

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Virtual/Augmented Reality for Infrastructure Management of Road Networks

Joel Márcio Torres Carneiro

DISSERTATION



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Rosaldo J. F. Rossetti

June 27, 2018

Virtual/Augmented Reality for Infrastructure Management of Road Networks

Joel Márcio Torres Carneiro

Mestrado Integrado em Engenharia Informática e Computação

June 27, 2018

Abstract

The management and maintenance of road infrastructures, which include pavements, vertical and horizontal signage, traffic lights, access for people with physical disabilities, among others, require the use of a vast amount of information about their maintenance history and current situation. This information is mainly geo-referenced and maintained in geographic information systems to facilitate the access to it. However, nowadays the majority of the information still is stored on paper which makes difficult the access. Furthermore, this is not only applicable to the road infrastructures but also to all the infrastructures of our cities.

Indeed, it is necessary to make smarter the management and maintenance of infrastructures so that it is not only easier to access the data, but also to predict changes in due time. Thus, it is necessary to continuously monitor the conditions of the infrastructure, that is, monitor the indicators of degradation status, identification of faults or possible defects in infrastructure, among others. Tools are also needed to enable the addition of information when repair and maintenance work is performed, representing an important instrument for data collection.

This project studies the potential technical applications of Virtual and Augmented Reality (VR/AR) as a support to the management and maintenance of the road infrastructures. The developed applications use mobile platforms with video cameras for introducing and manipulate virtual objects while the real environment and its infrastructures are visualized. Thereby, the apps identify and present information about the infrastructures in the scene.

Several studies have been done to better integrate the Geographic Information System (GIS) and Building Information Modelling (BIM) technologies. Unfortunately, there is no standard ontology or some kind of method capable of doing it without the loss of important data. Thereby, it is proposed a new approach for integrating BIM and GIS, also using the IoT and VR/AR technologies leading to a smarter monitoring and a better visualisation.

As concept proof, two apps were developed. One allows for the introduction of virtual signals in a place without signage to a better evaluation of the future signal's positioning. The other application displays the information about an infrastructure when the user aims its mobile device's camera to it. The app uses a server, based on GIS and BIM data. Therefore, the data model has an association with the BIM technology, which describes all stages of the infrastructure's development and maintenance. Moreover, the system can be integrated with the IoT technology, which focuses on enabling communication between all objects either real or virtual.

The two applications were evaluated taking into account the AR's definition, where the system must mix the virtual and real worlds, be interactive in real-time, and the real and virtual objects must be aligned in 3D. Furthermore, it was evaluated if the displayed information about the infrastructure is correct and if it effectively belongs to the infrastructure in the scene, and the distance associated errors.

Keywords: AR, VR, BIM, GIS, IoT, smart infrastructure management, maintenance, management, smart cities.

Resumo

A gestão e manutenção das infraestruturas rodoviárias, que incluem pavimentos, sinalização vertical e horizontal, semáforos, acesso para pessoas com deficiência física, entre outros, requerem o uso de uma grande quantidade de informações sobre o histórico de manutenção e sobre a sua situação atual. Esta informação é principalmente georreferenciada e mantida em sistemas de informação geográfica para facilitar o seu acesso. No entanto, hoje em dia quase todas as informações são armazenadas em papel, o que dificulta esse mesmo acesso. Além disso, não é apenas aplicável às infraestruturas rodoviárias, mas também a todas as infraestruturas das nossas cidades.

Na verdade, é necessário tornar mais inteligente o gerenciamento e a manutenção das infraestruturas, de modo a que não seja apenas mais fácil aceder aos dados, mas também prever mudanças atempadamente. Assim, é necessário monitorar continuamente as condições da infra-estrutura. Ferramentas para permitir a adição de informações quando o trabalho de reparação e manutenção é realizado são também necessárias, representando um instrumento importante para a recolha de dados.

Este projecto estuda as potenciais aplicações técnicas da Realidade Virtual e Aumentada (VR/AR) como suporte à gestão e manutenção das infra-estruturas rodoviárias. As aplicações desenvolvidas utilizam plataformas móveis com câmeras de vídeo para introduzir e manipular objetos virtuais, enquanto o ambiente real e as suas infraestruturas são visualizados. Assim, as apps identificam e apresentam informações sobre as infraestruturas na cena.

Diversos estudos foram realizados para integrar da melhor forma as tecnologias Geographic Information System (GIS) and Building Information Modelling (BIM). Infelizmente, não há uma ontologia padrão ou algum tipo de método capaz de fazê-lo sem a perda de dados importantes. Assim, propõe-se uma nova abordagem para integrar BIM e GIS, utilizando também as tecnologias IoT e VR/AR, levando a um monitoramento mais inteligente e uma melhor visualização.

