AR Door to the History of TCD

Amar Zia Arslaan B.Tech

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Augmented and Virtual Reality)

Supervisor: Aljosa Smolic

August 2019

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Amar Zia Arslaan

August 14, 2019

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Amar Zia Arslaan

August 14, 2019

Acknowledgments

I would like to express my gratitude to Prof. Aljosa Smolic, Iman Zolanvari and Alan Cummings, for their expertise, assistance and guidance. Without your support this dissertation would not have been possible. I'd like to thank my classmates: it has been a pleasure. My deepest appreciation belongs to my family for their patience and understanding.

Last but not the least, my sincere thanks goes to my friends for continuously helping me maintain a positive outlook, whenever I faced any challenges throughout my time here in Trinity.

Amar Zia Arslaan

University of Dublin, Trinity College
August 2019

AR Door to the History of TCD

Amar Zia Arslaan, Master of Science in Computer Science
University of Dublin, Trinity College, 2019

Supervisor: Aljosa Smolic

Utilising publicly accessible frameworks and tools, this application based research contrives the design and implementation of a mobile application that enables full scale simulation of the Front Square of the Trinity College Dublin (TCD) campus using Augmented Reality (AR). The application generates a Gateway (Portal) on any valid detected surface that is tapped, through which people can walk into a virtual 3D environment representing the Old Trinity Campus. The 3D mesh of the Trinity Campus also features virtually simulated crowd in vintage costumes that walks inside the 3D mesh of the Front Square, which is precisely overlaid on top of the real campus. As the buildings have not vastly changed since the 18th century, the environment looks more realistic but with completely different articles augmented to the real world, providing a convincing illusion of the old TCD. This research also discusses the optimisation of 3D meshes for mobile AR and the investigation carried out to determine the better approach between marker based and marker-less AR for similar applications.

Summary

The dissertation is structured as follows: first the motivation and objective of this research is discussed. Then the background information will be discussed, followed by an elaborate discussion of State of the Art in this research problem. Starting from discussion of the technologies behind Augmented Reality, the design of the proposed system will be laid out and every component will be discussed in detail. Additionally, the approaches tried in the experimentation phase will be discussed, and how they align with our goal of solving this research question. Further, there will be a detailed examination of the implementation, followed by defining and validating performance metrics and analysis of the results. Furthermore, there will be a deliberation of the limitations and discussion about the results. Finally, a conclusion will be given followed by future scope of this project.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter contains some of the essential background knowledge about the topic that is studied, with a brief discussion about the relevancy and probable impact of the same. Further, the objective and structure of the dissertation hereby submitted are also discussed.

1.1 Motivation

Technologies such as Augmented Reality (AR) and Virtual Reality (VR) have been the center of scientific research for a few years now. Until recently, VR was considered the more popular and widely available out of these computer graphics and vision technologies. However, the release of games such as Pokémon GO [1] and devices like Hololens [2] have been introducing an increasing number of people [3] to AR and it is slowly moving towards mainstream adoption, outdistancing VR to become the more popular amongst the two technologies. There are a number of studies published which explore the different domains and possibilities of AR. However, not a lot of relevant research has been carried out to determine the scope of superimposing full scale 3D models of buildings on top of the original structures using AR, provided the limited computational power and tracking capabilities of handheld mobile devices. Similarly, has there been a substantial research on the use of shaders inside Unity with respect to AR and using it to generate something like an AR portal.

Generally, the concept of AR is to project the virtual entity on a screen, which can be from a handheld device like a mobile phone or a tablet, onto a real-world platform. Further interactions can be performed on it via the screen or using hand gestures. This technology can be beneficial from a historical point of view because it can enable users to witness famous historical personalities and monuments while preserving history, and acting as a great tourist attraction simultaneously. As further academic and commercial research is transpiring, AR is expanding and becoming more accessible to the masses. Recent use of this technology has been employed in Education (eg. Teaching difficult concepts like Electromagnetism [4]), Medicine (eg. Assisting the surgeons during surgery [5]), Manufacturing (eg. AR aided product assembly [6]), Marketing (eg. E-Commerce [7]), Entertainment (eg. Video games [8]) and even Psychology (eg. Therapy for people with phobias [9]). The AR portal can also have applications in real estate [10] where the properties can be viewed in an entertaining and interactive way without the need to travel. As mentioned above, use of this technology can be inclined towards both educational and recreational purposes in conjunction, such as interactive learning for students who would be interested in learning architecture or studying history.

1.2 Objective

The primary intention of this dissertation is to integrate full scale 3D models of cultural heritage sites into the real world using Handheld Mobile Augmented Reality (HMAR) and explore the variables involved in working with the corresponding research problem. The study seeks to investigate the different approaches that can allow users to experience full scale 3D models into the real world environment. The process involves examining different AR technologies available in the handheld AR ecosystem, and discovering the one that provides the most efficient solution to this research question. In addition, prospective solutions to the registration problem in AR are analysed, implemented and evaluated, and a stable solution is proposed where the placement of markers are not required.

As a result, an inter-dimensional portal is developed which allows the user to walk inside through a virtual door and experience the historical 3D model of the Front Square of Trinity College Dublin (TCD). The aforementioned model is in full scale and features crowd simulation where the agents are dressed in vintage clothes, thus enabling the user to experience the campus virtually in its former glory during the 18th century. As the campus buildings have not vastly changed since the 18th century, the environment looks more realistic but with completely different articles augmented to the real world, providing a convincing illusion of the old TCD. This research further analyses and focuses on the possibilities this ecosystem coupled with various interfaces facilitating AR provides us with when we try working with APIs and Shaders inside Unity. Furthermore, defining and evaluating performance metrics for crowd simulation and subsequently validating the generated results is a significant portion of this research. If we consider external factors like fluctuating lighting conditions due to variable weather conditions, tourists constantly obfuscating views, and inconsistent computer vision techniques for AR registration; there are numerous challenging technical pitfalls in implementing the visualisation of 3D content in outdoor environment using AR. Being an application based dissertation, it will cover several different topics and associated hurdles, treating them as our secondary research questions. Examples of such secondary objectives are - Ensuring that the user is present physically at the right place at the Front Square of TCD campus (absolute positioning), construction of Nav Mesh for crowd simulation and instantiation inside the 3D model, optimisation of models by using lighter mesh, maintaining consistency with the crowd animation played within the virtual environment, ensuring that the crowd animation being played aligns accurately with the point of view of the user. Other issues that need to be considered would be related to the crowd simulation - whether the simulation would be a persistent story, that is, will it continue even if the user isn't watching it, or will it be a story that resets every time a user runs the application. The research also aims to explore whether or not multiple users will be seeing identical crowd simulation when using the application concurrently.

1.3 Outline

This dissertation is divided into 6 chapters, each dealing with one specific section in detail.

- Chapter 2 overviews the current State of the Art in AR, and discusses in detail preliminary and ongoing literature that was reviewed throughout the duration of this dissertation.

- Chapter 3 discusses the design and pipeline that was used to solve the research question.
- Further, Chapter 4 focuses on the theoretical background and implementation.
- Chapter 5 analyses and discusses the results and evaluation of this study.
- Finally, Chapter 6 draws the conclusion, summarises the proposed contributions and reviews the limitations of this study, and further presents the scope of improvements and future work.

Chapter 2

Background and State of the Art

This chapter reviews the current State of the Art in Augmented Reality. As AR is a vast field which covers a broad range of topics, we will focus on the topics that are imperative for the reader to understand the latest advances in AR and underlying concepts in this dissertation. It begins by formally defining AR and categorising the different types of AR that are available today. Further, the impact of this technology and its recent applications in cultural heritage and tourism are examined. Finally, the technical background and related work are discussed.

2.1 Definition of Augmented Reality

The term Augmented Reality (AR) was coined by a former Boeing researcher named Tom Caudell in the year 1990. Although the concept of augmenting the real world by virtual data was initially used by a number of applications in the late 1960s and 1970s [11], this was the first time someone had conceived it officially. The term AR has been given various different meanings [11]. Milgram et al. [12] mention a more restricted definition where AR is seen as "form of virtual reality where the participant's head-mounted display is transparent, allowing a clear view of the real world". But Wu et al. [13] do not believe that AR is restricted to any type of technology. Accordingly, Diegmann et al. [11] broadly define AR as "a situation in which a real world context is dynamically overlaid with coherent location or context sensitive virtual information" [14] and consider AR as a concept which is conceptualised beyond technology. Azuma

[15] defines AR as a variations of Virtual Environments (VE), which combines real and virtual imagery, is interactive in real time, and registers the 3 Dimensional imagery with the real world. There can be many different applications of AR in various different domains. Zhou et al. [16] define AR as a technology which allows real-time overlaying of computer generated virtual imagery on physical objects. Unlike virtual reality (VR), where the user is completely immersed in a virtual environment, this research defines AR as a technology that allows the user to interact with the virtual images using real objects in a coherent manner.

2.2 Types of Augmented Reality

Stewart et al. [17] classify AR technology into 6 types, all belonging to either Triggered or View-Based category -

2.2.1 Triggered

Triggers are the external stimuli or characteristics that initiate or "trigger" the augmentation [17]. In AR, triggers can be GPS locations, markers(paper or object), dynamic augmentations of objects, as well as a combination of dynamic augmentation of objects and GPS locations.

2.2.1.1 Marker Based

Marker based AR requires a marker to activate an augmentation. Visual markers or objects are tracked to determine the camera's perspective, and virtual objects are placed on top of those markers. One limitation of this method is that the location of the marker must be known at all times, and tracking stops as soon as the marker disappears from the view frustum. Marker based AR can further be sub categorised into the following -

2.2.1.1.1 Object Markers Based

When the AR is object based, the object could be any reasonable sized physical entity like a real-world $20 bill [17], which can activate an entertaining, patriotic animation.

Sometimes augmentation created through object based markers may be used to enhance the image or object, and sometimes it can just be a means to access the digital content, using objects that may or may not be related to the augmentation. An example of Object Marker Based AR in use can be HP Reveal [18], previously known as Aurasma.

2.2.1.1.2 Paper Marker Based

Paper markers share most of the characteristics of the object markers. The only difference being the physical properties of the marker. Here it is not a real-life object but a pattern printed on the paper. Mostly the pattern is something unique so that it can be easily tracked by the application to produce an augmentation. A few examples of applications of Paper Marker Based AR are String [19] and Blippar [20]

2.2.1.2 Location Based

AR technology makes it easy to visualise spatial information. It could be the position or direction of locations and objects for location based applications that process and present information based on the user's position in the real world [21]. Whenever the information to the user is processed and presented, location based systems take the user's location into account. This method uses a digital compass or GPS technology to track the device on a specific location, and is usually helpful for overlaying information that refers to nearby services or directions. Real life usage of Location Based AR systems is very well demonstrated by Instagram [22].

2.2.1.3 Dynamic Augmentation

Dynamic Augmentation is a sub category of AR where meaningful and interactive augmentation takes place. There is also a possibility of motion tracking and object recognition [17]. An example of Dynamic Augmentation in real world is FaceCake [23].

2.2.1.4 Complex Augmentation

Complex Augmentation is an extension of Dynamic Augmentation, where the dynamic view is augmented and information is pulled from the internet, based on the markers, location or object recognition [17]. One prime example of this technology is Google Glass [24]

2.2.2 View-Based

Other forms of AR are view-based, which includes either digitised augmentations without reference to what is in view or augmentation of a stored/static view. View based AR can be subdivided into 2 categories -

2.2.2.1 Indirect Augmentation

Indirect Augmentation involves images of the real world augmented intelligently [17]. Google's cross platform application named "Just a Line [25]" is a great example of Indirect Augmentation which allows the user to draw or doodle in the air using AR.

2.2.2.2 Non-specific Digital Augmentation

Non-specific Digital Augmentation can augment any camera view regardless of location. An example of the same is an IPad game called VirtualPop [26]

2.3 HMAR in Cultural Heritage and Tourism

The most easily accessible AR medium for cultural heritage is a mobile handheld device like a smartphone or a tablet [27]. The fact that these devices are cheap and easily available makes them more accessible and makes a strong case for their use with AR technology to make use of the pre-existing hardware. Ding [28] reviews some of the recent mobile handheld AR applications for cultural heritage as discussed below.

2.3.1 ArtLens 2.0

ArtLens 2.0 [29] is a cross platform application(available for both IOS and Android) by Cleveland Museum of Art [30]. It is a small and lightweight application which enhances the visitor experience by providing facts and information about the artwork using AR, offering access to interactive real-time maps of the museum and the option to design individual tours. They are able to guide the users indoor through the application using mapping and beacon technology, helping the visitors explore the museums. The version 2.0 of the application is a much improved version of the original application, and uses the database of the museum to recognise art. As a result, as soon as a painting is added to the museum, it automatically gets updated in the application as well.

2.3.2 Skin and Bones

A free application for iPhones and iPads, called Skin and Bones [31] by Smithsonian's National Museum of Natural History [32] is a great example of use of AR technology in preservation of historical artefacts. Aiming to improve the Bone Hall Exhibition [33], which has not been upgraded in a very long time, the application brings the skeletons to life using AR. Before the release of this application the museum used to display information on small signs near the respective skeletons, which being in a largely scientific language made it difficult for the visitors to understand them. The application can be used from anywhere in the world. When the users are not physically present in the museum, they need to open the application and point their device's camera towards specially printed papers or displays, acting as markers, in order to give the users a 3-Dimensional experience. It can also visualise the way the animals looked like and how they moved when they were alive. The museum reports that the visitors using the Skin and Bones application while visiting the museum tend to spend much more time in the Bone Hall, demonstrating an average dwell time of 14:00 min with the application as compared to 1:34 min without the application.

Figure 2.1: Skins & Bones app by Smithsonian's National Museum of Natural History

2.3.3 Blanton Museum of Art

The application for The Blanton Museum of Art [34] at the University of Texas in Austin [35] is the third and last application discussed by Ding. An exhibition called

"The Crusader Bible: A Gothic Masterpiece" is present in this University's museum, which displays the elements of the exhibition in only three languages: Latin, Persian and Judeo-Persian. For obvious reasons, English speaking visitors find the exhibition hard to understand. So an iPad application was developed for translating the exhibition elements into English language using the AR technology. The percentage of visitors using this application was not very high, at about 20%, simply because not all the visitors owned iPads, but the response to the application was still positive and the dwelling time of the visitors was longer compared to the times before this application was released.

2.3.4 Taipei Fine Arts Museum

Chang et al. [36] give another example of handheld mobile AR for cultural heritage. In 2012, they conducted a user behaviour experiment during an exhibition where the visitors to Taipei Fine Arts Museum [37] were asked to use their art appreciation AR application. The hardware used for the experiment was a 10 inch Android tablet, which the visitors had to point towards a painting, and then a series of events used to take place in the following order - The first step was the description step where the application provided an audio or visual description of the painting. The second step involved analysis, where the users could analyse the painting and get detailed information about it by zooming in/out. The third step was interpretation, where the users could zoom in/out of the painting again and could get info about the general theme of the painting and how it was composited. The final step was judgement where the visitor could use his/her knowledge about the previous painting to assess the other similar paintings recommended by the application. This experiment involved 135 college students, divided in three groups: an AR guided group, an audio guided group and a control group which had no guide at all. The experiment focused on variables like behavioural patterns, flow, learning, time spent on paintings, acceptance and user attitude towards the guidance systems. The results demonstrated that the AR guided group exhibited a significant increase in all these aforementioned variables and showed higher rates of acceptance towards the AR guided systems. The results in behavioural patterns of the audio guide group and the AR guide group were comparable. On the other hand, the non-guided group expressed a willingness to use a guide in the future, either audio or AR. The audio group showed interest in the AR guided systems, but was open to use the audio guided system again. Both the guided groups indicated interest in using the guiding systems again. Moreover, the non guided group revealed that they wished for a guide as it would have helped them better understand the paintings.

2.3.5 Melaka heritage site

An AR application named as AR@Melaka is another example of AR application for a cultural heritage site, and is proposed and developed by Pendit et al. [38] in their study published in the year 2014. The application aims at providing the users an enjoyable informal learning experience at the Melaka heritage site [39] in Malaysia. The user is provided with an overlay of information on top of the real world, and it annotates certain positions of the heritage site, which the users can click on to access more detailed information. Once the application development was completed, 200 visitors participated in the user testing, where they are asked to use the application and fill out a questionnaire thereupon. The questionnaire, consisting of 24 questions about informal learning, had a scoring system on a scale from 1 to 7, where 1 was the lowest and 7 the highest. The average score was approximately 5.5, meaning that the visitors learned something while using the AR application and enjoyed the process. The results the researchers obtained were limited in scope as there was a lack of a metric to measure the extent to which the mobile AR application enhanced level of enjoyable informal learning. Hence the study was not very conclusive, thus making it harder for the readers to draw conclusions or replicate the study.

Figure 2.2: Live View of the Mobile AR Application at Melaka Heritage Site

2.4 State of the Art

2.4.1 Mobile Augmented Reality

AR technology has several subcategories, like Mobile Augmented Reality (MAR) and Handheld Mobile Augmented Reality (HMAR), and any type of technology a user can carry to visually augment the world can be classified as belonging to one of these categories. At present, HMAR it is mainly being developed by Apple and Google. In the late 2017, both the companies released Software Development Kits (SDKs) for their respective Operating Systems (OS) to enable developers build applications that work on AR. Google's ARCore is a state of the art AR platform compatible with Android that can run on most of the newer smartphones. A majority of the Android devices are released with ARCore support nowadays. With the help of the camera and onboard sensors of a supported smartphone, ARCore is able to achieve basic light estimation, motion tracking and environmental understanding. Following the same suit, Apple released a similar platform for their iPhone Operation System (iOS) and named it ARKit. In fact, ARKit was released by Apple a few months earlier and thus facilitating AR development on IOS devices first, later followed by the release of ARCore for Android devices by Google. In the background, AR toolkit development was being pursued by small corporations such as Vuforia since the year 2011, where they released Qualcomm's AR platform called QCAR version 1.0 in year 2011, after elaborate beta testing. Later Vuforia version 2.0 was released and since then it has gained strong user base and developer support. Focusing primarily on marker based AR, Vuforia was later acquired by PTC [40] in late 2015. The strong backing by tech giants like Google, Apple and PTC has led to a tremendous increase in the popularity of AR in the last few years, and survey reports that validate this trend will continue in the future. For Example, Augmented and Virtual Reality Report 2019 by Perkins Coie [41] reveals that out of the 200 startup founders surveyed, 70% predicted that AR market will surpass VR in revenue. Additionally, 81% speculated that this is possible in the next 5 years. The following sections will give an overview of some of the technologies and processes that enable the production of AR experiences with ARCore, ARKit and Vuforia in order to understand the capabilities of these platforms and their limitations. This segment discusses some of the background behind Mobile Augmented Reality and the technologies that make possible to experience it in our smartphones. The first implementation of smartphone AR took place in early 2004, with some demonstrations of real-time tracking of 3D markers [42]. Due to advancements in mobile computing since then, AR can now have unlimited potential use cases and be distributed to a very large number of people.

