluation to assess the outcome of the programme of rehabilitation measures for the river Rhine in the Netherlands (2010).
9. *Samira Muhammad*: Development and implementation of air quality data mart for Ontario, Canada: A case study of air quality in Ontario using OLAP tool. (2010).
10. *Fredros Oketch Okumu*: Using remotely sensed data to explore spatial and temporal relationships between photosynthetic productivity of vegetation and malaria transmission intensities in selected parts of Africa (2011).
11. *Svajunas Plunge*: Advanced decision support methods for solving diffuse water pollution problems (2011).
12. *Jonathan Higgins*: Monitoring urban growth in greater Lagos: A case study using GIS to monitor the urban growth of Lagos 1990 - 2008 and produce future growth prospects for the city (2011).
13. *Mårten Karlberg*: Mobile Map Client API: Design and Implementation for Android (2011).
14. *Jeanette McBride*: Mapping Chicago area urban tree canopy using color infrared imagery (2011).
15. *Andrew Farina*: Exploring the relationship between land surface temperature and vegetation abundance for urban heat island mitigation in Seville, Spain (2011).
16. *David Kanyari*: Nairobi City Journey Planner: An online and a Mobile Application (2011).

17. *Laura V. Drews*: Multi-criteria GIS analysis for siting of small wind power plants - A case study from Berlin (2012).
18. *Qaisar Nadeem*: Best living neighborhood in the city - A GIS based multi criteria evaluation of ArRiyadh City (2012).
19. *Ahmed Mohamed El Saeid Mustafa*: Development of a photo voltaic building rooftop integration analysis tool for GIS for Dokki District, Cairo, Egypt (2012).
20. *Daniel Patrick Taylor*: Eastern Oyster Aquaculture: Estuarine Remediation via Site Suitability and Spatially Explicit Carrying Capacity Modeling in Virginia's Chesapeake Bay (2013).
21. *Angeleta Oveta Wilson*: A Participatory GIS approach to *unearthing* Manchester's Cultural Heritage 'gold mine' (2013).
22. *Ola Svensson*: Visibility and Tholos Tombs in the Messenian Landscape: A Comparative Case Study of the Pylian Hinterlands and the Soulima Valley (2013).
23. *Monika Ogden*: Land use impact on water quality in two river systems in South Africa (2013).
24. *Stefan Rova*: A GIS based approach assessing phosphorus load impact on Lake Flaten in Salem, Sweden (2013).
25. *Yann Buhot*: Analysis of the history of landscape changes over a period of 200 years. How can we predict past landscape pattern scenario and the impact on habitat diversity? (2013).
26. *Christina Fotiou*: Evaluating habitat suitability and spectral heterogeneity models to predict weed species presence (2014).
27. *Inese Linuza*: Accuracy Assessment in Glacier Change Analysis (2014).
28. *Agnieszka Griffin*: Domestic energy consumption and social living standards: a GIS analysis within the Greater London Authority area (2014).
29. *Brynja Guðmundsdóttir*: Detection of potential arable land with remote sensing and GIS - A Case Study for Kjósarhreppur (2014).
30. *Oleksandr Nekrasov*: Processing of MODIS Vegetation Indices for analysis of agricultural droughts in the southern Ukraine between the years 2000-2012 (2014).
31. *Sarah Tressel*: Recommendations for a polar Earth science portal in the context of Arctic Spatial Data Infrastructure (2014).
32. *Caroline Gevaert*: Combining Hyperspectral UAV and Multispectral Formosat-2 Imagery for Precision Agriculture Applications (2014).
33. *Salem Jamal-Uddeen*: Using GeoTools to implement the multi-criteria evaluation analysis - weighted linear combination model (2014).
34. *Samanah Seyedi-Shandiz*: Schematic representation of geographical railway network at the Swedish Transport Administration (2014).
35. *Kazi Masel Ullah*: Urban Land-use planning using Geographical Information System and analytical hierarchy process: case study Dhaka City (2014).