Como prova de conceito, duas aplicações foram desenvolvidas. Uma permite a introdução de sinais virtuais num local sem sinalização para uma melhor avaliação do posicionamento de um sinal futuro. A outra aplicação exibe as informações sobre uma infra-estrutura quando o utilizador direciona a camera do seu dispositivo móvel para a infra-estrutura. Esta usa um servidor, com base em dados GIS e BIM. Portanto, o modelo de dados tem uma associação com a tecnologia BIM, que descreve todas as etapas de desenvolvimento e manutenção da infra-estrutura. Além disso, o sistema pode ser integrado à tecnologia IoT, que se concentra em permitir a comunicação entre todos os objetos, seja real ou virtual.

As duas aplicações foram avaliadas levando em conta a definição da realidade aumentada, onde o sistema deve misturar os mundos virtual e real, ser interativo em tempo real, e os objetos reais e virtuais devem estar alinhados em 3D. Além disso, foi também avaliado se as informações exibidas sobre a infra-estrutura estão corretas e se pertencem efetivamente à infra-estrutura na cena, e os erros de distância associados.

Palavras-chave: AR, VR, BIM, GIS, IoT, gestão inteligente de infraestruturas, manutenção, gestão, cidades inteligentes.

Acknowledgement

I would like to thank the Faculty of Engineering of the University of Porto by making available the resources for this research, being an important support to the development of this study. The Armis company also provided favourable conditions to the development of the concept of proof, being a crucial help on the project. I also would like to thank my supervisor Rosaldo Rossetti for the tips and all the support, to my company's supervisors José Macedo and Guilherme Soares for the counselling and guiding. And finally, I want to thanks my parents, my closest friends, and my girlfriend for all the comprehension in the difficult moments.

Joel Márcio Torres Carneiro

*“I have no special talents.
I am only passionately curious.”*

Albert Einstein

Contents

1	Introduction	1
1.1	Problem	2
1.2	Motivation and Objectives	2
1.3	Proposed Solution	2
1.4	Context and Background	3
1.5	Structure of the Dissertation	3
2	Systematic Literature Review	5
2.1	Research	7
2.2	Building Information modelling (BIM)	7
2.3	Geographic Information System (GIS)	9
2.4	Differences and similarities between BIM and GIS	10
2.5	BIM and GIS Integration	12
2.5.1	BIM-GIS at the Application Level	12
2.5.2	BIM-GIS at the Data Level	13
2.5.3	BIM-GIS at the Process Level	15
2.6	Internet of Things (IoT)	17
2.7	IoT, BIM and GIS Integration	18
2.8	Virtual and Augmented Reality (VR/AR)	21
2.8.1	Computer Vision	23
2.8.2	Computer Vision Algorithms in Mobile Devices	24
2.9	AR/VR, IoT, BIM and GIS Integration	26
2.10	Summary	27
3	Methodological Approach	33
3.1	Problem Statement	33
3.2	Proposed Solution	34
3.3	ARIMaRN-PSAL	34
3.3.1	Ground Detection	34
3.3.2	Traffic Signal	34
3.4	ARIMaRN	35
3.4.1	The Database	35
3.4.2	Server	36
3.4.3	AR Visualisation	36
3.5	Application Metrics	36
3.6	Technologies	37
3.6.1	Global Positioning System Background	37
3.6.2	Augmented Reality Frameworks	41

CONTENTS

3.6.3	Geolocation Frameworks	43
3.7	Tools	44
3.7.1	Revit Software	44
3.7.2	AutoCAD Software	45
3.7.3	QGIS Software	45
3.7.4	Unity	45
3.7.5	Microsoft Visual Studio	46
3.7.6	Android Studio	46
3.7.7	Databases	47
3.7.8	Server	48
3.7.9	Eclipse	49
3.8	Summary	49
4	Implementation	53
4.1	ARIMaRN-PSAL	53
4.1.1	Unity Setup	53
4.1.2	Scene	53
4.1.3	Scripts	54
4.1.4	Application's Build and Run	54
4.2	ARIMaRN	55
4.2.1	Architecture	55
4.2.2	Database	55
4.2.3	Server	60
4.2.4	AR Integration	67
4.3	Summary	75
5	Case Study	77
5.1	Database	77
5.1.1	BIM Database	77
5.1.2	GIS Database	78
5.1.3	Coordinates' Mapping	78
5.2	Results	81
5.2.1	ARIMaRN-PSAL	81
5.2.2	ARIMaRN	82
5.3	Summary	87
6	Conclusions	89
6.1	Main Contributions	91
6.1.1	ARIMaRN-PSAL	91
6.1.2	ARIMaRN	92
6.2	Further Development	93
6.2.1	ARIMaRN-PSAL	93
6.2.2	ARIMaRN	93
6.3	Future Research Lines	94
	References	97

CONTENTS

A Database	109
A.1 BIM database	109
A.2 Server	111
A.2.1 Configuration	111
A.3 Database RNSFP	113
A.3.1 BIM data	113
A.3.2 GIS data	114
A.4 ARIMaRN-PSAL Setup	116
A.4.1 Wikitude Approach	116
A.5 ARIMaRN Setup	118
A.5.1 Wikitude Approach	118
A.5.2 Mapbox Approach	119