2.4.2 MAR for integrating full scale 3D models in real world

Appreciable research has been carried out in this area of study in the past, with researchers devising various methods of achieving accurate full scale augmentations on monuments. One example of such a method is discussed in an article published by Vlahakis et al [43], who, along with his colleagues developed a tool called Archeoguide, which stands for Augmented Reality-Based Cultural Heritage On-Site Guide. The article demonstrates how a system that's designed well can work as a personal electronic tour guide to historical heritage sites. The system enables the users to collect and exploit archaeological information on any location. The system is primarily based on optical tracking techniques. It uses DGPS for location accuracy, but still gets alignment($<$ 1m) and viewing angle($<$ 0.5 degrees) errors. The system uses a Client–Server architecture, so network access is essential for the experience to work, and it requires complex and non portable hardware to be able to function(Figure 3.5). This is the biggest shortcoming of this system; and the kind of infrastructure it requires is impractical for general public use. For the Archeoguide experience to work, the user needs a Head Mount Display (HMD), DGPS reciever, camera, compass and a very powerful laptop. However, in spite of these limitations the output achieved by this system is commendable and does move a step forward towards finally integrating AR in real life using mobile devices(2.3). One other solution is provided by Panou et al. [44], who collectively attempted to restore the demolished minaret of a glass mosque in Crete, Greece using AR. The study was published in the year 2018 so it is fairly recent, and it works on a similar approach as Archeoguide. However, Panou and his colleagues designed a location-based application which works on mobile phones and does not require any specialised or complicated hardware. The researchers primarily follow a geo-location and sensor based approach. The 3D augmentations (Figure 2.4) are a minor component of the architecture they built, and access to the server required throughout the session of the application. While this technique is more advanced and can cater to the public, it doesn't come without limitations. Occasionally, the researchers noticed inconsistency between the device orientation calculated by them and orientation calculated by the API. In addition, it was reported that the use of AR Camera in conjunction with GPS for longer periods resulted in rapid battery consumption. Finally, the researchers did not address the problem of memory efficiency or optimisation in this study, which is a critical matter that requires consideration for all kind of HMAR experiences, attributed to the limited memory and computing power of these devices.

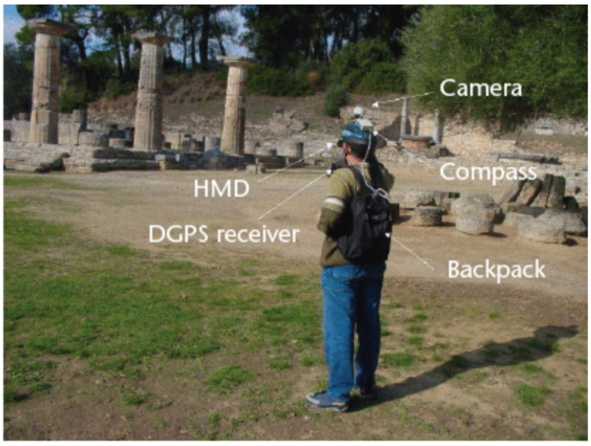

Figure 2.5: User of Archeoguide at a viewpoint with the necessary equipment

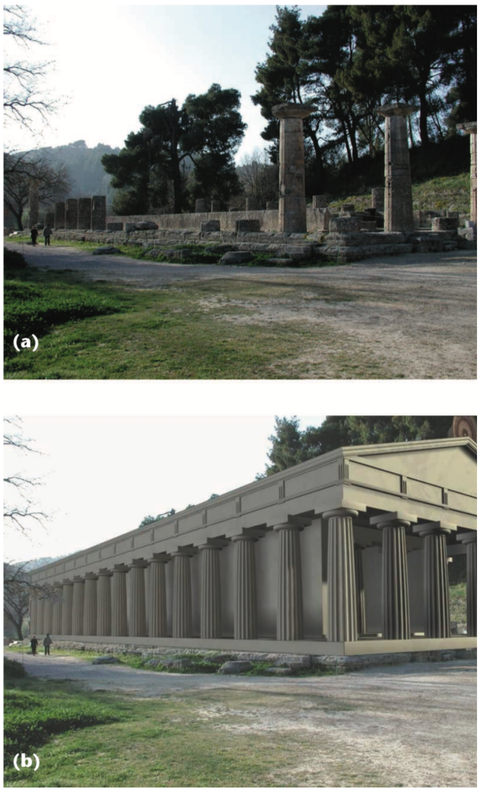

Figure 2.3: Renders by Archeoguide (a) Current state of the temple of Hera (b)Temple augmented with a rendered model with live video feed

Figure 2.4: (a) The Ottoman Glass Mosque in its current state (b) 3D depiction of the Glass Mosque featuring the now demolished minaret, as seen by the mobile's camera

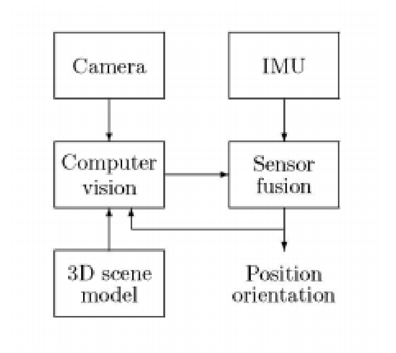

Figure 2.6: Diagramatic representation of Hol's Sensor Fusion approach

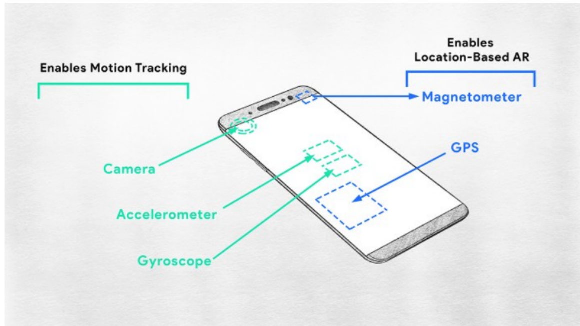

Figure 2.7: Mobile sensors required to make various AR technologies work

2.4.3 Motion Tracking

To describe the concepts of motion tracking, it is important that the reader is aware of the concept of Simultaneous Localisation and Mapping (SLAM). SLAM is the method of recognising the device position and rotation by mapping the surroundings. The block diagram in Figure 4.1 depicts the SLAM processes. It was a pretty popular topic among the mobile robotics community before it started getting used in smartphones. SLAM can be implemented using various different algorithms, but fundamentally they all gradually build a uniform map of the surroundings while establishing the device location within the map synchronously [45]. Since the introduction of SLAM, various algorithms have been presented to put this technique into practice [46]. As mentioned above, due to the high computing power of mobile devices nowadays, SLAM algorithm shows reasonably consistent results in smartphones. Klein and Murray [47] presented the first implementation of SLAM in mobile phones in the year 2009. A keyframe-based SLAM approach was suggested by them which could be run on smartphones just using camera input. However, if only images are used for 3D tracking, the results could be inaccurate.

This is where sensor fusion technology [48] comes into play. Illustrated in Figure 2.6, this technology combines the data of different sensors in order to have more precise and comprehensive information of the device, providing an overall improved motion tracking as a result. Lynen et al. [49] improve upon the basic SLAM algorithms with sensor fusion technology, increasing the accuracy and robustness of the results. To understand the device rotation and position in real-time, data is gathered with the help of Inertial Measuring Unit (IMU), which consists of an accelerometer and a gyroscope. It helps us acquire the acceleration, velocity, angle and position of the phone. Also, since IMUs do not provide reasonably precise tracking data, the image stream from the camera becomes necessary in this approach. By distinguishing characteristic visual attributes on every input image and then tracking them using SLAM, the camera pose can be recovered. The method of computing IMU data and images for motion tracking is called Visual-Inertial Odometry (VIO). However, this technique has a drawback: It does not stay at the same location as time goes by and objects tend to move away from there originally detected position. To counter this, SLAM has optimisation processes which help VIO in detecting previously visited areas and reduces map errors as a result, in a process called loop closure detection [50]; along with increasing the precision of the scene map in the background.

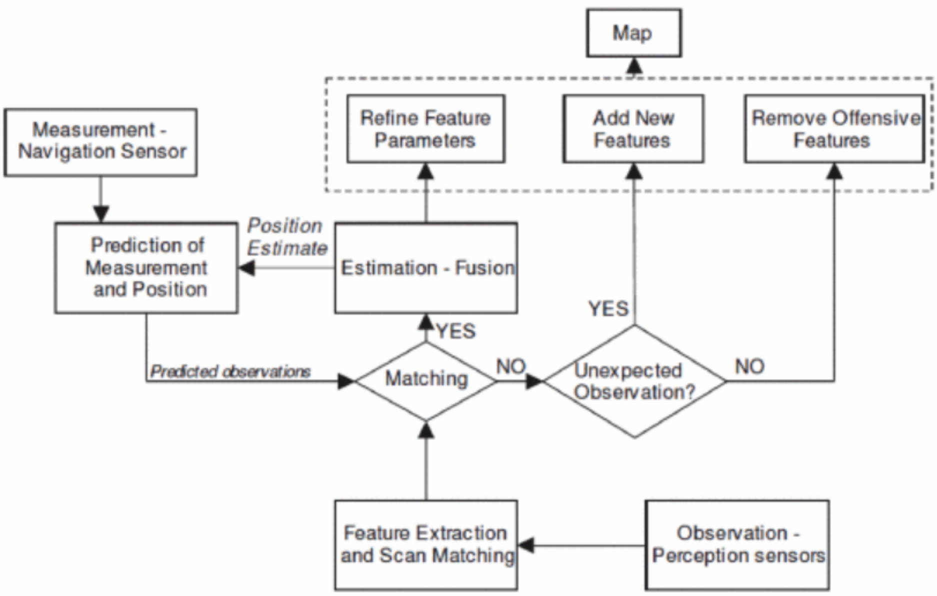

Figure 2.8: Block Diagram of SLAM Process

2.4.4 Light Estimation

Mobile AR processes a given frame to identify the illumination of its surroundings. Currently, AR technology can only make a global estimate of the lighting, estimating features like brightness, color, temperature. On the other hand, Google's ARCore calculates an average of incoming light by scanning the pixels of the camera image, which helps it best light an AR object according to the lighting situation of the given environment [51]. Light and shadows play a major role in helping the eye to determine and differentiate fake objects from real ones. If the light is matched correctly, the brain cannot tell the difference between a real object or a 3D model placed in the scene, which in turn makes the AR experiences more believable and enjoyable.

2.4.5 Scene Understanding

To place virtual objects on the scene, we need to be aware of the 3D representation of the real world (Smartphone 3D Reconstruction for Physical Interactions in AR, 2018). Once the camera pose has been tracked efficiently, the HMAR frameworks detect both vertical and horizontal planes in the scene. Polygons are used to define planes and they rest on top of the real-world surfaces which helps creating the illusion of virtual objects. We can achieve robust plane detection by detecting subsets of coplanar feature points from the monocular camera image [52]. In contrast, we can estimate surface normals [53] to improve the data of detected point cloud. This process is computationally expensive and is only preferred when the visual points are available in abundance.

2.4.6 Immersion

We can define immersion as a sensation of involvement in the virtual world [54], and it is linked to a mental sense of involvement. Mainly, the following three technical factors [54] affect digital immersion:

- Physical interactions: All the interactions between the physical world and the virtual objects need to be realistic. Otherwise the immersion effect may be lost, degrading the overall experience for the user.
- Hardware: Capable hardware is very important in giving the users experiences that feel realistic and engaging. Apart from that, smartphone AR has its own limitations in terms of the small amount of real world space it can cover when compared to other available devices like headsets and glasses, which present a

much more immersive experience, as the field-of-view (FOV) is increased and visual boundaries between physical and digital world are minimised.

- Visual perception: The way virtual objects are rendered in the real world environment can drastically alter the experience of the user [55]. A better visual perception can be achieved in two ways - Either by rendering augmented information as realistic as possible to be seamlessly integrated with the environment, or, the real world can be stylised to match the virtual objects to generate a consistent environment [56].

2.4.7 Crowd Simulation

According to literature [57], researchers like Challenger et al. [58] use the word 'crowd' for almost every situation involving an interaction between more than two individuals. According to several authors, the word crowd is used as a description for a large number of individuals walking through the same space at a certain point in time. However, the term has different meanings for different researchers, and thus has been used with varying definitions in different papers. For example, Hoogendoorn and Bovy [59] use the term 'crowd' to define a number of individuals walking through a train station, while the same word is used to describe the movement of pedestrians in front of a pop podium by Duncan [60]. According to Duives et al. [57], the term has no common definition in the field of transportation engineering. Even in sociology, where crowds have been researched for many years, by researchers like Wijermans [61] and McPhail [62], the definition used is quite broad: 'More than 2 people at the same location during the same time period'. This definition does not give a quantifiable reference point for engineering practices. It is very vague in the sense that any movement involving interaction of more than two individuals could be construed as a movement of crowd by this definition. Duives et al. [57] finally use the following definition of the term 'crowd' as their working definition -

"*A crowd is a large group of individuals ($N \geq 100$ P) within the same space at the same time whose movements are for a prolonged period of time ($t \geq 60$ s) dependent on predominantly local interactions (k P 1 P/m2).*"

The numbers N (number of individuals), k (density) and t (time) are chosen in a way as to exclude movements of non-existent interaction. The researchers get more specific with this definition by elaborating more about the crowd. They assume that pedestrians are in close contact with each other (interaction distance between individuals being less than 1 m) during crowd movements, making multiple split- second operational movement decisions. Secondly, the pedestrians that are a part of the crowd are in no external pressure to advance, but they do have a conjectural objective in mind towards which they are proceeding. Thirdly, it is assumed that the environment in the location where this crowd is considered, is friendly. As a result, no additional tension between the individuals exists. Furthermore, the crowd is assumed to be heterogeneous with no specific age-group or gender dominating, since we are discussing the prediction of crowd movements by simulation models at a public space where mixed demographics can be present. Since pedestrians in public places generally do not carry a lot of luggage with them, it is assumed that they are not carrying baggage other than a small backpack. Lastly, it was assumed that the pedestrians are not familiar with the layout of the infrastructure. When translating the above into engineering terms, the definition of 'crowd' gets a more direct meaning for pedestrian motion modelling.

2.4.8 Level of Detail

As the name rightly suggests, the concept of decreasing the details in a 3D model as it moves away from the viewer in space, is called Level of Detail (LOD). It also takes into account factors like the relevance, resolution and speed of the object. There is an implementation of the similar concept in Computer Graphics called mipmaps [63], where the computational cost of graphics pipeline is decreased by decreasing the number of polygons that need to be rendered. Apart from relieving heavy rendering, LOD can also omit the computation of expensive tasks.

2.4.9 Mobile Hardware Restrictions

Although Mobile AR is the most promising way to make this technology accessible to as many people as possible, it does have some limitations. The major issue with Mobile AR is the limited capabilities of the mobile hardware. This limitation is highlighted in the approaches discussed in 2.4.2, where the users of Archeoguide had to wear a kit consisting of a Head Mounted Display as shown in Figure 3.5, DGPS (Differential Global Positioning System), camera, compass and a backpack containing a powerful laptop - just to be able to experience full scale 3D models augmented to the ruins of the ancient architecture.

Low memory and insufficient computing power along with limited mobile network connectivity are all factors that need to be considered whenever an AR tourism application is developed [64]. To circumvent this problem, either the computationally expensive tasks can be offloaded to an external server, or the efficiency of those tasks can be improved. We can find examples of the former technique implemented by researchers like Fatima et al. [65], Gui et al. [66], and Zhang [67] where computationally expensive tasks like image recognition were offloaded to an external server. While the other options are available as mentioned above, this is the most popular way of implementing AR applications that require image processing. The only limitation is that the application becomes dependent on internet connectivity. Also, due to the memory limitations of smartphones, once the scene map becomes too big, the system needs to remove some of the information which will no longer be used for the reconstruction

Chapter 3

Design

The design of the proposed method is discussed in this chapter. Since this research expands on the 3D model of Trinity College Dublin created in the year 2017, the sections below will explain how to data set was acquired, and discuss what optimisations and processing has been applied to it. Further, each section in this chapter will focus on the design principles and methodologies considered before finalising a solution, and will explain the reasoning behind the same.

3.1 Dataset

The following dataset was utilised to exhibit the 3D mesh of the Front Square of TCD Campus inside the AR portal - Trinity College Dublin Drone Survey Dataset. The dataset used for this research was acquired from Trinity Access to research Archive (TARA) [68]. The aforementioned dataset comprises of 6 files in the form of zipped archives and a report explaining the technical details of the project. The files named Images 1, Images 2, Images 3 and Images 4 contain the original images from the drone survey, and an archived file called 3d_mesh contains the 3D model of the TCD campus in Wavefront Object (OBJ) and Polygon File Format (PLY) file formats. Further, the data points for the point cloud that was used to create the OBJ model are contained in the zip file named point_cloud. The 3D visualisation of the point cloud can be generated using an open-source WebGL based point cloud rendering tool called Potree [69].