36. *Alexia Chang-Wailing Spitteler*: Development of a web application based on MCDA and GIS for the decision support of river and floodplain rehabilitation projects (2014).
37. *Alessandro De Martino*: Geographic accessibility analysis and evaluation of potential changes to the public transportation system in the City of Milan (2014).
38. *Alireza Mollasalehi*: GIS Based Modelling for Fuel Reduction Using Controlled Burn in Australia. Case Study: Logan City, QLD (2015).

39. *Negin A. Sanati*: Chronic Kidney Disease Mortality in Costa Rica; Geographical Distribution, Spatial Analysis and Non-traditional Risk Factors (2015).
40. *Karen McIntyre*: Benthic mapping of the Bluefields Bay fish sanctuary, Jamaica (2015).
41. *Kees van Duijvendijk*: Feasibility of a low-cost weather sensor network for agricultural purposes: A preliminary assessment (2015).
42. *Sebastian Andersson Hylander*: Evaluation of cultural ecosystem services using GIS (2015).
43. *Deborah Bowyer*: Measuring Urban Growth, Urban Form and Accessibility as Indicators of Urban Sprawl in Hamilton, New Zealand (2015).
44. *Stefan Arvidsson*: Relationship between tree species composition and phenology extracted from satellite data in Swedish forests (2015).
45. *Damián Giménez Cruz*: GIS-based optimal localisation of beekeeping in rural Kenya (2016).
46. *Alejandra Narváez Vallejo*: Can the introduction of the topographic indices in LPJ-GUESS improve the spatial representation of environmental variables? (2016).
47. *Anna Lundgren*: Development of a method for mapping the highest coastline in Sweden using breaklines extracted from high resolution digital elevation models (2016).
48. *Oluwatomi Esther Adejoro*: Does location also matter? A spatial analysis of social achievements of young South Australians (2016).
49. *Hristo Dobrev Tomov*: Automated temporal NDVI analysis over the Middle East for the period 1982 - 2010 (2016).
50. *Vincent Muller*: Impact of Security Context on Mobile Clinic Activities A GIS Multi Criteria Evaluation based on an MSF Humanitarian Mission in Cameroon (2016).
51. *Gezahagn Negash Seboka*: Spatial Assessment of NDVI as an Indicator of Desertification in Ethiopia using Remote Sensing and GIS (2016).
52. *Holly Buhler*: Evaluation of Interfacility Medical Transport Journey Times in Southeastern British Columbia. (2016).
53. *Lars Ole Grottenberg*: Assessing the ability to share spatial data between emergency management organisations in the High North (2016).
54. *Sean Grant*: The Right Tree in the Right Place: Using GIS to Maximize the Net Benefits from Urban Forests (2016).
55. *Irshad Jamal*: Multi-Criteria GIS Analysis for School Site Selection in Gorno-Badakhshan Autonomous Oblast, Tajikistan (2016).
56. *Fulgencio Sanmartín*: Wisdom-volcano: A novel tool based on open GIS and time-series visualization to analyse and share volcanic data (2016).
57. *Nezha Acil*: Remote sensing-based monitoring of snow cover dynamics and its influence on vegetation growth in the Middle Atlas Mountains (2016).
58. *Julia Hjalmarsson*: A Weighty Issue: Estimation of Fire Size with Geographically Weighted Logistic Regression (2016).
59. *Mathewos Tamiru Amato*: Using multi-criteria evaluation and GIS for chronic food and nutrition insecurity indicators analysis in Ethiopia (2016).
60. *Karim Alaa El Din Mohamed Soliman El Attar*: Bicycling Suitability in Downtown, Cairo, Egypt (2016).

61. *Gilbert Akol Echelai*: Asset Management: Integrating GIS as a Decision Support Tool in Meter Management in National Water and Sewerage Corporation (2016).
62. *Terje Slinning*: Analytic comparison of multibeam echo soundings (2016).
63. *Gréta Hlín Sveinsdóttir*: GIS-based MCDA for decision support: A framework for wind farm siting in Iceland (2017).