CONTENTS

List of Figures

2.1	BIM’s Levels of Development	9
2.2	GIS’s Levels of Detail	10
2.3	IoT cloud architecture	18
2.4	AR System Components	23
3.1	ARIMaRN Application Architecture	35
4.1	ARIMaRN Application Detailed Architecture	56
4.2	BIM and GIS Added Tables	59
4.3	Database Architecture	60
4.4	Spring Boot, JPA, and Hibernate Architecture	61
4.5	ARIMaRN Server Architecture	67
4.6	Viewing of the Angle’s deviation according to the GPS Associated Error	69
4.7	ARIMaRN Main Application	70
4.8	Virtual Camera Rotation According to the Mobile Device’s Compass	73
4.9	Mapping the Real Coordinates to the Unity world	74
5.1	RNSFP’s AutoCAD File	78
5.2	AutoCAD <i>Dimension</i> Tool Orientation Angles of RNSFP	79
5.3	Mapping the Real Coordinates to the Unity World	80
5.4	Print of the ARIMaRN-PSAL simulation	81
5.5	Server Data Logs	82
5.6	Creating Scene Logs	83
5.7	Moving Camera and Changing scene according to user’s Localisation	83
5.8	Traffic Light BIM data Information	84
5.9	Distance error	85
5.10	Traffic light orientation error between the upper traffic light and the virtual one	85
5.11	Plot of the accuracy influence of the GPS readings	86
A.1	Revit sample UML diagram cut with an emphasis in the Material table and its relationships	109
A.2	BIM and GIS used Tables from Armis Database	110
A.3	bim_crosswalks table data	113
A.4	bim_espiras table data	113
A.5	bim_materials table data	114
A.6	bim_intersection table data	114
A.7	arm_traffic_light table data	114
A.8	gis_crosswalks table data	115
A.9	gis_espiras table data	115

LIST OF FIGURES

A.10 Wikitude SLAM Application screen	118
---	-----

List of Tables

2.1	Description of the Developed Features using the Discussed Technologies	28
2.2	Features Categorization	29
2.3	The most successful integrations with the highest relevance in the infrastructures management and maintenance	30
3.1	ARCore Android Devices Supported Models	44
3.2	Supported plugins, APIs and IDEs	44
3.3	SDK or Library Main Features	44

LIST OF TABLES

Abbreviations

GIS	Geographic Information System
BIM	Building Information Modelling
IoT	Internet of Things
AR	Augmented Reality
VR	Virtual Reality
IS	Information Systems
FAST	Accelerated Segment Test
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features
CNN	Convolutional Neural Networks
PDB	Armis' Porto City Hall Database
MIEIC	Master in Informatics and Computing Engineering
RNSFP	Rua Nossa Senhora de Fátima of Porto
ARIMaRN	Augmented Reality for Management of Road Networks
ARIMaRN-PSAL	Augmented Reality for Management of Road Networks - Place Signage at Location

Chapter 1

Introduction

The enormous technological advance in the last few years brings many opportunities, especially in the smart cities context. The smart cities holistic concept presents a city connected through a network of sensors capable of gathering information about all of the city's infrastructures. Furthermore, all the infrastructures in the city are intelligent. Thereby, they are capable of doing their management and maintenance. The infrastructures have access to a vast set of sensors transmitting information about them and the environment, bringing a lot of possibilities when it comes to the monitoring, maintenance, and management of the infrastructures. Thus, the smart infrastructures will have access to their parameters, such as the energy consumption, temperature, water levels, among others, and can also use that information to decision-making.

Nowadays, we are far from the smart cities concept. However, several efforts have been made to reach the smart cities concept holistically. Indeed, some technologies have been developed in the construction area to get 3D models and specific information about our infrastructures. The standard technologies used in construction are the Building Information Modelling (BIM) and the Geographic Information Systems (GIS). However, these two must be integrated to give us a tool to access all the available information about infrastructures. Unfortunately, the integration between the BIM and GIS technologies is a big puzzle for the developers. The best works towards that integration always fail in some conversion, or something else, resulting in loss of essential data. The biggest problem is the interoperability issues between the two technologies due to the different technologies' backgrounds and different ways of representing the data.

The Internet of Things (IoT) paradigm and the Virtual and Augmented Reality (VR/AR) technology integrated with BIM and GIS technologies can also result in having the data updated in real-time and in better visualisation of the data. Thereby, with these technologies is possible to create new tools to monitor our infrastructures better.

1.1 Problem

The road infrastructures' management and maintenance, which include vertical and horizontal signage, traffic lights, among others, require a vast amount of information about the maintenance history and the current situation of the infrastructures. However, the information about the life-cycle of the infrastructures is mostly stored on paper, resulting in difficult access to data. Thus, a smarter management and maintenance of the road infrastructures are necessary to make easier the access to the data and also to predict events.