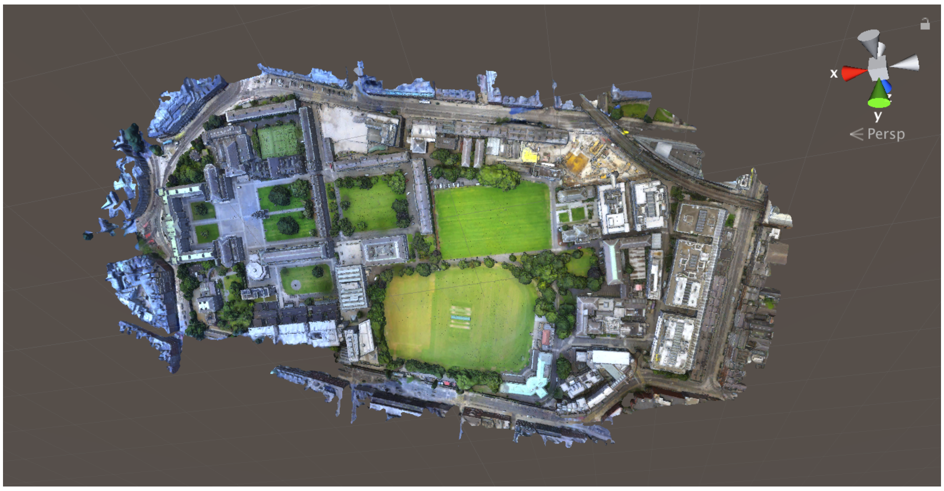

Figure 3.1: Trinity College Dublin Drone Survey Dataset

3.2 Experimentation

In order to address the aforementioned research question, a series of approaches was considered and implemented. The following section discusses the merits and limitations of each approach and explains why the given approach was further pursued or not.

3.2.1 Marker Based AR vs Markerless

To institute the development process in the early stages of the research, an important distinction was required to be performed between the two major categories of AR available in the market today. As the section 2.2 expands on the subject in detail, we'll just discuss the abstract choice available between Marker based AR or Markerless AR. To summarise the two, Marker based application of AR involves the use of visual markers like 2D images and barcodes to align virtual data with their real world surroundings. In contrast, as the name suggests, Markerless application of AR integrates 3D virtual objects in the environment in real-time and constantly tracks the environment to detect feature points and planes, in order to provide as realistic AR experience to the user as possible.

For this research, Marker based AR approach was considered due to higher position accuracy compared to Markerless AR, as discovered by Cheng et al. [70]. Also, according to Genc et al. [71], the overall robustness of the application is increased

Figure 3.2: ARToolkit Marker Samples

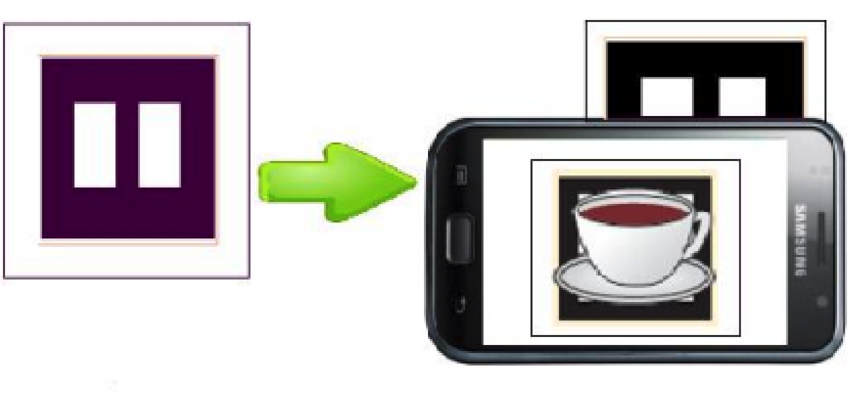

Figure 3.3: Example of markers being used in MAR

and computational requirements are reduced when markers are used. The tool used to conduct the experiments is called Vuforia [72], because it is the most popular Marker based AR tool, and supports almost all smartphones available in the market. The section below discusses in detail the approaches considered to solve the given research question.

3.2.2 Vuforia

Vuforia is an augmented reality software development kit (SDK) for mobile devices. It provides a high-level interface and enables users to easily create AR experiences and applications for Android, iOS, and Universal Windows Platform (UWP) devices. Considered to be the most popular Marker based AR development SDK, Vuforia uses computer vision to recognise and track planar images, also known as image targets, and small 3D objects, along with 3D target types including Markerless image targets and fiducial markers. To develop the required solution using Vuforia SDK, an additional developer tool is required, the supported tools being Gradle, Android SDK Build

Tools, Android Studio and Unity Editor. Also, it is essential for the developer to have an account on the official Vuforia Developer Portal, in order to generate development key for the project. Our developer tool of choice is Unity Editor, a cross platform game development engine. We use version 2018.3 because it comes with Vuforia already integrated in the engine, and it is the first version of Unity to support Navmesh components, thus enabling crowd simulation in our application. Generating AR experiences using Vuforia involves a few fundamental steps. The first step is to generate a license key on the Vuforia Developer Portal, and copying the key as it required for Vuforia to work inside Unity. The second step involves acquiring a target that can be an image or in the form of a cuboid, cylinder or 3D object. Further, the third step involves adding a database in Vuforia Target Manager. In the fourth step, the target is added to the database, and a Unity Engine compatible file is obtained. The next step consists of procuring a 3D model that is desired for generating the AR experience. Finally, a project is created inside Unity and image target and the database are imported.

3.2.3 Vuforia Image Detection

This approach involved using a real life photograph of the Front Square as image target, and generating the 3D model using the dataset mentioned in section 3.1. Opposed to the expected result, the model spawned at full scale but demonstrated constant jitters. A probable reason for this unexpected behaviour could be the difference in the lighting situations when the original picture was recorded and when the test was performed. This phenomena, also known as pose jitter, is a known limitation of Vuforia. Pose jitter exists with almost all 2D planar targets. Other than the inconsistency in lighting conditions, it can also be caused due to the use of Incorrect target sizes, as they have the potential to negatively affect the detection or tracking algorithms that Vuforia operates upon. Due to the unstable behaviour this approach was not considered for further experiments.

3.2.4 Vuforia Marker Detection

This approach follows the same principles as Vuforia Image Detection, the only difference being the use of fiducial markers instead of photographs as image targets for

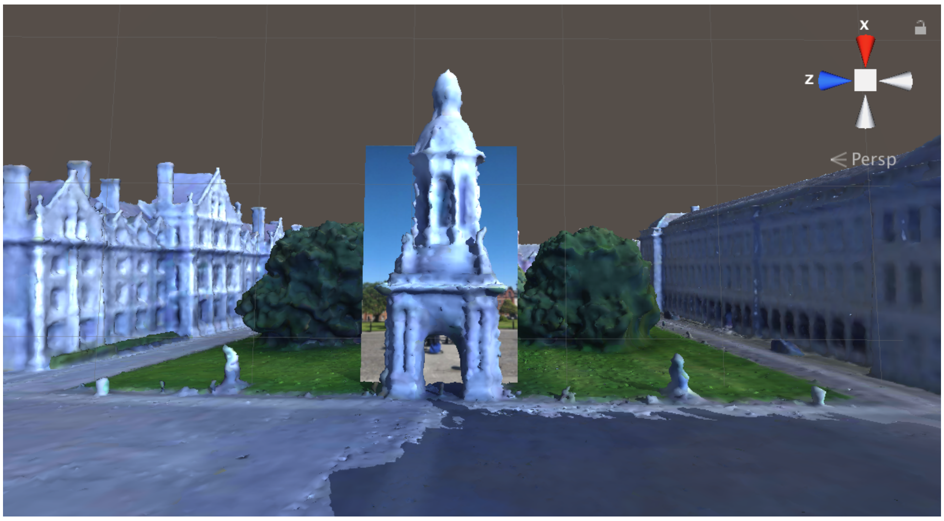

Figure 3.4: An Image of the Campanile used as an Image Target inside Unity

generating the AR experiences. In our case the fiducial marker as seen in Figure 3.6 was printed on an A4 size paper sheet in a specific black and white format, with high contrast and no repetitive patterns, all the characteristics of a good image target as recommended by Vuforia [73]. While this technique addresses the limitations of using a picture of the Campanile as the image target, it has a different set of constraints. One example of such a constraint is that when the user spawns the model by tracking the image target, the marker has to be in the AR camera's field of view (FOV) in order to keep the model stable and visible on the screen. As soon as the marker goes out of the camera's FOV (generally due to user movement), the tracking is lost and the model disappears. One feasible solution to this is a feature built into Vuforia called Extended Tracking [74] which allows the target pose information to be available even when the target is no longer in the view. But this approach is not robust and it exhibited instability and inconsistent results when used to spawn the model using this technique.

3.2.5 Vuforia Object Scanner

Vuforia provides developers with an Android application that enables them to scan physical 3D objects. As a result, a file called Object Data (*.OD) containing the source data is generated that defines object target in target manager image. Prior to

Figure 3.5: Model attached to the marker Image Target in Unity

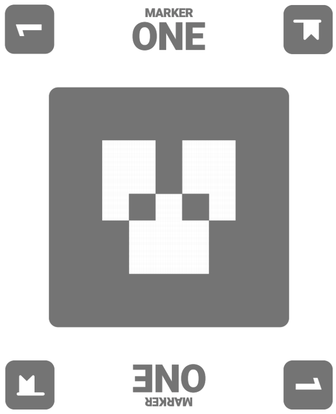

Figure 3.6: Marker used as the Image Target

scanning objects the developer is required to print out a specification called Object Scanning Target (OST) Image. It is required to establish the position of the object target relative to its local origin. The object to be scanned is placed inside the bounding box as shown in figure Image. The triangular pattern on the OST is called the feature region and it serves two main purposes - It assists the scanner in accurately recognising the pose of the object to be scanned in the grid region, and it helps define the culling region of the scanning space [75]. As evident with the description of this technique, the object to be scanned needs to be small in size, as the size of OST required in all cases is almost four times the size of the object to be scanned. And given the large scale of physical structures that need to be scanned, this approach becomes unfeasible.

3.2.6 Vuforia Model Target Generator

Model target generator (MTG) is a promising approach when working with 3D models, as it doesn't require the developer to scan the object in order to generate a model target. It is a standalone windows compatible package, which converts an existing 3D model into a Vuforia Engine Database, which can further be used for model target tracking [76]. The MTG also presents the developers with a guide view or a detection position which allows the users to align with the object to be augmented, in order to generate the 3D augmentation precisely overlaid on top of the object. After installing the MTG

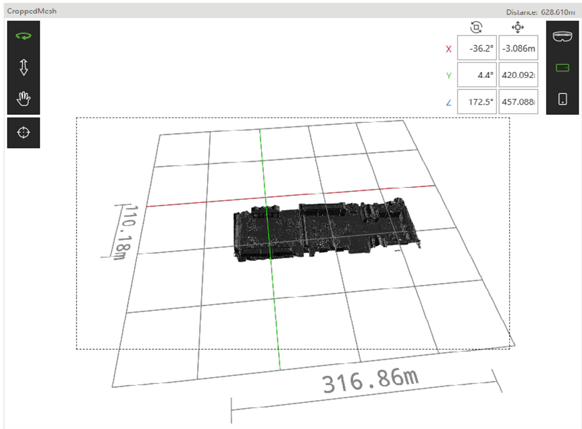

Figure 3.7: Front Square model imported in Vuforia Model Target Generator app

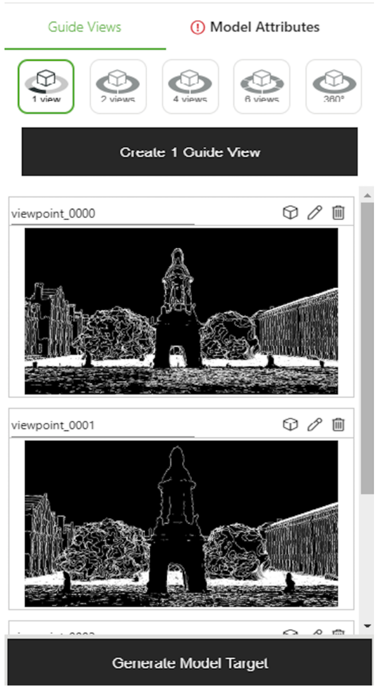

Figure 3.8: Guide views generated by Model Target Generator app

application in their system, the user needs to provide the application with any of the following compatible formats - Creo View (.pvz), Collada (.dae), FBX (.fbx), IGES (.igs, .iges), Wavefront (.obj), STEP (.stp, .step), STL (.stl, .sla), VRML (.wrl, .vrml), glTF 2.0. The official Vuforia website reports best results with the following formats - Creo View Adapter, Collada, FBX, and JT [76].

The properties of physical objects that are supported by MTG are as follows -

- Static objects with fixed spatial position
- Colored or patterned surface with high contrast
- A complex model with sufficient geometric details
- Rigid and non flexible model
- Model matching the real object exactly in shape and scale

For this approach to work, Vuforia recommends some best practises to choose the CAD model. The properties of an ideal CAD model are as follows -

- A maximum of 400,000 polygons or faces
- Comprise of a maximum of 10 parts
- Consists of a maximum of 5 textures
- Use a right hand coordinate system

However, the CAD model obtained after visual culling and then further decimation still had 484,063 faces, reduced by the aforementioned processes from 806,774, which was considerably higher than the recommended number. Further, the obtained model lacked high contrast, did not have the exact shape and scale of the real object and did not contain sufficient details. When the given model was tested with this approach, the device was not able to recognise the object as the target was derived from a model lacking sufficient detail. This approach was not pursued further due to the lack of robustness.

3.2.7 Vuforia Smart Terrain

Smart terrain is a feature provided by Vuforia that helps the user reconstruct and augment their physical environment, to create a more immersive AR experience. According to the official Vuforia documentation, smart terrain can be helpful in creating new kinds of gaming and visualisation applications [77]. But smart terrain is recommended only for stable and well-lit indoor environments and AR events to the scale as big as near-range table top experiences. Notwithstanding, this technique was tested with the front square and due to the enormous scale of the campus it was not possible to scan the terrain and develop an AR experience on top of it.

3.2.8 Vuforia Multi-Targets with Mesh Slicing

This method is an extension of the method in section 3.4.3, where the developer is required to slice the mesh of Front Square, isolated pieces of the model representing different parts of the Front Square of the TCD campus. The idea proposed here uses a separate image target for every demarcated piece of the mesh, and that solves the problem of the model disappearing when image target is out of the FOV of the camera, as there is at least one target in the FOV of the camera at any given time. The limitation of this approach is that the alignment of the image targets together in order to create immersion of the entire Front Square would require extreme precision, and is susceptible to human error. Similarly, this approach can be employed using the markers instead of image targets, as they are more robust than image targets. Although, using the markers with this technique would require precision twice as good as the previous approach, because here the developer will have the align the markers together as well, in addition to the isolated pieces of the mesh, rendering this technique unappealing. This shortcoming makes this approach less feasible and less robust.

3.2.9 Vuforia Marker Detection with Gyroscope Control

This technique is an extension of the method in section 3.4.4, and it attempts to solve the limitations of the aforementioned approach. Once the marker is outside the FOV of the AR camera, a custom script takes control over the gyroscope and a second camera replaces the AR camera inside the 3D model. This process occurs in such a short span of time that the user doesn't notice the shift in AR camera, and as the new camera replaces the last valid position of the original AR camera, there is no obstruction in immersion. However, this method has a major limitation. Due to the limited gyroscope control the user cannot walk around the inside the 3D model as the displacement data is not stored. This leads to no change in position inside the 3D model even if the user walks in different directions in real life, thus rendering an experience that doesn't feel real to the user.

3.2.10 Anchoring

A superior solution, called anchoring, demonstrated promising outputs given the constraints and objectives of this research. Anchoring falls under the category of Markerless AR, and it requires analysis of the surrounding planes to generate an environment map. Anchor points are points in the environment that the device knows should always keep a static position with respect to the AR camera. Anchor points are required because the current motion tracking in AR is not flawless. As the user walks around, the error in the position of the augmentation, also known as the drift, accumulates and the pose of the device might not be able to reflect the correct position of the user in 3D space. They contribute in correcting the drift by keeping track of significant points in 3D space. Vuforia provides a solution called Ground Plane detection that can be used to detect planes and place anchors in 3D space, but it detects planes based simply on the device orientation. That is, the ground does not get detected dynamically, rather there exists a predefined plane that reacts to the gyroscope and camera angle once the model is placed onto it [78]. This helps in creating an illusion of depth and placement on the ground. Google provides a more superior solution in the form of ARCore anchors as it detects ground plane with the help of feature points cloud. This approach is relatively stable and robust unless the user alters the orientation of the device. Every time the user refocusses the ground plane or reorientates the device, the augmentation drifts by a little extent, cumulating the marginal changes in position over a period of time causing drift error. Since the ability to shift the device orientation from portrait to landscape or vice versa is not as essential as the overall robustness of the solution, the tool of choice here for implementing Markerless AR is ARCore as it is much more stable and robust compared to Vuforia's ground plane detection.

Chapter 4

Implementation

4.1 Framework Pipeline

The pipeline of this application consists of 5 main sections. The first part is composed of acquiring a powerful mobile device and pre-processing the dataset to slice out the portion representing the Front Square out of the acquired model of TCD campus, using a 3D mesh processing software system called Meshlab [79]. The second section involves decimating, that is, reducing the number of faces in the 3D models of the Front Square of the campus and the crowd used in crowd simulation using Blender. The third section comprises of detailed explanation of the experiments carried out to discover the most efficient and robust method to spawn the 3D model in real world environment using AR. The fourth section describes the development of an AR portal door which allows users to walk inside in order to experience dynamic 3D models in real-time. Finally, the fifth section illustrates how the crowd simulation was implemented in the application.