64. *Jonas Sjögren*: Consequences of a flood in Kristianstad, Sweden: A GIS-based analysis of impacts on important societal functions (2017).
65. *Nadine Raska*: 3D geologic subsurface modelling within the Mackenzie Plain, Northwest Territories, Canada (2017).
66. *Panagiotis Symeonidis*: Study of spatial and temporal variation of atmospheric optical parameters and their relation with PM 2.5 concentration over Europe using GIS technologies (2017).
67. *Michaela Bobeck*: A GIS-based Multi-Criteria Decision Analysis of Wind Farm Site Suitability in New South Wales, Australia, from a Sustainable Development Perspective (2017).
68. *Raghdaa Eissa*: Developing a GIS Model for the Assessment of Outdoor Recreational Facilities in New Cities Case Study: Tenth of Ramadan City, Egypt (2017).
69. *Zahra Khais Shahid*: Biofuel plantations and isoprene emissions in Svea and Götaland (2017).
70. *Mirza Amir Liaquat Baig*: Using geographical information systems in epidemiology: Mapping and analyzing occurrence of diarrhea in urban - residential area of Islamabad, Pakistan (2017).
71. *Joakim Jörwall*: Quantitative model of Present and Future well-being in the EU-28: A spatial Multi-Criteria Evaluation of socioeconomic and climatic comfort factors (2017).
72. *Elin Haettner*: Energy Poverty in the Dublin Region: Modelling Geographies of Risk (2017).
73. *Harry Eriksson*: Geochemistry of stream plants and its statistical relations to soil- and bedrock geology, slope directions and till geochemistry. A GIS-analysis of small catchments in northern Sweden (2017).
74. *Daniel Gardevärn*: PPGIS and Public meetings – An evaluation of public participation methods for urban planning (2017).
75. *Kim Friberg*: Sensitivity Analysis and Calibration of Multi Energy Balance Land Surface Model Parameters (2017).
76. *Viktor Svanerud*: Taking the bus to the park? A study of accessibility to green areas in Gothenburg through different modes of transport (2017).
77. *Lisa-Gaye Greene*: Deadly Designs: The Impact of Road Design on Road Crash Patterns along Jamaica's North Coast Highway (2017).
78. *Katarina Jemec Parker*: Spatial and temporal analysis of fecal indicator bacteria concentrations in beach water in San Diego, California (2017).
79. *Angela Kabiru*: An Exploratory Study of Middle Stone Age and Later Stone Age Site Locations in Kenya's Central Rift Valley Using Landscape Analysis: A GIS Approach (2017).
80. *Kristean Björkmann*: Subjective Well-Being and Environment: A GIS-Based Analysis (2018).
81. *Williams Erhunmonmen Ojo*: Measuring spatial accessibility to healthcare for people living with HIV-AIDS in southern Nigeria (2018).

82. *Daniel Assefa*: Developing Data Extraction and Dynamic Data Visualization (Styling) Modules for Web GIS Risk Assessment System (WGRAS). (2018).
83. *Adela Nistora*: Inundation scenarios in a changing climate: assessing potential impacts of sea-level rise on the coast of South-East England (2018).
84. *Marc Seliger*: Thirsty landscapes - Investigating growing irrigation water consumption and potential conservation measures within Utah's largest master-planned community: Daybreak (2018).
85. *Luka Jovičić*: Spatial Data Harmonisation in Regional Context in Accordance with INSPIRE Implementing Rules (2018).
86. *Christina Kourdounouli*: Analysis of Urban Ecosystem Condition Indicators for the Large Urban Zones and City Cores in EU (2018).
87. *Jeremy Azzopardi*: Effect of distance measures and feature representations on distance-based accessibility measures (2018).
88. *Patrick Kabatha*: An open source web GIS tool for analysis and visualization of elephant GPS telemetry data, alongside environmental and anthropogenic variables (2018).
89. *Richard Alphonse Giliba*: Effects of Climate Change on Potential Geographical Distribution of *Prunus africana* (African cherry) in the Eastern Arc Mountain Forests of Tanzania (2018).