The construction and the management of road infrastructures have many stakeholders involved and it can be a big puzzle to share all the essential information to the respective stakeholder. Moreover, any error in the infrastructure is not automatically communicated to all the stakeholders, or even to any of them. Thus, it is necessary to build smart systems capable of managing and do the management and maintenance of the road infrastructures. Thereby, is demanding a full integration of BIM and GIS technologies alongside with IoT and VR/AR to reach the wanted smart system.

1.2 Motivation and Objectives

Several works have been done to achieve a good integration between BIM and GIS. However, due to their interoperability problems, all of the systems lose important data. Furthermore, some studies have also been done to integrate the IoT and the VR/AR to have real-time updated models and better visualisations.

Despite these efforts, it is not known an integration that uses the four technologies together in the same system. Many approaches tried the BIM-GIS integration, others the BIM-GIS-IoT integration, others the BIM-IoT-AR/VR integration, and others the GIS-IoT-AR/VR integration. This integration brings new kinds of analysis and data collection. Moreover, the system allows their use in real-time. Thereby, we can have a tool to smarter monitor all steps of the infrastructure management and maintenance. The system also brings essential advances to have cities with better sustainability, less energy consumption, infrastructures crash prediction, air quality evaluation, among many others possible analyses. Looking in a broad point of view, these advances lead us to better world sustainability, improving our health as well as our way of life.

1.3 Proposed Solution

There were developed two applications. One of them has a database with the BIM and GIS data. The information provided by the IoT sensors can update in real-time the data of the BIM-GIS database. The visualisation of the data and interaction with the system is performed by a mobile application that communicates with a developed server to gather the necessary information. Therefore, the app will display the information about one infrastructure when the user aims its mobile device's camera to it. Hence, the user can see real-time information about the infrastructure.

Introduction

The other application allows the inserting of virtual signals in a place without signage to a better evaluation of the positioning of a future signal to that place.

To evaluate the system is taking into account the AR 's definition, where the system must mix the virtual and real worlds, be interactive in real-time, and the real and virtual objects must be aligned in 3D. Furthermore, it was evaluated if the displayed information about the infrastructure is correct and if it effectively belongs to the infrastructure in the scene. The distance's error between the real and the virtual infrastructures was also studied.

1.4 Context and Background

The developed system uses the Virtual and Augmented Reality to display the information about the road infrastructures. Moreover, a database with geo-referenced data is the base of the displayed information. Thereby, the system belongs to the Information Systems (IS) area and, inside of it, we can specify the Geographic Information Systems area.

The system results from the partnership with the Armis Group company. Moreover, it is also the basis of a starting project in the company. The Armis company aims to develop robust technological solutions in the information technology area as well as in the intelligent transports area. Therefore, the management and maintenance of the road infrastructures are a concern to the company. Thanks to that, the idea to develop a system for smarter monitoring of those infrastructures started to have shape, and the development of the presented system was the first step to the company's project.

1.5 Structure of the Dissertation

The Dissertation is structured as follows: Chapter 2 presents state of the art, where are analysed several papers to a better comprehension of the developed system concerns. Chapter 3 shows the methodology used and the system architecture. In the Chapter 4 is presented how was performed the implementation and also more detailed information about the methodology and architectures of the systems. Chapter 5 shows the case of study where the two applications were tested and the results. Finally, in the Chapter 6 the conclusions are discussed and is also highlighted the future work.

Introduction

Chapter 2

Systematic Literature Review

In this Chapter, is discussed how possible it is to make the road infrastructures maintenance and management smarter and which are the obtained improvements from integrating technologies such as AR/VR, BIM, GIS, and IoT. Furthermore, it will be discussed some existing methods towards that integration. However, in the analysed studies, nowhere are the four technologies integrated into a full stack. Many approaches tried the BIM-GIS integration, others the BIM-GIS-IoT integration, and others BIM-IoT-AR/VR or GIS-IoT-AR/VR integration.

The practice of this integration will allow for new kinds of data analysis, data collection as well as use them in real-time. Thus, it is possible to monitor all steps of the infrastructure management and maintenance in a smarter way. Hence, it brings significant advances to have cities with more sustainability, less energy consumption, infrastructures crash prediction, and so on, which also leads to better global sustainability.

The smart cities concept is increasingly a real thing in our lives. Cities with sensors and smart management and maintenance allow a greener world due to the environmental parameters control. With smart cities also comes the intelligent buildings and the smart infrastructures. Thus, it is demanding some advances in our technologies to reach the smart cities holistic concept.

Continuously updated models are needed to have better models of our infrastructures. We can do this work manually, but it takes too long and has high costs, or we can use the emergent concept of smart cities. The management and maintenance of road infrastructures require the use of a vast amount of information about their maintenance history and current situation. However, nowadays, almost all the information is difficult to access due to the interoperability issues between software programs and also due to, until now, paper stores nearly all the information. That is not only applicable to the road infrastructures. Indeed, all of them have the same needs. Thereby, it is necessary to make smarter the management and maintenance of infrastructures so that it is not only easier to access the data, but also to predict changes and add data in due time. Thus, it is necessary to monitor the conditions of the infrastructure continuously, that is, monitor the indicators of degradation status, identification of faults or possible defects in one infrastructure, among others.