4.1.1 Hardware and Software Specifications

Majority of this dissertation work, including the evaluation was carried out on a personal computer with the following CPU: 3.1 GHz Intel Core i5-7267U and GPU: Intel Iris Plus Graphics 650 (1536MB). Consequently, the performance evaluation presented in chapter 5 are conducted on a moderately powerful system and would presumably be substantially higher on a more recent and powerful machine. The code was implemented on Mac Mojave Operating System (Version 10.14.4). Visual Studio version

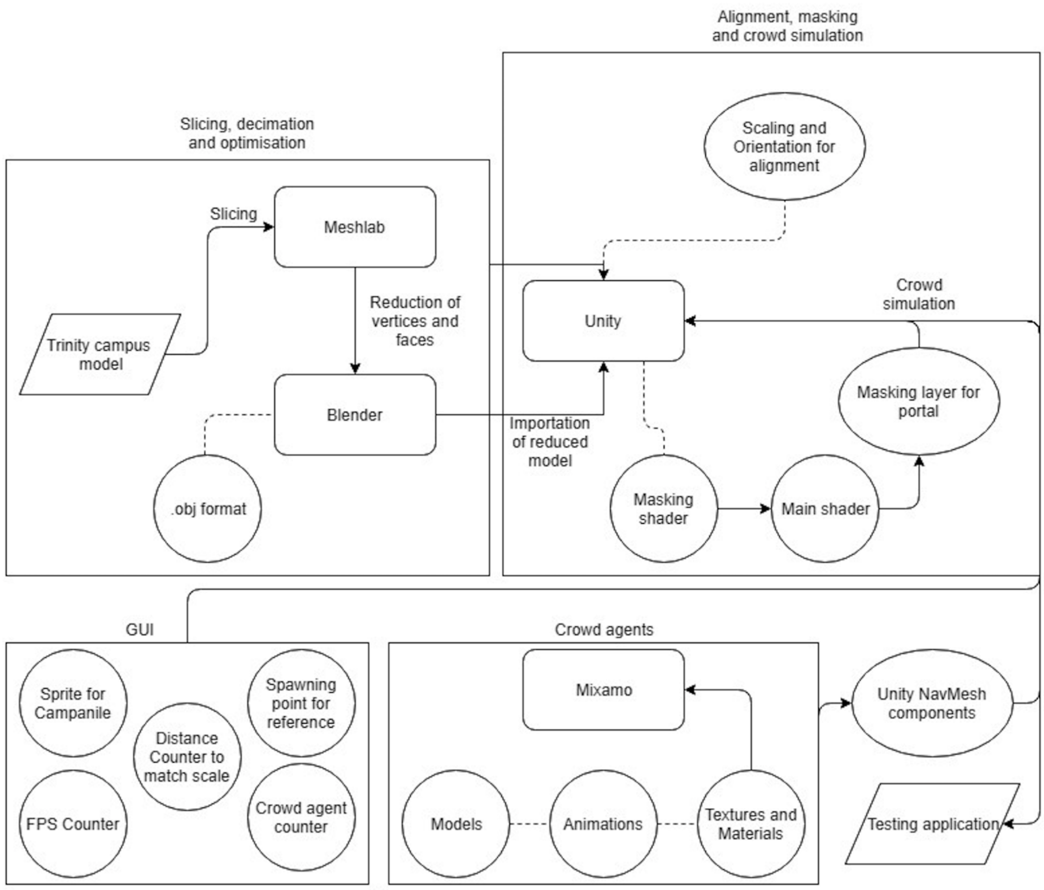

Figure 4.1: Framework Pipeline

7.6.9 was used as the platform for scripting, along with Unity version 2018.3.7 for the development of the AR application. This specific version of Unity was chosen because it was the latest version of Unity available at the beginning of the project, and all versions of unity post 2017.2 come integrated Vuforia Engine by default. Further, Higher level API for NavMeshComponents is available for this version of Unity, which was used in implementing crowd simulation using NavMesh. For experimentation purposes, Vuforia version 8.1 was used as it was the latest release during the experimentation period. Finally, ARCore version 1.2 was used in view of the fact that it was the first version of ARCore to support cloud anchors API and vertical plane detection.

4.1.2 Pre-Processing

The dataset discussed in Section 3.1 consists of a three dimensional representation of the whole TCD campus, instead of just the front square. Further, the model is very large in size, and given the limited computing power of the available hardware, that is, both the computer used for processing the model and the mobile phone used for running the application, it became imperative to reduce the number of polygons in the model and optimise it to maximum extent without sacrificing the overall quality in order to maintain the illusion of the TCD campus. In the world of computer graphics, these are crucial operations that are applied to 3D models to enhance the overall performance of the systems. Thus it is important to define the pertinent terminology in order to proceed with the report, decimation and back-face culling being such two terms. Decimation is the term used to describe downsampling in a 3D mesh. The number of faces in a model are reduced in the process, which reduces the size of the model, and hence the amount of computational power required to render it. On the other hand, back-face culling is a step in the graphical pipeline which determines whether a polygon of a graphical object is when projected onto the screen. It ensures that only the parts of the model that are visible on the screen get rendered and the back faces of the models are eliminated. In this research, visual culling was carried out on the original model manually, and the resulting mesh was then decimated with a ratio (original number of faces/ new number of faces) of 0.8, 0.6, 0.4 and 0.2 respectively. The comparison of visual quality and the number of faces are discussed in the results later in this report. On comparing the decimated meshes and the original mesh, it was decided that the mesh with a ratio of 0.6 was a good balance of visual quality and model complexity. Hence, all further development entailed the use of aforementioned mesh.

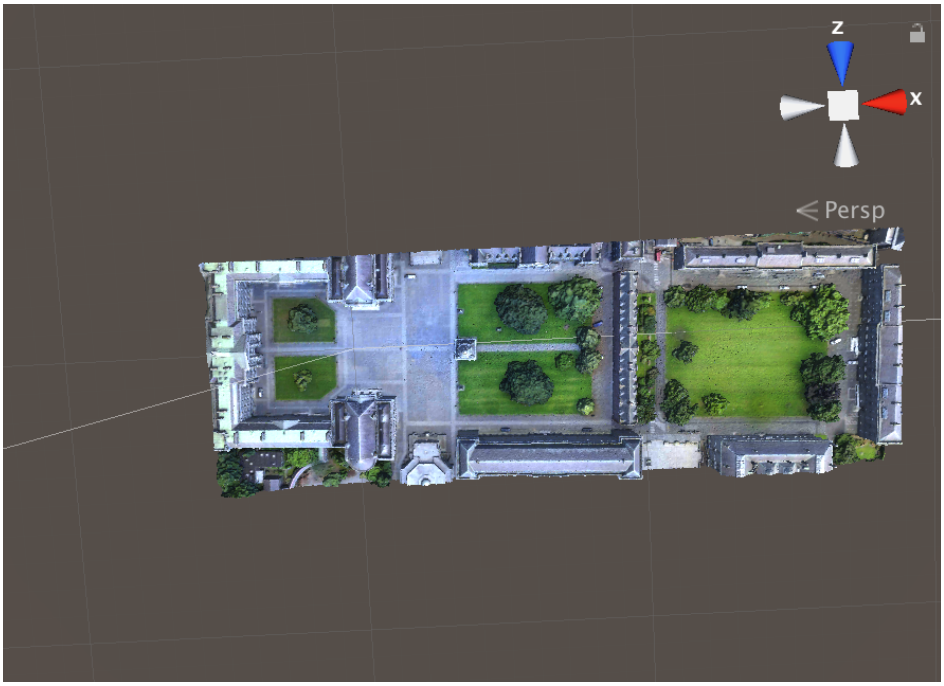

Figure 4.2: Front Square model without visual culling

Figure 4.3: Front Square model with visual culling

In order to move ahead with the project the provided data was pre-processed using Meshlab and Blender [80]. Meshlab was used to slice out the parts of the 3D model that do not represent the front square, and Blender was used for decimating the resulting model, that is, to reduce the number of faces and polygons in the model. Before that, the model of the TCD campus had to be cropped to only retain the information of the front square in the mesh, as the computing power available was not enough to process a 3D model of the whole campus, and it was out of the scope of this dissertation. Although cropping the model in the shape of a regular rectangle reduced the number of faces from 4,967,748 in the original mesh to 800,722 in the cropped mesh(Figure 4.2)(providing 83.88% reduction), it caused a tremendous loss of data as half of the buildings were sliced from the middle thus making the model look incomplete. In such a situation, anyone experiencing this model in full scale would find the illusion of the augmented campus unconvincing.

So finally in order to make the immersion of the front square in AR appear realistic, parts of the model were sliced in such a way that only the faces of the buildings visible to someone standing in the front square were included in the decimated model as shown in Figure 4.3, while eliminating the parts not visible. This visual culling on the model allows maximum level of detail and minimum performance impediment. Additionally, it is interesting to note that this approach reduces the number of faces in the model from 4,967,748 to 806,774 (83.76% reduction), which is almost exactly the same as the previous approach, but with no sustantial loss of information.

4.1.3 Anchoring based Solution with ARCore

In order to start the development of the AR application, a Unity project was created as instructed on official Google Quickstart guide [81], and the ARCore SDK was imported. Google provides a sample scene called HelloAR with the ARCore SDK, which would be expanded upon as explained in the following sections in order to achieve the desired results. Following this step, the project was configured as indicated by the quickstart guide. Building and running the default project on a compatible Android device enables the users to spawn 3D models of Andy the Android wherever they touch the screen on a valid detected surface, using hit testing. The HelloAR sample scene uses anchoring as explained in section 3.2.10, to place the Andy prefab in 3D space and then maintain its location throughout the session. To be able to spawn the 3D model of the front square instead of Andy the Android Image, the decimated and visually culled mesh of the Front Square was imported inside Unity, converted to a prefab and attached to the public GameObject of Example Controller. On building and running the application we were able to spawn the desired mesh on any valid detected surface just like the Andy Android prefab. Consequently, the first working application establishing the proof of concept was realised.

4.1.4 AR Portal Door

The implementation of the AR Portal Door involved 5 major steps. After implementing the first minimum working prototype as mentioned above, the portal development work was initiated. To create the frame of the door, 4 different cubes were created inside one empty GameObject and arranged in order to depict the look of a real door frame as shown in fig. Image. The frame was then put on the ground with the help of a plane GameObject that was located at position (0,0,0). Surrounding the door, walls were created in the form of cube meshes, to be large enough to enclose full-scale (approximating 900 metres in length and 300 metres in height) model in inspection. The walls were arranged to form an enormously large cube which could be accessed only through the portal door generated in the previous step. Once the cube and the portal door were generated and aligned together, the next step was to write custom shaders enabling the portal effect, to make the whole cubical structure invisible to the viewer unless the access is attempted through the designated AR portal door. In order to enable the portal effect, two custom shaders were written, one called MainShader and the other called MaskShader. MainShader was added to the door GameObject, along with a material called DoorMaterial to create an illusion of transparency.

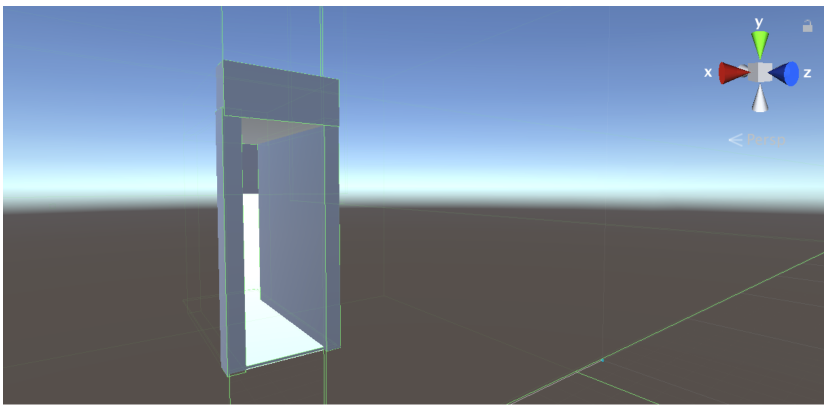

Figure 4.4: AR Door Front view

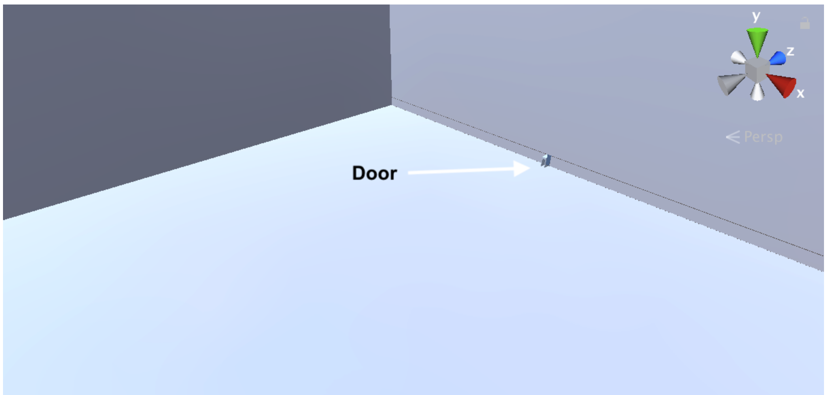

Figure 4.5: AR Coor view from inside the cube

MainShader is the vertex shader here which transforms the attributes of vertices, specifically color in this case, from the original color area to display area. MaskShader is the fragment shader here which would take in the output of MainShader and output the functionality to the main function. Shader MainShader is responsible for applying black and white mask onto the cube prefab by taking the texture component and assigning the white color to it, nullifying the Bumpmap and forming a stencil. It assigns the albedo, normalmap, metallic and smoothness of the color respectively. This information is further passed onto the fragment shader, wherein, the input is the white mask. It takes this stencil's white mask and returns it as (0,0,0,0) wherein R = 0, G=0, B=0, and Alpha = 0, which makes the surface completely transparent, masking the 4 walls of the cube as a result while making sure that the portal is still visible. This data is then passed onto the main function for it to render. This enabled the portal door to display all the contents inside the door while blocking out all the content that is outside the boundary of the door shown above.

4.1.5 Crowd Simulation

The crowd simulation was implemented using Unity's built in NavMesh components. A NavMesh stands for a navigation mesh which defines walkable surfaces for agents in the scene in Unity. The process of assigning the walkable surface for agents is called baking the NavMesh. To bake a new NavMesh, a new GameObject was created and a pre-defined component called NavMeshSurface was attached to the GameObject. It provides the developer with the option to bake the NavMesh with the default settings or alter the settings to suit the user requirements. Once the characteristics of the agent were defined in the settings, a blue surface on top of the 3D model, that is, the NavMesh was baked, assigning the walkable and non walkable areas on the model. The chosen agent had a radius of 0.5 metres, height of 1.8 metres, step height of 0.4 metres and 37 degrees of maximum inclination. These specific merits were chosen to facilitate the agent to navigate the whole mesh on the walkable areas as shown in Figure 4.7, as the irregularities of the mesh surface magnified when the mesh was scaled up to a real life magnitude, making it difficult to navigate for an agent with conventional navigating abilities and physical properties. This was an easily rectifiable problem by just altering the agent settings in Unity(especially maximum slope), but is something the reader should be aware of. Three different models dressed in vintage clothing were employed as the crowd and each model had a different spawning point, with one destroyer point and one point in their path each to add variation of movement. The variety in spawner and destroyer points contributed in creating the illusion of a busy crowd walking on the TCD Front Square mesh.

Figure 4.6: simulated crowd inside TCD campus Front-Square mesh

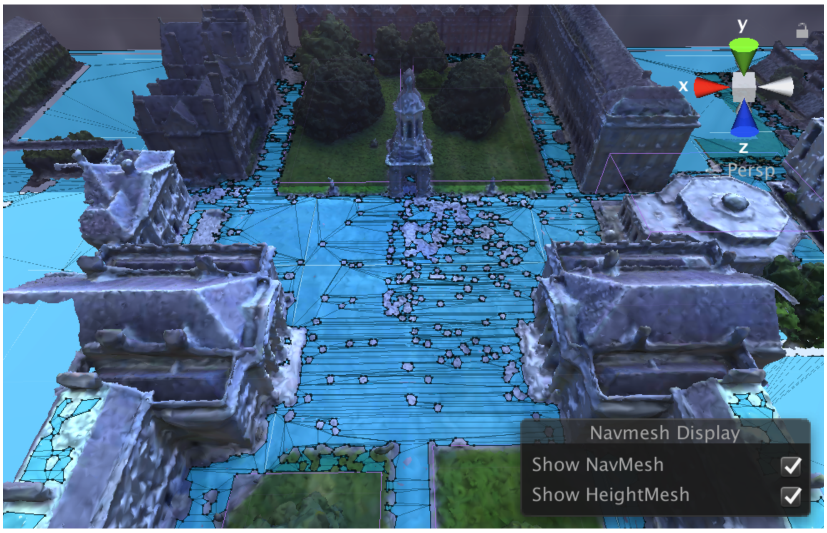

Figure 4.7: NavMesh baked with default agent settings had surface irregularities

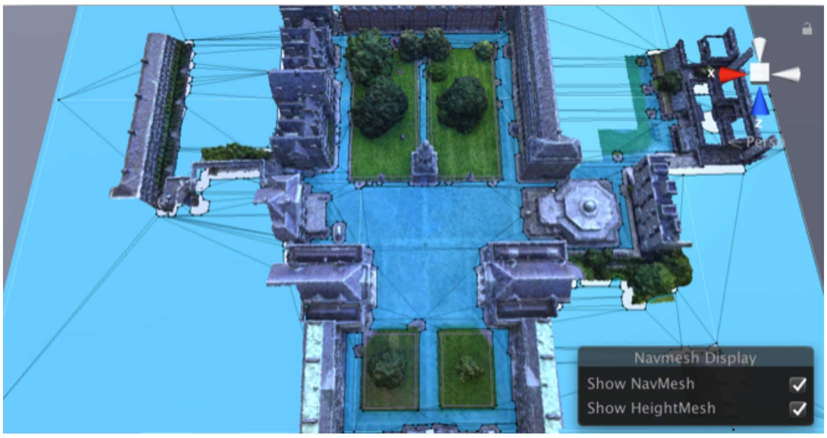

Figure 4.8: NavMesh baked after altering the agent settings

4.1.6 Application Operation Flow

Please refer to Appendix D Figure 20

4.1.7 Application Walkthrough

This section accords a simple walkthrough of the Android application that has been developed to demonstrate the proposed solution to the research question. The solution is platform independent in a sense that is it developed in Unity Engine which offers an additional preview package called ARFoundation [82] which allows multi-platform handheld AR development; so that the developers can build and run a project as both an Android and IOS application just by customising the configuration of the project in build settings. This presents the developer with the flexibility to extend the solution to different operating systems as desired without the need to port the code manually.