90. *Eiður Kristinn Eiðsson*: Transformation and linking of authoritative multi-scale geodata for the Semantic Web: A case study of Swedish national building data sets (2018).
91. *Niamh Harty*: HOP!: a PGIS and citizen science approach to monitoring the condition of upland paths (2018).
92. *José Estuardo Jara Alvear*: Solar photovoltaic potential to complement hydropower in Ecuador: A GIS-based framework of analysis (2018).
93. *Brendan O'Neill*: Multicriteria Site Suitability for Algal Biofuel Production Facilities (2018).
94. *Roman Spataru*: Spatial-temporal GIS analysis in public health – a case study of polio disease (2018).
95. *Alicja Miodońska*: Assessing evolution of ice caps in Suðurland, Iceland, in years 1986 - 2014, using multispectral satellite imagery (2019).
96. *Dennis Lindell Schettini*: A Spatial Analysis of Homicide Crime's Distribution and Association with Deprivation in Stockholm Between 2010-2017 (2019).
97. *Damiano Vesentini*: The Po Delta Biosphere Reserve: Management challenges and priorities deriving from anthropogenic pressure and sea level rise (2019).
98. *Emilie Arnesten*: Impacts of future sea level rise and high water on roads, railways and environmental objects: a GIS analysis of the potential effects of increasing sea levels and highest projected high water in Scania, Sweden (2019).
99. *Syed Muhammad Amir Raza*: Comparison of geospatial support in RDF stores: Evaluation for ICOS Carbon Portal metadata (2019).
100. *Hemin Tofiq*: Investigating the accuracy of Digital Elevation Models from UAV images in areas with low contrast: A sandy beach as a case study (2019).
101. *Evangelos Vafeiadis*: Exploring the distribution of accessibility by public transport using spatial analysis. A case study for retail concentrations and public hospitals in Athens (2019).

102. *Milan Sekulic*: Multi-Criteria GIS modelling for optimal alignment of roadway by-passes in the Tlokweng Planning Area, Botswana (2019).
103. *Ingrid Piirisaar*: A multi-criteria GIS analysis for siting of utility-scale photovoltaic solar plants in county Kilkenny, Ireland (2019).
104. *Nigel Fox*: Plant phenology and climate change: possible effect on the onset of various wild plant species' first flowering day in the UK (2019).
105. *Gunnar Hesch*: Linking conflict events and cropland development in Afghanistan, 2001 to 2011, using MODIS land cover data and Uppsala Conflict Data Programme (2019).
106. *Elijah Njoku*: Analysis of spatial-temporal pattern of Land Surface Temperature (LST) due to NDVI and elevation in Ilorin, Nigeria (2019).
107. *Katalin Bunyevácz*: Development of a GIS methodology to evaluate informal urban green areas for inclusion in a community governance program (2019).
108. *Paul dos Santos*: Automating synthetic trip data generation for an agent-based simulation of urban mobility (2019).
109. *Robert O' Dwyer*: Land cover changes in Southern Sweden from the mid-Holocene to present day: Insights for ecosystem service assessments (2019).
110. *Daniel Klingmyr*: Global scale patterns and trends in tropospheric NO₂ concentrations (2019).
111. *Marwa Farouk Elkabbany*: Sea Level Rise Vulnerability Assessment for Abu Dhabi, United Arab Emirates (2019).
112. *Jip Jan van Zoonen*: Aspects of Error Quantification and Evaluation in Digital Elevation Models for Glacier Surfaces (2020).
113. *Georgios Efthymiou*: The use of bicycles in a mid-sized city – benefits and obstacles identified using a questionnaire and GIS (2020).
114. *Haruna Olayiwola Jimoh*: Assessment of Urban Sprawl in MOWE/IBAFO Axis of Ogun State using GIS Capabilities (2020).
115. *Nikolaos Barmpas Zachariadis*: Development of an iOS, Augmented Reality for disaster management (2020).