Systematic Literature Review

With real-time updates of our infrastructures' parameters and elements, nearly real-time actions can avoid some disasters or even catastrophes. Tools are also needed to enable the addition of information when repair and maintenance work is done, representing an important instrument for data collection, especially in the first times when some infrastructures are not capable of doing it by themselves.

Building Information Modelling (BIM) is the standard in the construction industry for the last two decades [PSCD17]. BIM allows the creation of the different stages' model of the infrastructures in their life-cycle. Wherefore, BIM can be a useful tool to smarter management of our infrastructures through the 3D modelling and its vast construction information either geometric and semantic. However, it does not include the surrounding spatial information [RDFS14]. A Geographic Information System can overcome this lack. BIM also allows the addition of additional dimensions, like cost, energy consumption, and so on, to the infrastructures' models. Hence, we can analyse, for instance, the energy consumption levels of one infrastructure to a better management.

A Geographic Information System (GIS) lets us visualise, manage, store and analyse spatial data [MN14, DCA16b]. GIS can do spatial analysis based on the functional and physical relationship of the environment outside of the building and can do it at large spatial scale. GIS data are also geo-referenced [SC06], while in BIM the data aren't related to the real-world coordinates. However, in contrast to BIM, it lacks the vastly detailed building components and project information [LWW⁺17]. Thus, GIS allows us to have access to the real world infrastructures, in its coordinates, but we cannot have detailed information about its indoors.

Internet of Things (IoT) focuses on enabling communication between all objects either real or virtual. Some examples are vehicles, buildings and other physical objects that have some sensors and network connection capable of transmitting data. Thus, these objects can interact with each other inside the network and with some designed services [Isi15, BAS⁺17]. The technology advance allowed the appearance of the first IoT sensors in our lives. Wherefore, we can monitor our infrastructures in a smarter way which provides less work and also lower costs. However, some security, usability, privacy, and interoperability issues are a barrier to the IoT use and development.

In the Virtual Reality (VR) the user is in a computer-generated environment, while in Augmented Reality (AR), are presented objects from a computer-generated environment to the user in the real environment. The AR can connect the virtual world objects and the real world ones, spatially and cognitively speaking [SH17].

Having in mind the four different technologies, we can predict a full integration that can bring significant improvements and advances when we are talking about maintenance and management of infrastructures. Several works have been done to integrate BIM and GIS technologies, BIM and GIS with IoT, and AR/VR with all the others. Nevertheless, no one could achieve full integration of these, especially without losing valuable data. Among these technologies, the most challenging integration is the BIM and GIS integration. The two technologies have similar aims in some aspects. However, the background of each one is way different [ARMN16a]. Also, the supported files and the programming language are different [LSZJ14]. Hence, the integration is hard work,

and it is almost sure that we will lose some data. The integration of the four technologies described above can leverage us into the smart cities holistic concept. This integration allows the AR visualisation of the detailed infrastructures information (BIM-GIS) in real-time (IoT). Thus, all steps towards that integration are steps to a greener and smarter world.

The remainder of this chapter is structured as follows: Section 2.2 describes with more detail the BIM and its development. Section 2.3 highlights the GIS functionalities and its details. Section 2.4 exposes the differences and similarities between BIM and GIS. Section 2.5 presents the work that has been done until the writing of this document to integrate these two technologies. Section 2.6 focuses on the IoT paradigm and details it. Section 2.7 presents the work that has been done until the writing of this document to integrate IoT with BIM or GIS. Section 2.8 will describe VR/AR with some detail. Section 2.9 presents the work that has been done until the writing of this document to integrate VR/AR with BIM or GIS. The Section 2.8.1 describes the computer vision algorithms broadly. Section 2.8.2 presents the work that has been done until the writing of this document with the computer vision algorithms to use them in mobile devices. Section 2.10 summarizes the related work studied and brings the suggestions and the conclusions of this research.

2.1 Research

This study was performed using the Systematic Literature Review methodology. Papers from conferences and journals related to the BIM, GIS, IoT, AR/VR and Computer Vision technologies are the base of this research. After finding an article, if its keywords and abstract were related to the technologies in which we are interested, that paper is analysed. This study is based on about sixty-five papers to have a good result in our research.

2.2 Building Information modelling (BIM)

Building Information Modelling (BIM) is the standard in the construction industry for the last two decades [Isi15]. BIM allows building the different stages' model of the infrastructures in their life-cycle. Indeed, BIM is not only a software tool, but it is also a methodology that cooperates with the workflow and with the delivery process of the building project. The addition of other dimensions of the project, like costs, energy simulation, security, acoustics, among others is another benefit of BIM [TB09]. Moreover, during each phase of the building's life-cycle engineering analyses can be performed [LWW⁺17]. A good example is the infrastructure energy consumption analysis, allowing a better understanding of the infrastructures' energy expenditure.