Once the application is launched ARCore detects the surface planes in the scene and commences mapping the environment with the help of feature points. With every frame ARCore gets queried by the system for new or updated planes and it visualises them correspondingly, using different arbitrary colors for different planes. The user is required to align the orientation of Campanile in the sprite image as mentioned in the last section, along with the real Campanile and then tap the screen on the directed point to launch the AR experience. As a result, an AR portal door is generated as depicted in Figure 4.4, which enables the users to walk inside and experience the 3D model of the old TCD Front Square, overlaid on top of the real institution in full scale. Due to the correct scaling and manual refinements made to the 3D model, subject to the user tapping the correct point on the screen as instructed, the eventual output is a model that aligns perfectly with the real structures. Furthermore, crowd simulation with agents dressed up in vintage clothes provide a convincing illusion of the historical campus of TCD.

Chapter 5

Results and Discussion

This section expands on the results that were observed with this research, and then explicate how further processing on the dataset helped optimise the model and improve performance. The mobile hardware was stress tested to test the performance of the application by populating the scene with immensely large number of agents and the results were recorded. Further, this section discusses the other metrics that were defined to verify the accuracy of the overlay of the model.

5.1 Performance

Field tests were conducted by employing three devices:1) Samsung Galaxy Note 9 (Android Oreo 8.1, RAM 6GB, 2.7 GHz octa-core Qualcomm Snapdragon 845 CPU, Mali-G72 MP18 GPU, accelerometer, gyroscope), 2) Xiaomi Pocophone F1 (Android Oreo 8.1, RAM 6GB, 2.8 GHz octa-core Qualcomm Snapdragon 845 CPU, Qualcomm Adreno 630 GPU, accelerometer, gyroscope) and 3) Oneplus 5 (Android Pie 9.0, RAM 6GB, 2.3 GHz octa-core Qualcomm Snapdragon 835 CPU, Andreno 540 GPU, accelerometer, gyroscope). All three phones are part of ARCore's officially supported devices list, and are similarly specced, which makes the results comparable and easy to validate. The objective of these tests was to first investigate the accuracy of the overlay and the degree of displacement or alignment error using the metrics defined hereby. To establish the accuracy of the tests, parameters of First Person Camera in Unity were adjusted to match the physical properties of the respective devices as shown in Appendix D, and their respective applications were built matching the camera configuration as shown in Figures 17, 19 & 18. This process ensured that all the devices were receiving the same camera feed and thus produced results which could be compared.

To ensure that the virtual representation of the campus is perfectly masked on top of the real buildings, multiple considerations could be addressed, and the number of factors considered is directly proportional to the precision in the final output. However, the following two facets are critical to establish believable results - scale and orientation. The subsections below explain the approaches followed in detail.

5.1.1 Scale Matching

The acquired dataset of the Front Square model was in full scale, that is, the size of structures inside the 3D mesh were comparable to the real buildings of the TCD campus. Nevertheless, the mesh required manual adjustments in scale. The metric used to measure the accuracy of scale between the model and the Front Square was the distance between the Campanile and the Arc of the entrance door to the Front Square of TCD campus. An Android application called AeroGMS Measure Land (Version 1.9.7) (Refer Appendix D) was used to measure the distance between the aforementioned points. It is a robust utility that uses combination of custom built Geographic Information System (GIS) and Management Information System (MIS) to measure the distance between any two points on the map and provides exceptionally accurate results.

Further, the application was programmed to use the accelerometer of the respective device to display the displacement along z-axis, thus updating the distance travelled in a straight line in real-time. The argument validating this metric is that for a proportionally error-free model, if the distance between two points on the model is identical to the distance in real life, then the model should be the same scale as the real life structures. Figure 5.1 depicts a graph that indicates the comparison of the distance between the two points measured by the three devices as compared to the real distance. The device that achieved the results closest to the real distance would be considered the most accurate scale wise. As evident by the graph, the device that demonstrated maximum accuracy was Samsung Galaxy Note 9, and a probable explanation for this performance could be the latest hardware and the most advanced sensors in the group of test devices.

Table 5.1: Table of distance accuracy of the 3 devices as compared to the real distance

Device	Distance Measured(Metres)	Percentage Error
Original Distance	90.40	NA
Oneplus 5	93.19	3.09%
Samsung Galaxy Note 9	90.16	-0.27%
Xiaomi Pocophone F1	92.72	2.57%

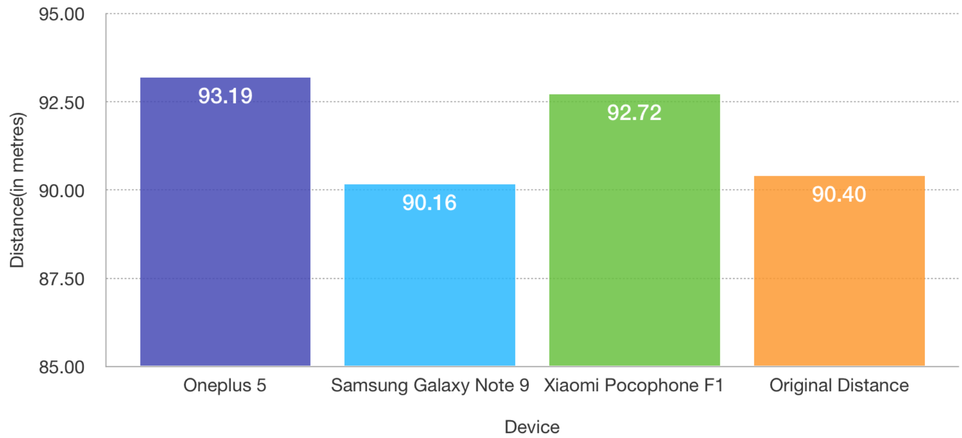

Figure 5.1: Graph of distance accuracy of 3 devices compared to real distance

5.1.2 Orientation Validation

The metric used to validate the orientation of the model with respect to the real structure was also linked to the iconic Campanile. The proposed method includes a sprite image of the Campanile clicked from a specific point under the Arc of entrance door to the Front Square, which acts as a guide view for the user to align the screen to before tapping on the detected surface to spawn the AR Portal. To ensure optimum alignment, various different locations and orientations were manually tested and ultimately this spot was discovered. Also, the point on the screen that needed to be tapped to spawn the model with perfect alignment was noted, which was later replaced by a green call to action button, as shown in Figure 5.2. The device pose and location were manually specified such that when the user aligned the guide view Image with the Campanile from the designated point and pressed the green button on the screen, it spawned the model in AR with commendable alignment. Although the location and orientation of the device before spawning the model are considered, there are various other factors that need to be taken into account to improve the accuracy of the final output. One such factor could be the height of the device before the guide view is aligned to spawn the AR Door. In this case, the height of the developer is 1.75 metres, and the mobile device was constantly kept at the developer's shoulder height of 1.5 metres throughout the assessment of results.

Figure 5.2: Guide View for orientation validation

After completing the evaluation of the primary objective of the dissertation, additional analyses and evaluation of the results of the secondary objectives was carried out. The dataset discussed in section 3.1 was pre-processed as mentioned in section 4.1.2

to optimise the performance of the application. Four different variations of the model were tested possessing unique visual characteristics and the results were recorded as demonstrated in table. Appendix B exhibits the stark difference in the visual fidelity of models, and as evident by the output, the model with a ratio of 0.6 appeared to be the optimum balance of visual fidelity and efficiency. Hence it was designated as the model utilised in the development of the AR Door application.

Table 5.2: Comparison of properties of mesh for various levels of decimation

Ratio	Vertices	Faces	Results
1(Original)	402218	806774	Sharp edges and good detail
0.8	321527	645419	Model gets slightly rounded, loses its sharp edges due to reduction of Triangles
0.6	240787	484063	Edges become more organic, further reduction in the appeal due to rounding
0.4	159936	322709	The model looks very blobby, most of the sharp edges are now rounded
0.2	78755	161353	Details in the model greatly distorted with almost no resemblance of the original shapes

Further, the performance of the crowd simulation was evaluated, and the outputs from each device were compared. To achieve this, a total of 6 Android Package Kit (APK) files were exported, two for each device due to custom first person camera settings tailored to the respective hardware. Out of the two APK files exported for each device, one contained crowd simulation with a destroyer point in it and the second featured crowd simulation without a destroyer point, which meant the crowd was allowed to accumulate for a certain period of time in each device, and the resultant frame-rate was plotted on a 2-axis graph as shown in Figures 5.6, 5.7 and 5.8. The application was programmed to spawn an agent at each of the 3 spawn points in the scene every 4 seconds. The aforementioned figures plot the frame-rate against number of agents for a scene without crowd destroyer points as the application is run for 30 minutes for each case. On the other hand, Figures 5.3, 5.4 and 5.5 represent the plot of frame-rate against number of agents for a scene with the crowd destroyer points present for each agent.

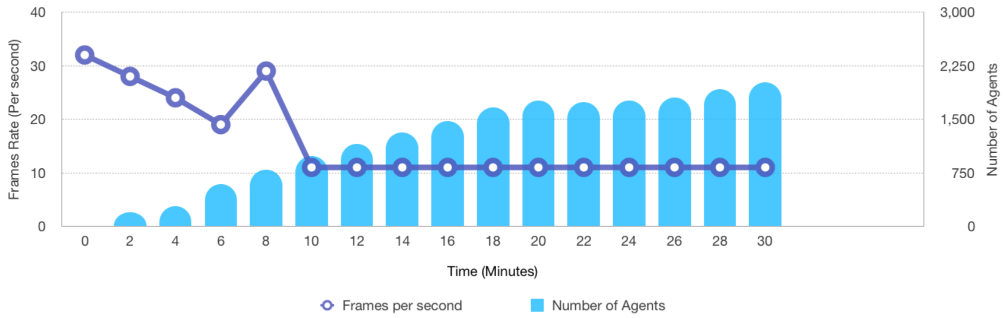

Figure 5.3: Oneplus 5 two axis chart with destroyer points

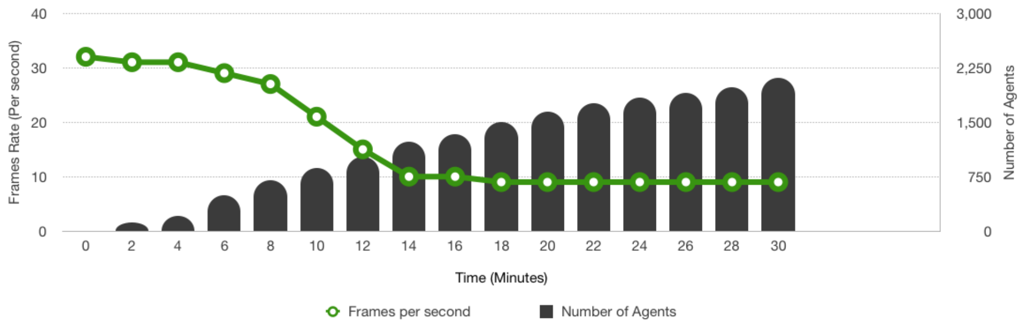

Figure 5.4: Pocophone F1 two axis chart with destroyer points

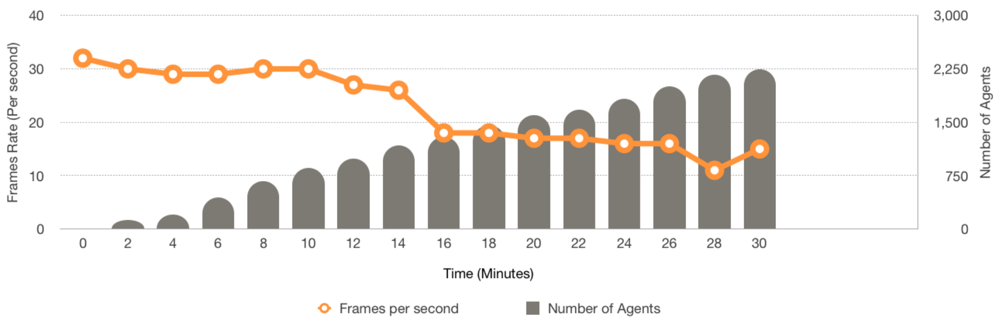

Figure 5.5: Galaxy S9 two axis chart with destroyer points

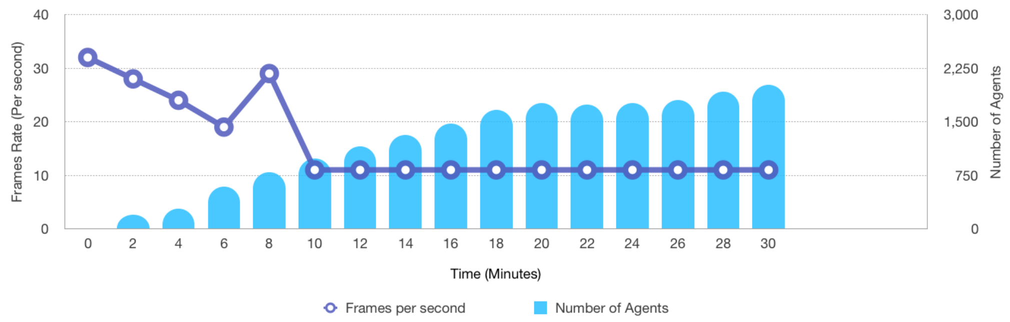

Figure 5.6: Oneplus 5 two axis chart without destroyer points

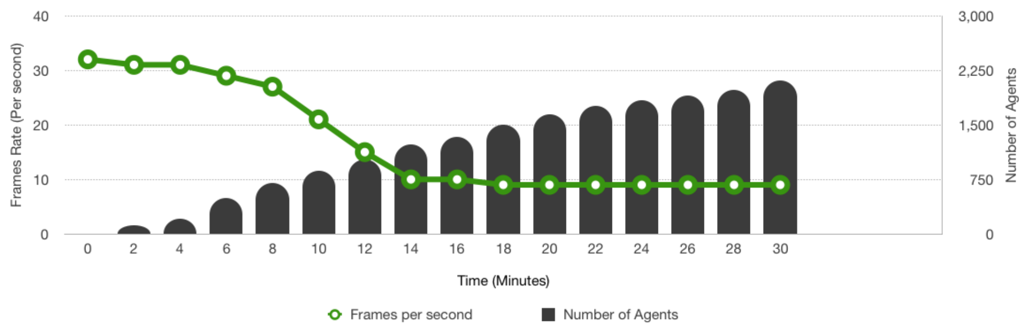

Figure 5.7: Pocophone F1 two axis chart without destroyer points

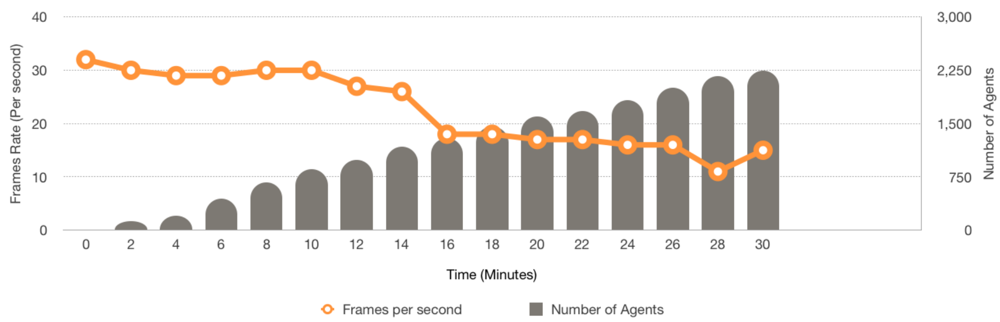

Figure 5.8: Galaxy S9 two axis chart without destroyer points

The trends observed in the 6 plots do not yield exactly the expected results. In Figure 5.3, the Oneplus 5 demonstrated a steady increase in the number of agents when the scene contained a destroyer point for each member of the crowd. however, after the

10 minute mark, the rate at which the crowd was spawning and the rate at which the agents were reaching their respective destroyer points balanced out, exhibiting a steady frame-rate of 11 fps after 10 minutes. The Pocophone (Figure 5.4) however, performed better than the Oneplus 5, and steady kept on spawning the agents with the frame-rate dropping steadily until it evened out like the Oneplus 5, however taking almost twice the time than the Oneplus 5 to reach a constant low frame rate. A probable reason for such a behaviour could be the fact that the Pocophone has slightly more powerful and more recent (Snapdragon 835 CPU in the OP5 compared to Snapdragon 845 in the F1) processor, which gave it an edge during these tests. The best performer, however, was the Samsung Galaxy Note 9 (Figure 5.5), being the most powerful and most recent device out of the ones tested. The frame-rate decreased from around 32fps at the beginning of the test to around 16fps towards the end of the designated 30 minutes, while not achieving a constantly low frame rate. Given the powerful hardware of the device, it is assumed that it would requrie a bit more time compared to the other two devices to bottleneck in a test like this. There also a slight spike in frame-rate observed towards the end of the test in Note 9, but that could be discarded as an outlier as there is no probable explanation for a similar spike in performance when the resource requirement is growing linearly with time.

When the destroyer points were removed from the scene, the agents started populating the scene at a much lower rate than earlier in all devices, even though the frame-rates were similar to the previous comparisons. One plausible explanation for this unexpected behaviour could be the use of A* pathfinding algorithm used by the agents to go from one point to the other, combined with the fact that as the agents were not getting destroyed, they were queuing up in a line and thus blocking the path of new agents that were spawned and thus slowing down the rate of crowd generation in the process. A* algorithm tries to find to least cost shortest path from one point to another, but it fails when it cannot be proved that the least cost choice yields the optimal solution. When a large number of agents populated the scene, gathering around the spawn point, the least cost path would cause conflict of interest with other agents causing the algorithm to fail. The removal of destroyer points from the scene also demanded very high computational power from the devices even though the frame rates were not dropping significantly, and this could be linked to the previous point. At one instance, the test had to be cancelled due to safety concerns after 28 minutes in the case of Oneplus 5 due to extreme overheating. However, the aforementioned hypothesis can be tested as a part of the future work. In case of absense of destroyer points, Pocophone F1 (Figure 5.7) demonstrated slight decrease in frame-rate after 14 minutes in the test, while the Note 9 (Figure 5.8) performed similar to the previous test and did not achieve a constant low frame rate, attributed to the short duration of the test on the relatively powerful hardware of Note 9.

5.2 Discussion and Limitations

Although the proposed method was robust in a majority of cases (Figure 5.9), the model still didn't exhibit superior alignment in some cases. One potential reason for that could be the hinderance generated in feature point and ground plane detection by the crowd of tourists on the campus during the tests. Being in the field of view of the camera, the tourists and their clothing also gets detected and the feature points on them are identified, but when they move it causes movement in detected feature points, destabilising feature point space as a result. ARCore generates anchors on the basis of the pose yielded by a hit test. It uses feature points or planes, also known as trackables, to anchor the virtual object to, as the trackables remain stable even if the user moves around. ARCore tracks the position of these surrounding objects over time, and tries to build an environmental understanding to be able to produce convincing augmentations.

Being in the FOV of the camera, the tourists get detected as feature points by ARCore, and when it anchors the mesh of the Front Square to the feature points, the movement in feature points causes ARCore to display abnormal behaviour. Furthermore, drift is a known issue with most of the AR tools available currently, and ARCore is no exception. Every time the user refocusses the ground plane or reorientates the device, the augmentation drifts by a little extent, cumulating the marginal changes in position over a period of time causing drift error. Conceivably the most noteworthy shortcoming highlighted in surveys [83] was limitations of the software or instability, and the imprecision of the hardware.

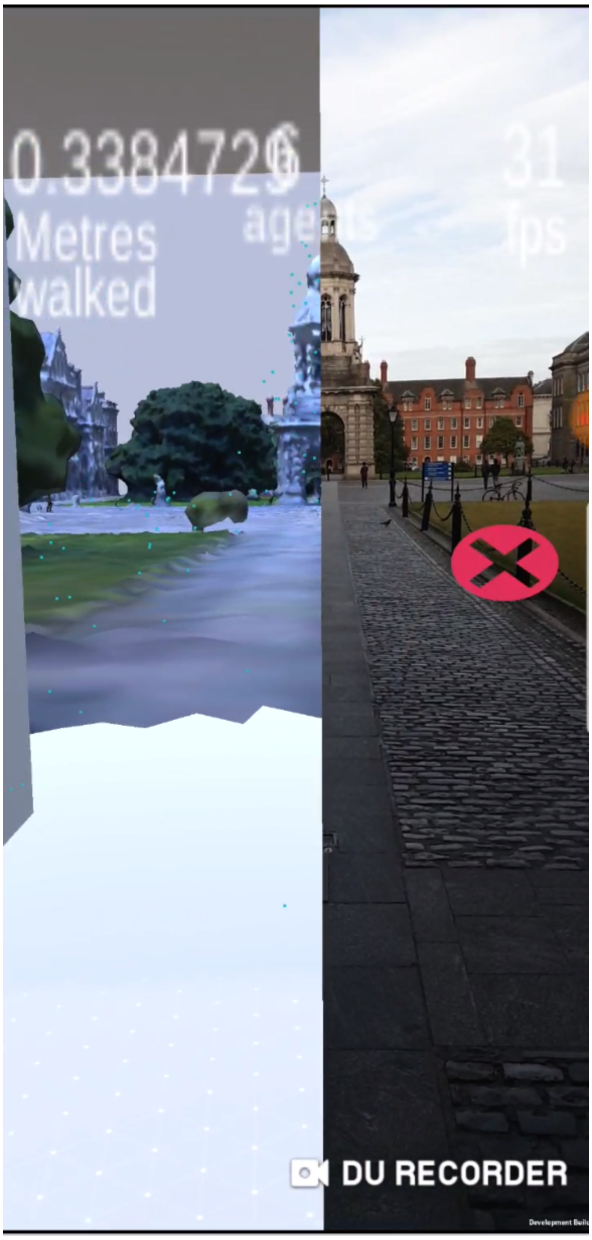

Figure 5.9: An example of misalignment of the model with the landmark

Chapter 6

Conclusion

In this dissertation, a consumer-grade mobile AR application for the general audience was designed with no specialised hardware and the eventual goal of amplifying the synergy between cultural heritage sites and the tourists. By utilising 3D meshes of the historical monuments through AR, gap between the real world and digital media was bridged. An extendable platform was developed, that can comfortably accommodate additional cultural heritage sites, demanding slight preparation work. The presented system will equip specialists in the future to demonstrate their creations using various other kinds of representation techniques. Notwithstanding the fact that handheld mobile AR puts forward a variety of localisation and registration challenges, it dispenses unique experiences to a broad range of audience.

Modern smartphones are technologically equipped to facilitate the development of robust AR experiences that complements the apprehension of ancient datasets. However, the computational power required to process such datasets is still not easy to obtain in those compact devices. AR technology has recently seen widespread adoption, and according to the trends it certainly will gain more popularity in the near future. This has motivated researchers to develop complicated registration and tracking algorithms that perform commendably on the insufficiently powered mobile devices available in the market currently.

The complications and limitations of the common registration and tracking algorithms are far from resolved. However, the availability of powerful and user friendly game engines such as Unity 3D and Unreal Engine, combined with the plethora of extensive resources accessible to public would certainly promote the development of advanced interactions and high-fidelity graphics in the coming years, leading to more widespread adoption of the technology.

While mobile outdoors AR for Cultural Heritage has been extensively researched in the past, most of the investigation that has been carried out uses specific complex and non-portable hardware, with no consideration of optimisation. The approach discussed here intents to expand the topical functionalities of mobile handheld AR by enhancing the scene comprehension around the device, hence compelling the researchers to focus on optimising the devices to perform with the limited computational power of this kind of mobile devices. Nevertheless, these outcomes can be used as a starting point for future work in this area, even if using alternate mediums for recreation. The application presented in this paper, in its current state, does not support social interactions or consistent crowd simulation between the users, but adding a communication layer to this application would help boost the tourists' interest in historical sites and will increase their enthusiasm to engage with similar applications on a deeper level.

6.1 Future Work

There's a multitude of directions that this body of work can be expanded into in future. The flexibility in all the modules enables addition of new features or 3D models to this application very easy. Currently, the application considers only the Front Square of TCD campus, which can be expanded by using the model of the whole TCD campus. While using a model of such an enormous scale would require tremendous computational power to render on handheld mobile devices, the techniques used in this research to decimate, cull and thus optimise the models to improve program efficiency can be employed and tested on larger models as well.

Future work might also address a more in-depth examination of the methodology followed to overlay the 3D models on top of real life structures in full scale. The current approach relies on human precision which makes it prone to errors in alignment. Even though the anchors in ARCore used to affix the model at a specific point in 3D space are robust if the environment around has enough feature points, the error in alignment is still a major possibility as it is introduced by human oversight.

Additionally, advanced crowd simulation can be implemented in the application, with more intelligent agents and random pathways instead of following predefined points on the NavMesh, to make the crowd look more natural. In addition, more variety and human interaction using agents that explain the users about the monument can be added to the crowd to make the application more engaging. As studies show, including audio prompts and information relevant to landmarks has proven to increase tourists' inclination to pay more at tourist spots [84].

Furthermore, the current system uses the entire model of the Front Square (including the floor) and spawns it using AR. A more immersive solution could be to just use the models of buildings and not the floor, allowing the intelligent agents of the simulated crowd to walk on transparent NavMesh. Another feature that can be added to the crowd is the ability to avoid collisions with the members of the real life crowd and adapt to the surroundings, which will shape the experience to be much more convincing and realistic to the users.

Bibliography

[1] "Pokémon GO!" [Online]. Available: https://www.pokemongo.com/en-us/

[2] "Microsoft HoloLens." [Online]. Available: https://www.microsoft.com/en-us/hololens

[3] "Augmented Reality Market by Offering & Application - Global Forecast 2023." [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/augmented-reality-market-82758548.html

[4] J. Bacca, S. Baldiris, R. Fabregat, S. Graf, and D. Kinshuk, "Augmented Reality Trends in Education: A Systematic Review of Research and Applications," *Educational Technology and Society*, vol. 17, pp. 133–149, Oct. 2014.

[5] H.-G. Ha and J. Hong, "Augmented Reality in Medicine," *Hanyang Medical Reviews*, vol. 36, no. 4, p. 242, 2016. [Online]. Available: https://synapse.koreamed.org/DOIx.php?id=10.7599/hmr.2016.36.4.242

[6] J. Novak-Marcincin, J. Barna, M. Janak, and L. Novakova-Marcincinova, "Augmented Reality Aided Manufacturing," *Procedia Computer Science*, vol. 25, pp. 23–31, Dec. 2013.

[7] X. Zhang, N. Navab, and S. Liou, "E-commerce direct marketing using augmented reality," in *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, vol. 1, Jul. 2000, pp. 88–91 vol.1.

[8] W. Piekarski and B. Thomas, "ARQuake: The Outdoor Augmented Reality Gaming System," *Commun. ACM*, vol. 45, pp. 36–38, Jan. 2002.

[9] M. C. Juan, M. Alcaniz, C. Monserrat, C. Botella, R. M. Banos, and B. Guerrero, “Using augmented reality to treat phobias,” *IEEE Computer Graphics and Applications*, vol. 25, no. 6, pp. 31–37, Nov. 2005.

[10] V. Lang and P. Sittler, “AUGMENTED REALITY FOR REAL ESTATE,” p. 14, 2012.

[11] P. Diegmann, M. Schmidt-Kraepelin, S. van den Eynden, and D. Basten, “Benefits of Augmented Reality in Educational Environments – A Systematic Literature Review,” Mar. 2015.

[12] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, “Augmented reality: a class of displays on the reality-virtuality continuum,” H. Das, Ed., Boston, MA, Dec. 1995, pp. 282–292. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=981543

[13] H.-K. Wu, S. W.-Y. Lee, H.-Y. Chang, and J.-C. Liang, “Current status, opportunities and challenges of augmented reality in education,” *Computers & Education*, vol. 62, pp. 41–49, Mar. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360131512002527

[14] E. Klopfer and K. Squire, “Environmental Detectives—the development of an augmented reality platform for environmental simulations,” *Educational Technology Research and Development*, vol. 56, pp. 203–228, Apr. 2007.

[15] R. T. Azuma, “A Survey of Augmented Reality,” p. 48.

[16] F. Zhou, H. B.-L. Duh, and M. Billinghurst, “Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR,” in *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ser. ISMAR ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 193–202. [Online]. Available: https://doi.org/10.1109/ISMAR.2008.4637362

[17] A. Edwards-Stewart, T. Hoyt, and G. M. Reger, “Classifying Different Types of Augmented Reality Technology,” p. 4.

[18] “HP Reveal.” [Online]. Available: https://www.hpreveal.com/

[19] “String.” [Online]. Available: https://string.co/

[20] Blippar, “Blippar - Augmented Reality (AR) & Computer Vision Company.” [Online]. Available: https://www.blippar.com/

[21] G. Reitmayr and D. Schmalstieg, “Location Based Applications for Mobile Augmented Reality,” in *Proceedings of the Fourth Australasian User Interface Conference on User Interfaces 2003 - Volume 18*, ser. AUIC ’03. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2003, pp. 65–73, event-place: Adelaide, Australia. [Online]. Available: http://dl.acm.org/citation.cfm?id=820086.820103

[22] “Instagram.” [Online]. Available: https://instagram.com

[23] “FaceCake Marketing Technologies, Inc.” [Online]. Available: https://www.facecake.com

[24] “Google Glass.” [Online]. Available: https://www.google.com/glass/start/

[25] “Google Just a Line.” [Online]. Available: https://justaline.withgoogle.com/

[26] “Virtualpop,” Dec. 2014. [Online]. Available: http://www.virtualpopgames.com/

[27] R. d. Koeijer, “Augmented reality for cultural heritage - Comparing a cardboard device with a mobile handheld device in a museum setting,” 2018. [Online]. Available: http://dspace.library.uu.nl/handle/1874/366191

[28] M. Ding, “Augmented Reality in Museums,” p. 13. [Online]. Available: https://amt-lab.org/s/Augmented-Reality-in-Museums.pdf

[29] JDeBonis, “ArtLens App,” Nov. 2012. [Online]. Available: http://www.clevelandart.org/artlens-gallery/artlens-app

[30] “Cleveland Museum of Art.” [Online]. Available: http://www.clevelandart.org/

[31] “Skin & Bones.” [Online]. Available: https://apps.apple.com/us/app/skin-bones/id929733243

[32] "Smithsonian Homepage | Smithsonian Institution." [Online]. Available: https://www.si.edu/

[33] "Bone Hall | Smithsonian National Museum of Natural History." [Online]. Available: https://naturalhistory.si.edu/exhibits/bone-hall

[34] "Home." [Online]. Available: https://blantonmuseum.org/

[35] "The University of Texas at Austin." [Online]. Available: https://www.utexas.edu

[36] K.-E. Chang, C.-T. Chang, H.-T. Hou, Y.-T. Sung, H.-L. Chao, and C.-M. Lee, "Development and behavioral pattern analysis of a mobile guide system with augmented reality for painting appreciation instruction in an art museum," *Computers & Education*, vol. 71, pp. 185–197, Feb. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360131513002868

[37] "Taipei Fine Arts Museum," Jul. 2019, page Version ID: 905758097. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Taipei_Fine_Arts_Museum&oldid=905758097

[38] U. C. Pendit, S. B. Zaibon, and J. A. A. Bakar, "User experience on enjoyable informal learning via mobile AR: Development and evaluation," 2014.

[39] "Malacca Travel Guide - Everything You Need To Know About Malacca." [Online]. Available: http://www.malacca.ws/

[40] "Digital Transformation Solutions to Unlock the Value of IIoT | PTC." [Online]. Available: https://www.ptc.com/en

[41] "2019 Augmented and Virtual Reality Survey Results." [Online]. Available: https://www.perkinscoie.com/en/ar-vr-survey-results/2019-augmented-and-virtual-reality-survey-results.html

[42] V. A. Prisacariu, O. Kähler, D. W. Murray, and I. D. Reid, "Real-Time 3d Tracking and Reconstruction on Mobile Phones," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 5, pp. 557–570, May 2015.

[43] V. Vlahakis, M. Ioannidis, J. Karigiannis, M. Tsotros, M. Gounaris, D. Stricker, T. Gleue, P. Daehne, and L. Almeida, "Archeoguide: an augmented reality guide for archaeological sites," *IEEE Computer Graphics and Applications*, vol. 22, no. 5, pp. 52–60, Sep. 2002.

[44] C. Panou, L. Ragia, D. Dimelli, and K. Mania, "An Architecture for Mobile Outdoors Augmented Reality for Cultural Heritage," *ISPRS International Journal of Geo-Information*, vol. 7, no. 12, p. 463, Dec. 2018. [Online]. Available: https://www.mdpi.com/2220-9964/7/12/463

[45] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[46] A. R. Khairuddin, M. S. Talib, and H. Haron, "Review on simultaneous localization and mapping (SLAM)," in *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Nov. 2015, pp. 85–90.

[47] G. Klein and D. Murray, "Parallel Tracking and Mapping on a camera phone," in *2009 8th IEEE International Symposium on Mixed and Augmented Reality*. Orlando, FL, USA: IEEE, Oct. 2009, pp. 83–86. [Online]. Available: http://ieeexplore.ieee.org/document/5336495/

[48] J. D. Hol, T. B. Schon, F. Gustafsson, and P. J. Slycke, "Sensor Fusion for Augmented Reality," in *2006 9th International Conference on Information Fusion*, Jul. 2006, pp. 1–6.

[49] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Tokyo: IEEE, Nov. 2013, pp. 3923–3929. [Online]. Available: http://ieeexplore.ieee.org/document/6696917/

[50] M. Labbé and F. Michaud, "Online global loop closure detection for large-scale multi-session graph-based SLAM," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 2661–2666.

[51] "Light estimation - The basics of AR functionality | Coursera." [Online]. Available: https://www.coursera.org/lecture/ar/light-estimation-Qpv3O

[52] D. F. Cruz-Lunagomez, O. Alonso-Ramirez, A. Marin-Hernandez, F. M. Montes-Gonzalez, and L. F. Marin-Urias, "Floor plane recovery from monocular vision for autonomous mobile robot on indoor environments," in *CONIELECOMP 2011, 21st International Conference on Electrical Communications and Computers*, Feb. 2011, pp. 222–226.

[53] N. J. Mitra and A. Nguyen, "Estimating Surface Normals in Noisy Point Cloud Data," p. 7.

[54] D. Shin, "How does immersion work in augmented reality games? A user-centric view of immersion and engagement," *Information, Communication & Society*, vol. 22, pp. 1–18, Dec. 2017.

[55] W. Steptoe, S. Julier, and A. Steed, "Presence and discernability in conventional and non-photorealistic immersive augmented reality," in *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. Munich, Germany: IEEE, Sep. 2014, pp. 213–218. [Online]. Available: http://ieeexplore.ieee.org/document/6948430/

[56] J. Chen, G. Turk, and B. MacIntyre, "Watercolor Inspired Non-photorealistic Rendering for Augmented Reality," in *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '08. New York, NY, USA: ACM, 2008, pp. 231–234, event-place: Bordeaux, France. [Online]. Available: http://doi.acm.org/10.1145/1450579.1450629

[57] D. C. Duives, W. Daamen, and S. P. Hoogendoorn, "State-of-the-art crowd motion simulation models," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 193–209, Dec. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X13000351

[58] R. C. Shepherd, *Understanding Crowd Behaviours, Volume 2: Supporting Theory and Evidence.* [Online]. Available: https://www.academia.edu/1108101/Understanding_Crowd_Behaviours_Volume_2_Supporting_Theory_and_Evidence

[59] S. P. Hoogendoorn and P. H. L. Bovy, "Pedestrian route-choice and activity scheduling theory and models," *Transportation Research Part B: Methodological*, vol. 38, no. 2, pp. 169–190, Feb. 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0191261503000079

[60] H. Duncan, "Pre-event crowd behaviour prediction system: a study within concert environment," *Journal of Crowd Safety and Security Management*, vol. 1, pp. 33–45, Jan. 2009.

[61] F. E. H. Wijermans, *Understanding crowd behaviour: simulating situated individuals.* University of Groningen, SOM research school, 2011. [Online]. Available: https://www.rug.nl/research/portal/en/publications/understanding-crowd-behaviour(0d9bc1ee-8123-4780-b2bb-6ecd02762d56).html

[62] C. Mcphail, "The Myth of the Madding Crowd," *Social Forces*, vol. 71, Jan. 1991.