BIM technology has a set of processes, policies, and technologies to manage the facilities' project data and its design process [How06]. BIM also allows finding the relationships between its different layouts and services (structure and mechanical layout, electrical and hydraulic service) through the join of the project components [DN10]. It is performed by the BIM technology,

overlapping the different data available in the infrastructure model. Furthermore, BIM allows sharing data between clients or users during the life-cycle of the building [WLK⁺13]. Hence, BIM allows an explicit and transverse information sharing among the stakeholders.

The Industry Foundation Classes (IFC) data model, developed by buildingSMART, was developed with the aim of presenting the building context and it is the BIM data exchange format [DCA16b]. IFC is object-oriented and is stated as a data modelling language EXPRESS [BKD⁺14]. In the version IFC4, some new steps were done to a better interoperability between BIM and GIS [LWW⁺17]. IFC models are classes that can describe the building elements and its details during the entire construction process [Isi15]. Indeed, each infrastructure element can be modelled and changed during its life-cycle. Thus, we can have access to all data about an element or one infrastructure at any stage of development. It also includes some different types of geometric representations such as Primitive instancing, Boundary representation (B-Rep), Constructive Solid Geometry (CSG) and Sweep volumes geometry [OBD⁺17]. Thus, an object in IFC classes can have different geometric representations [BKD⁺14]. Hence, BIM allows a vast set of different ways to represent one infrastructure.

The IFC aims to get all the information about the building. Some examples are the detailed geometry of the entire building and each of its components, semantic data such as costs, among others [DCA16b]. Thus, IFC collects detailed information about every element of one infrastructure.

BIM uses the concept of levels of development (LODt) instead of levels of detail (LOD) in GIS. Figure 2.1 shows an example of LODts. As indicated by W. Ikerd, D. Merrifield, J. Vandezande, *et al.* [IMV⁺13] LODt100 is the lowest level of development. A symbol or other generic representation can represent the building. In LODt200 the building is represented as a generic system, object, size, shape, and orientation, among others. The building's model can also have non-graphic data. LODt300 represents the building as a specific system, size, shape, and orientation, among others. Non-graphic data can also be added. LODt400 represents the building as a particular system, quantity, and direction with detailing, fabrication, assembly, and installation information, among others. The last LODt is LODt500, and it represents the building with a verified representation of all of its parameters. Moreover, can also be modelled a LODt350. This one is almost the LODt400 but lacks in some details.

Some improvements in the BIM models bring a new way to physical modelling a building. Thanks to that, BIM allows using the construction materials, subsystems, and usage behaviour when the user is building the new models. Furthermore, IoT can bring monitoring data to integrate with BIM models. With such integration, it is possible to have an energy consumption simulation in real-time. Moreover, BIM has the capability of simulating different optimisation scenarios in the design phase [BAU⁺17]. These optimisations bring development of smarter buildings and infrastructures, with concern about environmental issues.

The BIM's semantic models produce improvements in the interoperability between different programs and the capability of creating specific views to some specific businesses. A peculiarity of the BIM semantics is that it uses ontologies to deal with models. Thus, the ontologies connect

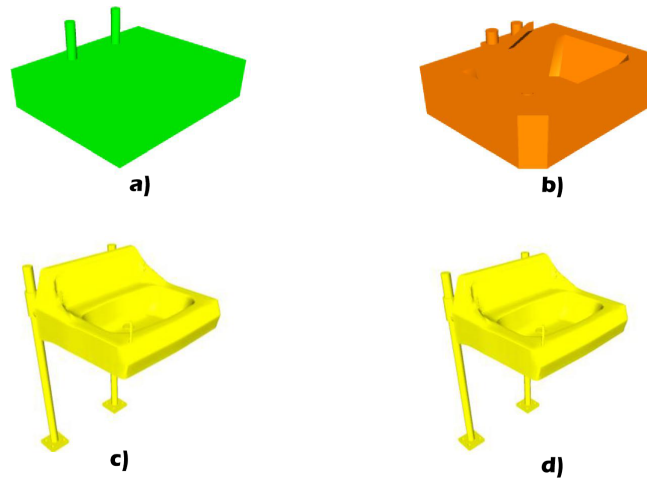


Figure 2.1: BIM's Levels of Development. The a) image corresponds to a LODt200, the b) image corresponds to the LODt300, the c) image corresponds to the LODt350 and the d) image corresponds to the LODt400. Images from W. Ikerd, D. Merrifield, J. Vandezande *et al.* [IMV⁺13]

the knowledge resulting of each stage of the facilities life-cycle. The ontology graph is deduced from the model built by the user [MN14]. Thus, its almost demanding the use of ontologies for integration with BIM. However, this isn't so linear as it looks. Section 4 will discuss this topic in more detail.

In its paper, Y. Arayici [Ara08] states that BIM can model existing infrastructures by modelling data captured by a 3D laser scanner, even though the initial goal of BIM is to model objects that already exist.

The two main limitations of BIM are the limited capability to handle the spatial data and the limited ability to locating objects in large-scale [TB09].