[63] L. Williams, "Pyramidal Parametrics," in *Proceedings of the 10th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '83. New York, NY, USA: ACM, 1983, pp. 1–11, event-place: Detroit, Michigan, USA. [Online]. Available: http://doi.acm.org/10.1145/800059.801126

[64] T. Giang, R. Mooney, C. Peters, and C. O. Sullivan, "ALOHA : Adaptive Level Of Detail for Human Animation Towards a new framework," p. 8, 2000.

[65] R. Fatima, I. Zarrin, M. A. Qadeer, and M. S. Umar, "Mobile travel guide using image recognition and GPS/Geo tagging: A smart way to travel," in *2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)*, Jul. 2016, pp. 1–5.

[66] Z. Gui, Y. Wang, Y. Liu, and J. Chen, "Mobile Visual Recognition on Smartphones," 2013. [Online]. Available: https://www.hindawi.com/journals/js/2013/843727/abs/

[67] B. Zhang, "Design of mobile augmented reality game based on image recognition," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, p. 90, Dec. 2017. [Online]. Available: https://doi.org/10.1186/s13640-017-0238-6

[68] R. Dahyot, M. Bourke, J. Byrne, J. Connolly, J. Su, D. Moloney, and V. Krylov, “Trinity College Dublin Drone Survey Dataset,” 2017, type: dataset. [Online]. Available: http://www.tara.tcd.ie/handle/2262/81836

[69] “Potree.” [Online]. Available: http://www.potree.org/

[70] J. Cheng, K. Chen, and W. Chen, “Comparison of marker-based AR and marker-less AR: A case study on indoor decoration system,” Jul. 2017.

[71] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar, and N. Navab, “Marker-less tracking for AR: a learning-based approach,” in *Proceedings. International Symposium on Mixed and Augmented Reality*, Oct. 2002, pp. 295–304.

[72] “Industrial Augmented Reality | Vuforia | PTC.” [Online]. Available: https://www.ptc.com/en/products/augmented-reality

[73] “Optimizing Target Detection and Tracking Stability.” [Online]. Available: https://library.vuforia.com/articles/Solution/Optimizing-Target-Detection-and-Tracking-Stability.html#

[74] “Extended Tracking.” [Online]. Available: https://library.vuforia.com/articles/Training/Extended-Tracking

[75] “Vuforia Object Scanner.” [Online]. Available: https://library.vuforia.com/articles/Training/Vuforia-Object-Scanner-Users-Guide

[76] “Model Target Generator User Guide.” [Online]. Available: https://library.vuforia.com/content/vuforia-library/en/articles/Solution/model-target-generator-user-guide.html

[77] “Smart Terrain.” [Online]. Available: https://library.vuforia.com/articles/Training/Getting-Started-with-Smart-Terrain

[78] “Augmented Reality tools comparison,” Jul. 2018. [Online]. Available: https://labs.imaginea.com/augmented-reality-tools-comparison/

[79] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, “MeshLab: an Open-Source Mesh Processing Tool,” *Eurographics Italian Chapter*

Conference, p. 8 pages, 2008. [Online]. Available: http://diglib.eg.org/handle/10.2312/LocalChapterEvents.ItalChap.ItalianChapConf2008.129-136

[80] B. Foundation, "blender.org - Home of the Blender project - Free and Open 3d Creation Software." [Online]. Available: https://www.blender.org/

[81] "Quickstart for Android | ARCore." [Online]. Available: https://developers.google.com/ar/develop/unity/quickstart-android

[82] "About AR Foundation | Package Manager UI website." [Online]. Available: https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.2/manual/index.html

[83] T. Olsson and M. Salo, "Online user survey on current mobile augmented reality applications," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, Oct. 2011, pp. 75–84.

[84] Z. He, L. Wu, and X. Li, "When art meets tech: The role of augmented reality in enhancing museum experiences and purchase intentions," *Tourism Management*, vol. 68, Mar. 2018.

Appendix A

Links to Demo Video -
http://tiny.cc/ARDoorDemoVideo
Link to Source Code -
http://tiny.cc/ARDoorSourceCode
Link to Android application with crowd destroyer -
http://tiny.cc/ARDoorWithCrowdDestroyer
Link to Android application without crowd destroyer -
http://tiny.cc/ARDoorWithoutCrowdDestroy