2.3 Geographic Information System (GIS)

GIS has all the functionalities that we found in an information system. Moreover, all data are geo-referenced [SC06]. Hence, the GIS coordinates correspond to the real world coordinates. Its spatial information uses a virtual or abstract model to represent reality. It also includes the spatial relationship between the building elements, the coordinates, and some non-spatial attributes. GIS allows 3D modelling and visualisation. Although in many cases the 2D it is enough, the 3D-GIS brings more detailed information and better visualisation. Normally, 2D-GIS uses the latitude and longitude to reference its objects [DCA16b].

CityGML is the standard language of the geographical domain data exchange [GP12]. Furthermore, CityGML is also the most used language in 3D modelling. It also represents a further step to interoperability with GIS [ARMN16b], allowing the insertion of new detailed information. CityGML is the most used language to exchange geographical information over the internet

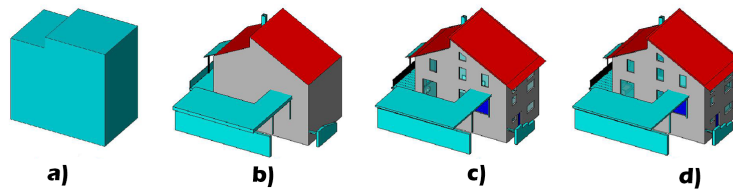


Figure 2.2: GIS's Levels of Detail. The a) image corresponds to a LOD1, the b) image corresponds to the LOD2, the c) image corresponds to the LOD3 and the d) image corresponds to the LOD4. Images from Filip Biljecki [Bil13]

[MN14] in the last years. Moreover, it can be shared among different applications and represents the objects in their geospatial context [GKNH12], with real-world coordinates.

Furthermore, GIS can handle large scales [MN14]. However, when it comes to the design of facilities the CityGML language isn't adapted yet, in contrast, the BIM's IFC language allows the detailed design of facilities. Different from IFC, the CityGML language is specified as Extensible Markup Language (XML). Nevertheless, it also has five levels of detail (LODs) [DCA16b], although, in BIM they are named as levels of development (LODt).

In CityGML the five levels of detail go from LOD0 to LOD4 [DCA16b]. Figure 2.2 shows an example of LODs. As indicated by C. Mignard and C. Nicolle [MN14], the LOD0 is a 2D or 2.5D map that represents the terrain. The ground surface does not appear in this LOD. This LOD is used to represent large areas like regions or landscapes. The LOD1 is used to represent city-scale regions and reveals the contours of the building. When we see details in some roofs and the textures, we are talking about LOD2 in CityGML. This LOD is more adapted to show city districts. LOD3 gives the 3D representation and all details in roofs and walls of the buildings. Moreover, this LOD adds urban objects and flora into the representation. Thus, this LOD can represent the buildings' exterior. The last LOD in CityGML is LOD4. This one models some interiors elements of the building in addition to LOD3.

Beyond all features detailed above, CityGML can also be used to get demographic information based on a map or the building geometry [DCA16b].

The GIS technology is capable of decision making, for instance, during floods when the workplace is too big, and its topology is too complicated for humans. Thus, GIS is a powerful technology that can also be used in the demolition planning, emergency responses simulation and others [Ban11]. Hence, with the GIS analysis, some disasters could be avoided.

2.4 Differences and similarities between BIM and GIS

As shown above, BIM and GIS have some similarities and some differences. It is imperative to know them if we want to make integration between the two technologies.

Systematic Literature Review

BIM uses 3D modelling centred in the buildings' design process. Its modelling relies on the geometric design, on the building's construction details, and on the project information [CDA15]. In contrast, GIS uses its 3D modelling with the aim to model the real world. Thus, GIS has its focus on the geographic data and buildings elements shape.

BIM aims to represent essentially objects that do not exist until the BIM's model creation. On the other hand, GIS aims to model the existing real objects [LWW⁺17].

BIM does not allow analysing the relationship between buildings' elements, or either to directly apply query functionalities to the model. In contrast, GIS can analyse, monitor and manage lots of spatial data and non-spatial data too [KI15]. The GIS tools provide the spatial query without any extra software. By opposite, BIM needs this kind of filters. Hence, they need each other to bring a full infrastructure management and maintenance.

BIM and GIS use component-based languages and, thanks to that, data related to a specific building element can be withdrawn separately [DCA16b]. It can be a good start point to achieve a functional integration between them.

Indeed, it is conclusive that their backgrounds are way different, as well as their application domains until now. This characteristic is one of the most significant barriers to their integration. Despite the differences between them, BIM and GIS domains have mutual needs of information from each other as we have seen above. GIS is older than BIM, and its scope is wider [MN14]. Thus, due to the different backgrounds, the information of each one is incompatible if we try to do a simple integration between them.