Appendix B

Figure 1: Original model of TCD campus

Figure 2: Cropped model of Front Square undecimated

Figure 3: Model decimated with a ratio of 0.8

Figure 4: Model decimated with a ratio of 0.6

Figure 5: Model decimated with a ratio of 0.4

Figure 6: Model decimated with a ratio of 0.2

Figure 7: Quality comparison of original TCD model

Figure 8: Quality comparison of Model decimated with a ratio of 0.8

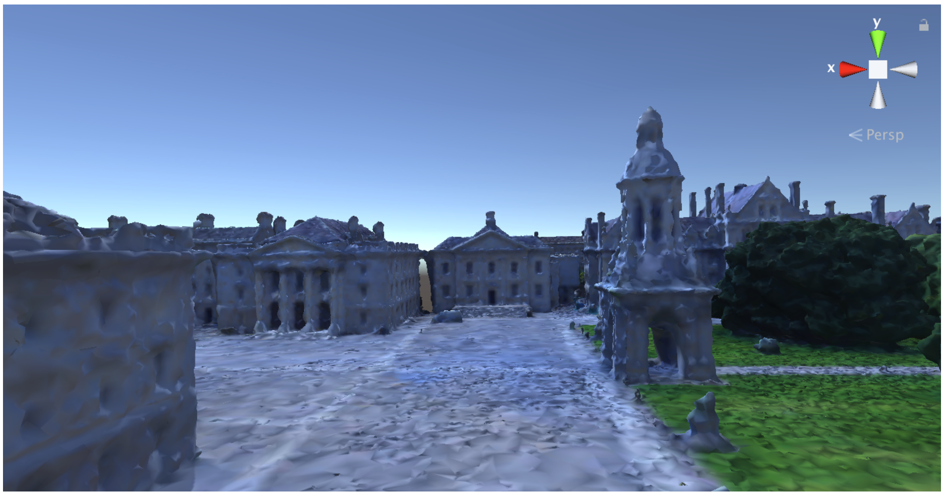

Figure 9: Quality comparison of Model decimated with a ratio of 0.6

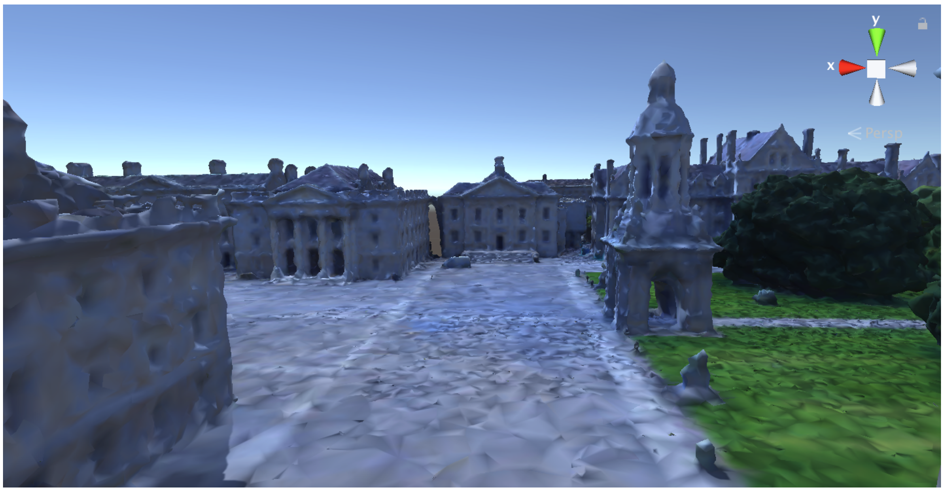

Figure 10: Quality comparison of Model decimated with a ratio of 0.4

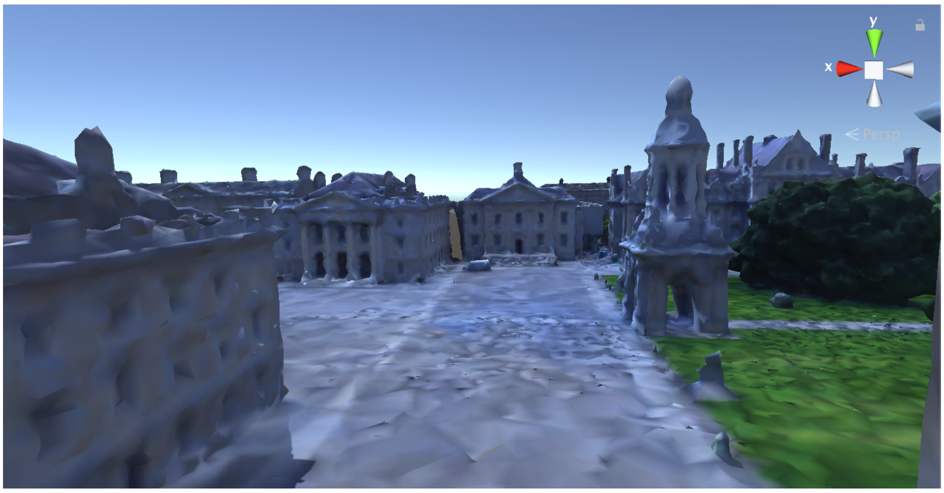

Figure 11: Quality comparison of Model decimated with a ratio of 0.2

Appendix C

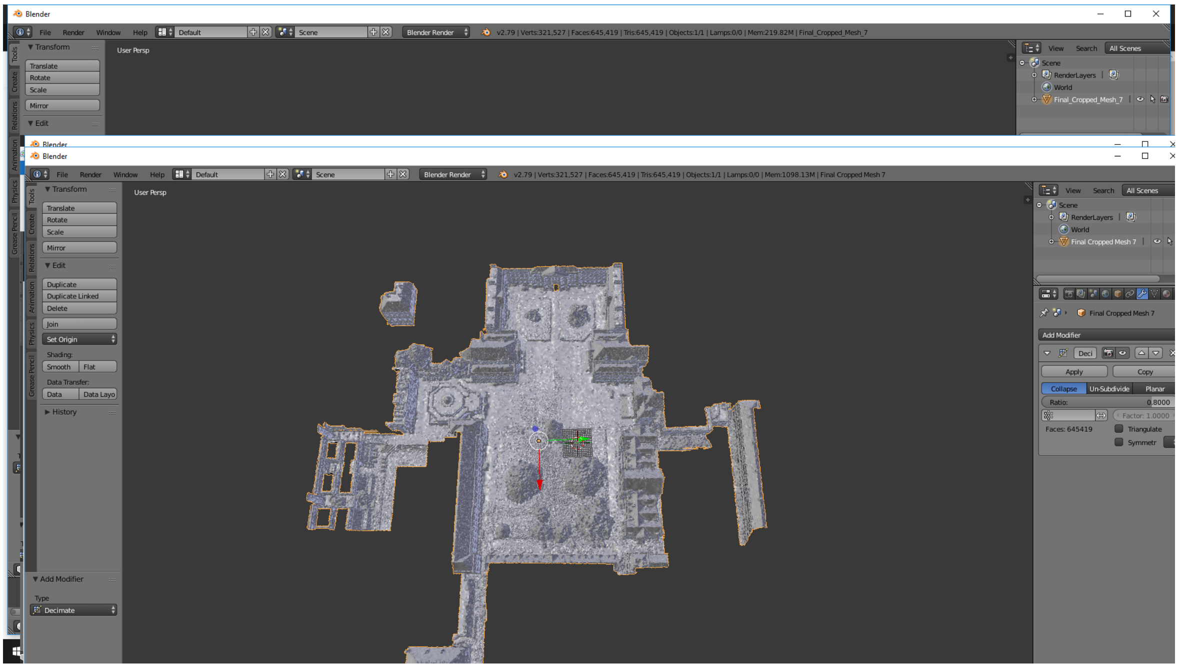

Figure 12: Blender Decimate Modifier settings for 0.8 ratio

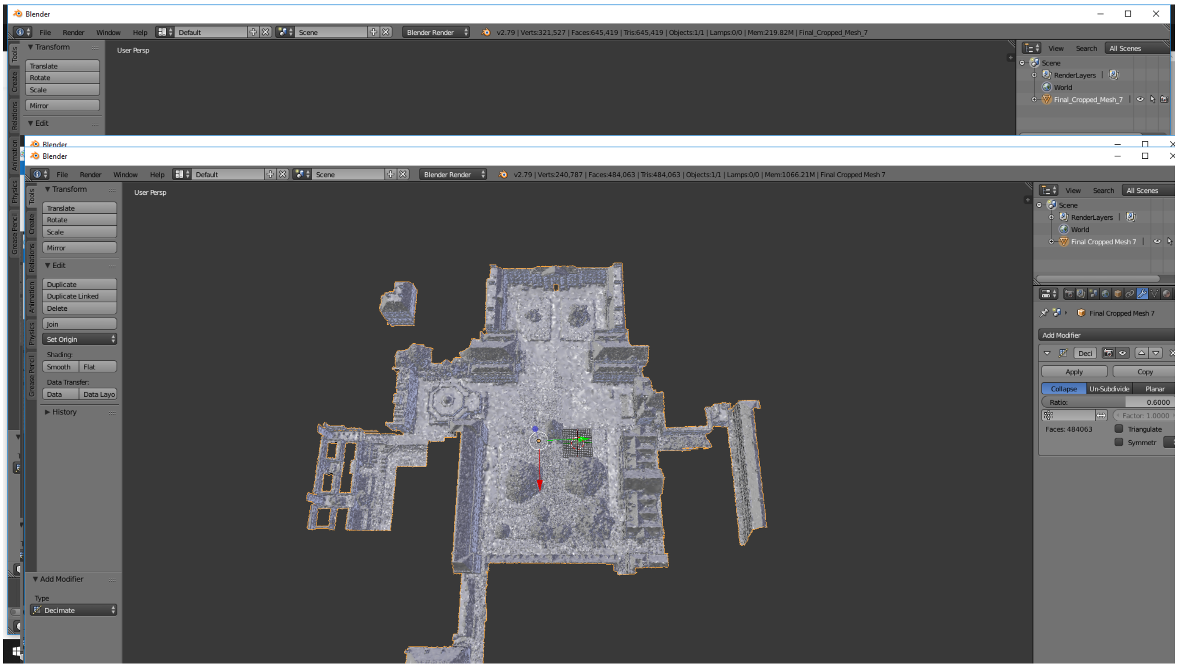

Figure 13: Blender Decimate Modifier settings for 0.6 ratio

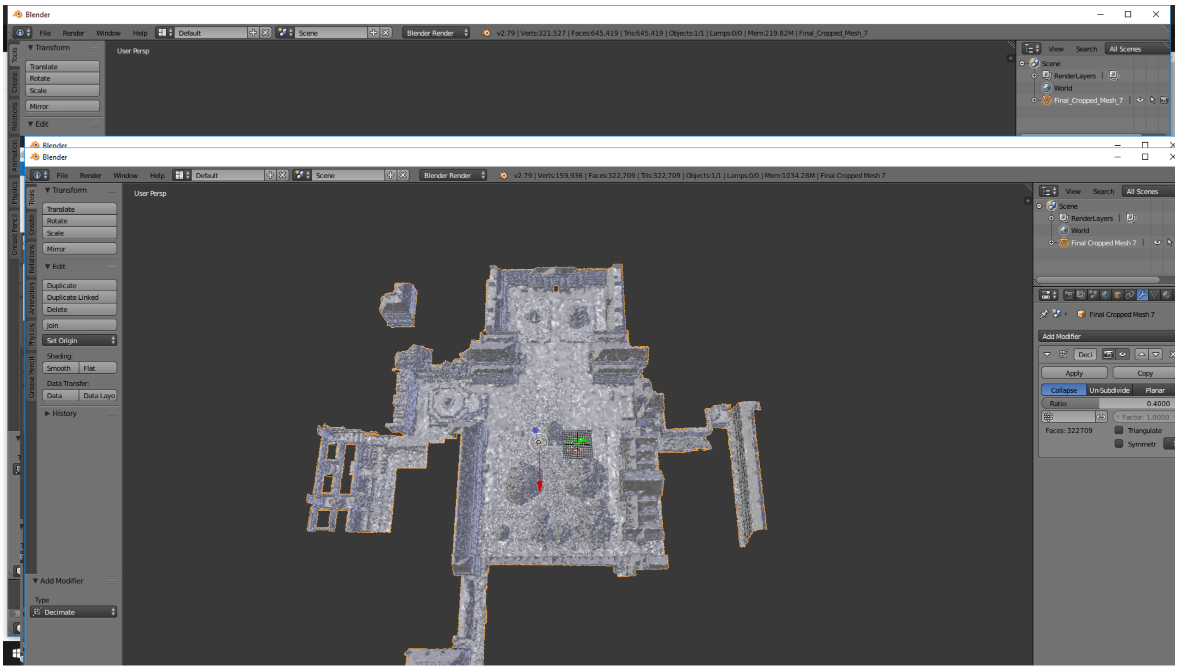

Figure 14: Blender Decimate Modifier settings for 0.4 ratio

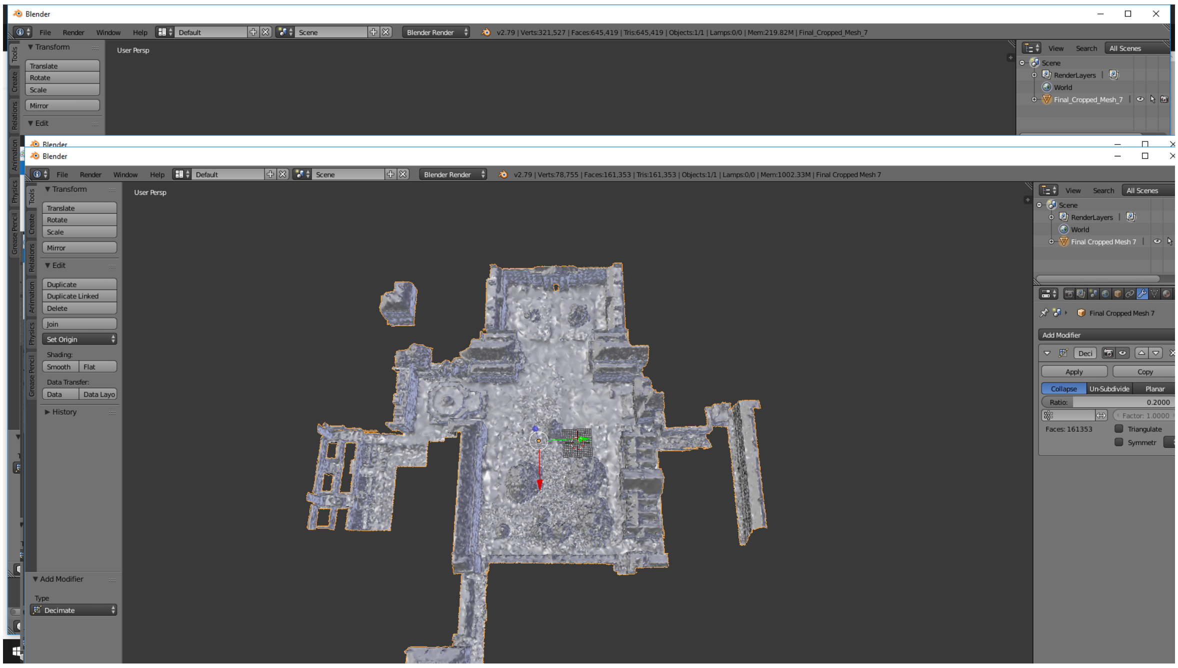

Figure 15: Blender Decimate Modifier settings for 0.2 ratio

Appendix D

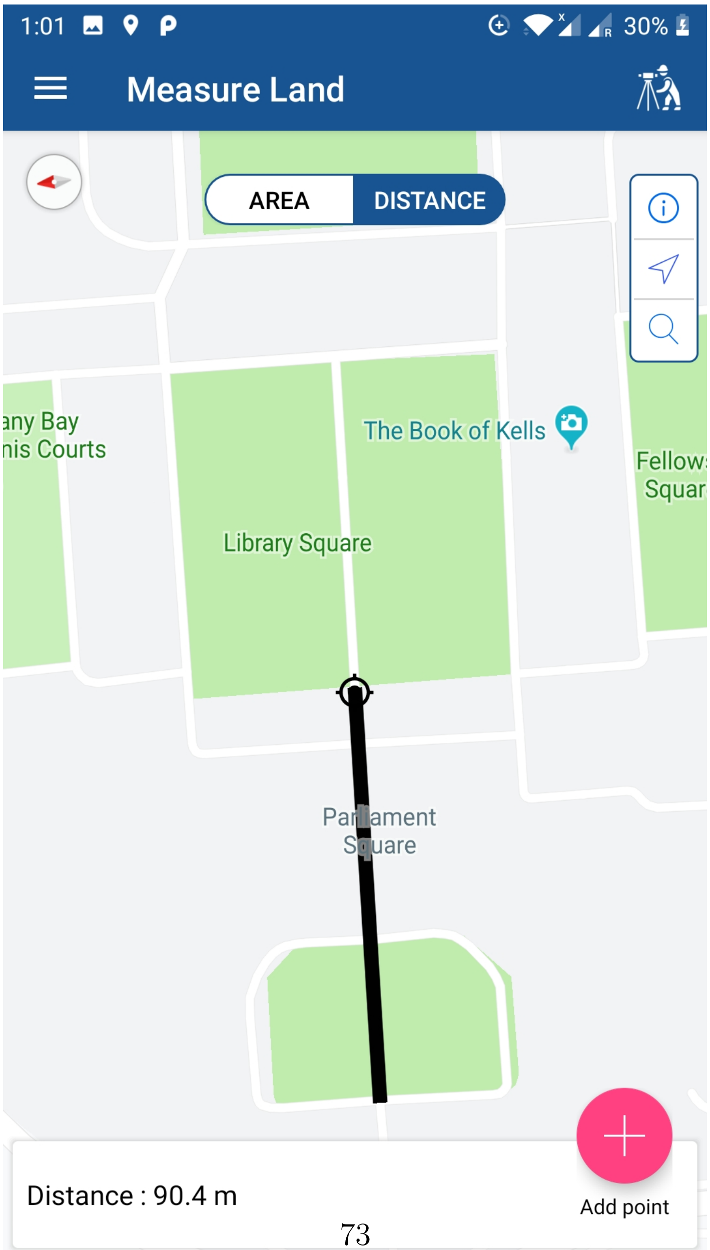

Figure 16: Blender Decimate Modifier settings for 0.8 ratio

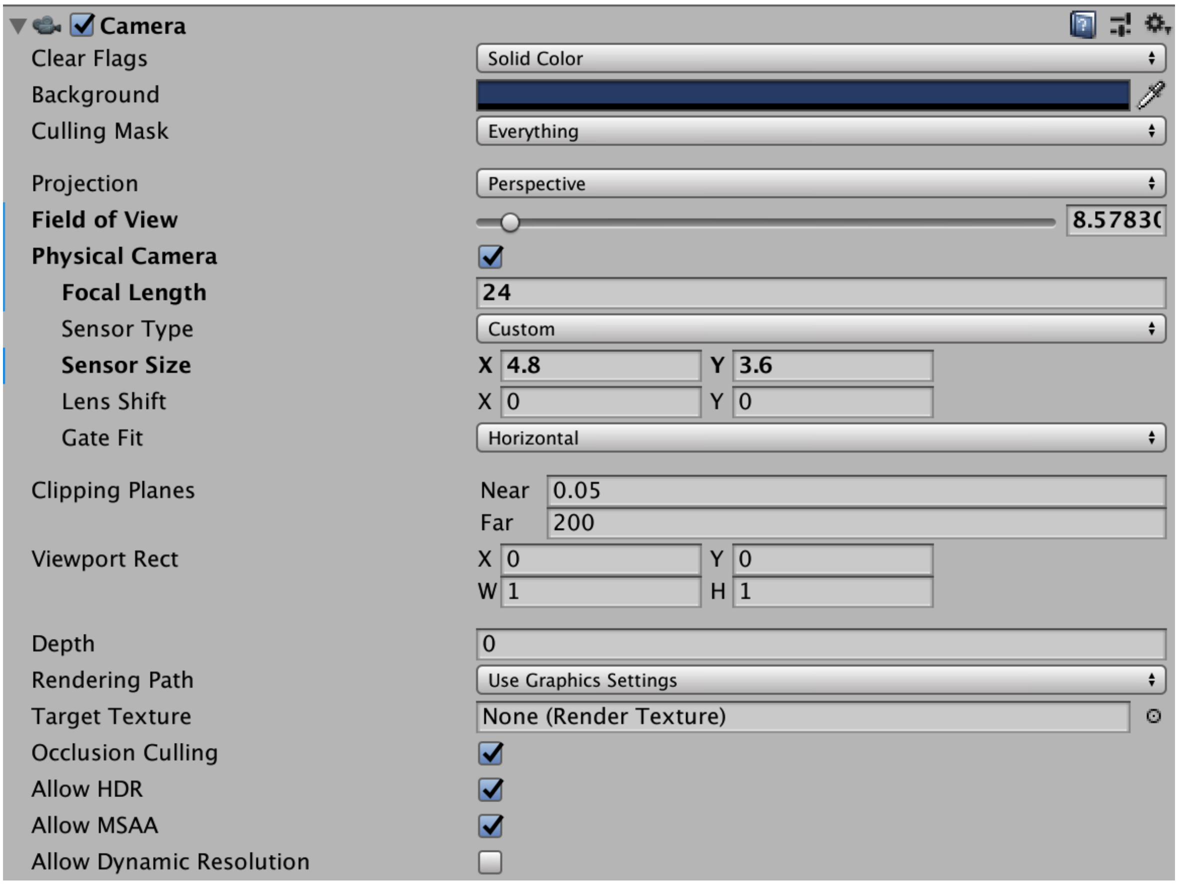

Figure 17: Physical camera parameters of the OnePlus 5 Camera fed into Unity

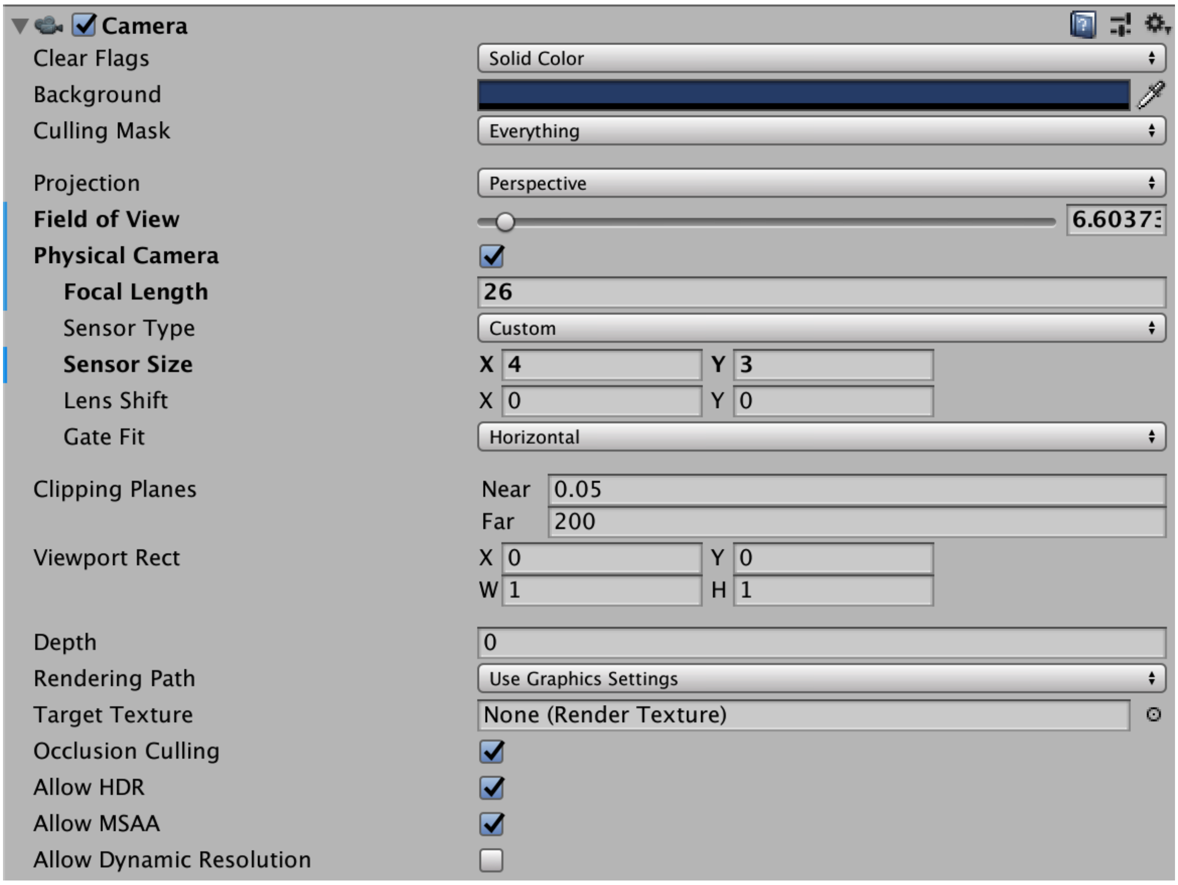

Figure 18: Physical camera parameters of the Pocophone F1 Camera fed into Unity

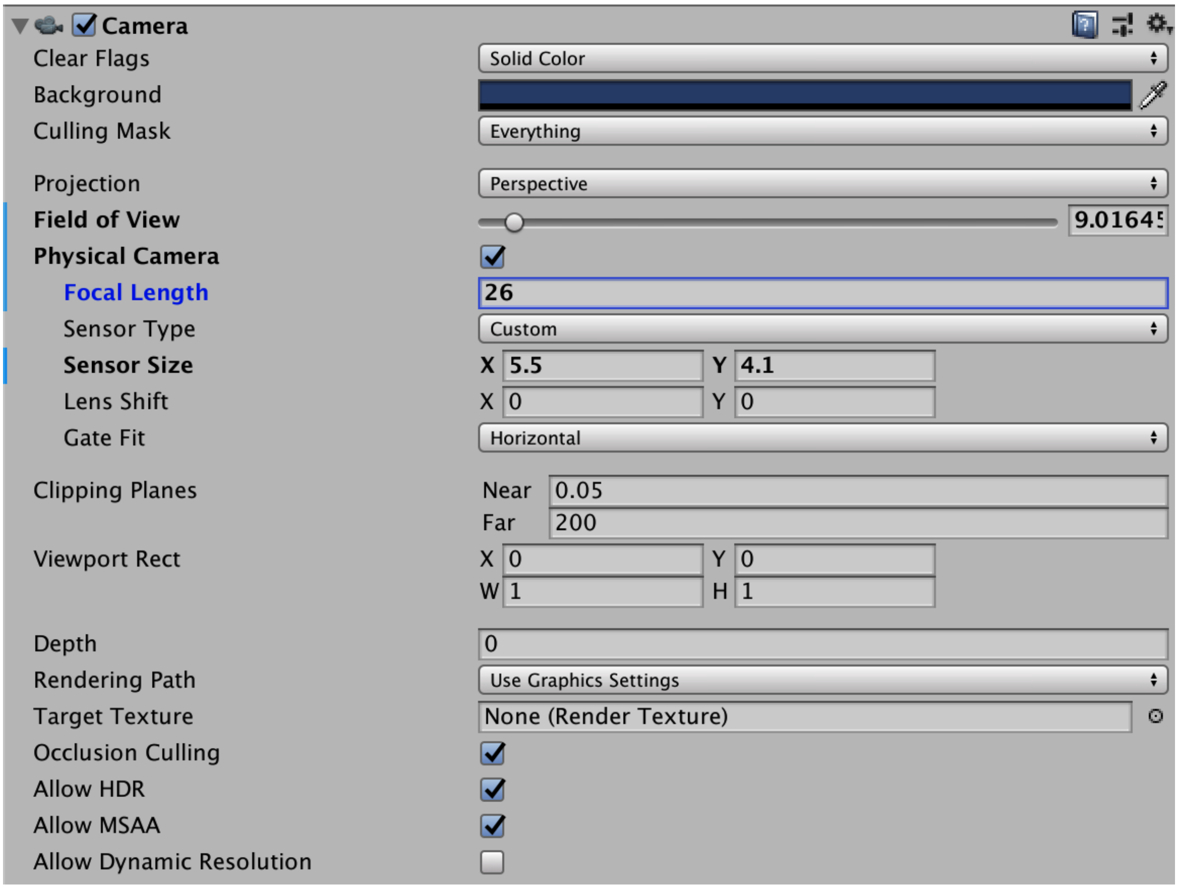

Figure 19: Physical camera parameters of the Samsung Galaxy Note 9 Camera fed into Unity

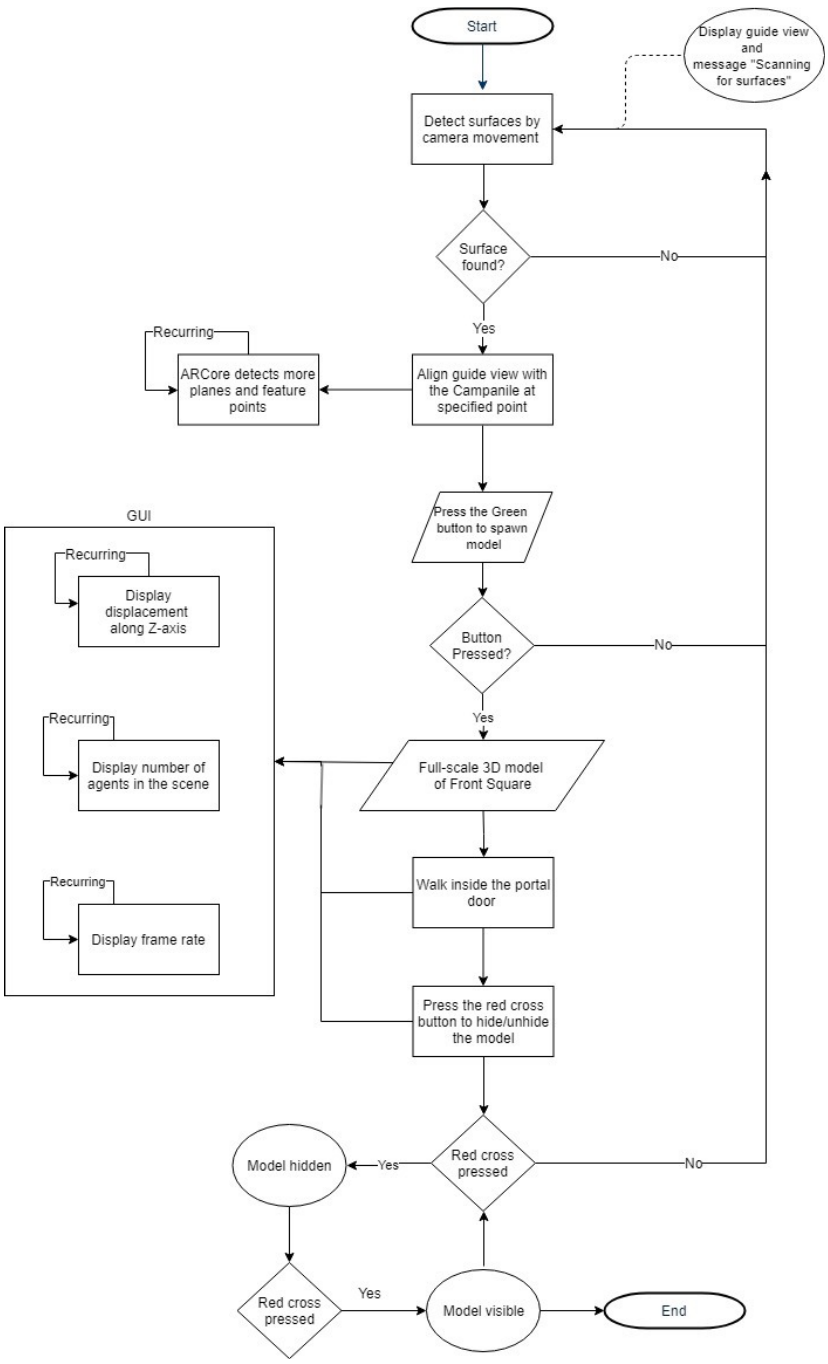

Figure 20: Application Operation Flow