BIM uses different types of geometric representation (Primitive instancing, B-Rep, CSG and Sweep volumes geometry). GIS only uses the Boundary Representation (B-Rep). In this representation, GIS represents the objects as surfaces. Hence, this is a big problem when we are trying to integrate the two technologies. Until now, it is necessary that the BIM's objects are all in B-Rep geometry. Otherwise, the system developed needs to have a converter to the B-Rep geometry. Notwithstanding, these converters are difficult to implement without getting any errors and, in the majority of the cases, it is also hard to have a good performance and result.

X. Liu, X. Wang, G. Wright, *et al.* [LWW⁺17] summarised the differences between BIM and GIS as different users, different application domains, different levels of development of the technology, different scales, different coordinate system, and various geometric representations. Related to those differences, C. Mignard, and C. Nicolle, [MN14] stated that the principal problems of their integration are different organisations, different patterns, and different geometry models.

According to C. Tolmer, C. Castaing, Y. Diab *et al.* [TCDM13] the CityGML and IFC, even with an ideal integration, aren't enough to model the entire life-cycle of the infrastructures. Nevertheless, this integration brings many improvements and the IFC's, and CityGML's developers are always updating these two standards to overcome their missing features. Moreover, a full integration, or one with small losses of important data, is needed to leverage ourselves to the smart cities holistic concept and to explore the technologies and their potentials fully.

2.5 BIM and GIS Integration

Many works have been done to achieve integration between BIM and GIS technologies. Consequently, many different approaches were made to overcome the difficulties found in each work. During the research to this study, we have found many different methods or methodologies to reach the best possible solution. The most used methods are the unidirectional or bidirectional ones, and the methods at the semantic or geometric level.

The unidirectional mode only converts the data from one technology format to the other technology format, for instance, turning the BIM's data to a GIS supported data. The bidirectional method has the data converted to the two formats, allowing the use of the two technologies to create new data instead of only creating objects in one technology and then converting it to the other technology's supported format.

Furthermore, some of the works are using commercial software, and the other ones are using open source software, bringing more diversified approaches.

S. Amirebrahimi, A. Rajabifard, P. Mendis, *et al.* [ARMN16a] stated that the integration between the BIM and GIS technologies could be organised into three major layers or groups: (A) application, (B) data and (C) process levels. This study will also divide and analyse the BIM-GIS integrations based on this three-level grouping.

2.5.1 BIM-GIS at the Application Level

The BIM-GIS integration at application level does not involve any change in the data or an ontology. These integrations are only implemented in a specific case. Here, the BIM and GIS functions/analysis performed separately, and then the resulting information is integrated [ARMN16a, LWW⁺17].

In the metro construction, Q. Wang, P. Li, Y. Xiao *et al.* [WLXL14] proposed a BIM-GIS integration where GIS is used to planning route selection and analysis the spatial data, while BIM is used to model the infrastructures individually. BIM simulates the metro construction, the lights, the costs, and other dimensions. On the other hand, GIS does the management of the outside, based on the spatial data. Thanks to this integration, it is possible to manage better and planning the metro construction. For instance, GIS allows spatial analysis such as topography, geology, transportation, and economic ones. Furthermore, the whole integration brings the construction simulation, allows detecting conflict among professions, and can store the actual data of the monitoring. The conflict detection is advantageous due to the fact the metro construction has many different professions working together.

A system, aiming to introduce pre-retrofit models, that manage multiple forms of the building data was proposed by Ö. Göçer, Y. Hua and K. Göçer [GHG16]. The authors' goal in this study is creating a system to understand better and analyse a building that will be renovated, or even identify the need for doing that. Here, they used ArcGIS is connected to the BIM's models allowing direct queries through BIM technology. The results of this study showed that, with this integration, it is allowed more and better analysis of the infrastructures, better communication from

stakeholders side, and better data organisation. Thus, this integration improves the performance of the infrastructures' retrofitting.

J. Jia, Q. Zheng, H. Gao *et al.* [JJ17] proposed building models of ancient infrastructures using Revit software (BIM) to build the infrastructures component's models. The database of the infrastructures and its elements with their real-coordinates is in the GIS side. The principal objective of the authors is modelling the ancient infrastructures to protect their (almost) original states. This kind of infrastructures suffered many different destruction forms over the years, and they are disappearing too fast. With this BIM-GIS integration, it is possible to model the infrastructure and then protect and renovate it based on its models. Furthermore, it is also possible to evaluate the infrastructure evolution through the years.

S. Yamamura, L. Fan, and Y. Suzuki [YFS17] used the centre of Tokyo to test their GIS-BIM integration. They develop a system for urban energy planning with the aim of providing support for the smart cities planning. Their system uses GIS to get the information about the infrastructures and the BIM technology to gives the energy simulation for each infrastructure. After having the simulation of energy provided by BIM is returned this result to the GIS city model. The system allows the 3D visualisation either in GIS or BIM steps, where the user can see the energy consumption, simulation result, among others. As the authors said, this integration is more based on the GIS technology's analysis. A middle-ware integration would be more efficient and also could allow more analysis and queries to the data. However, the authors confirmed one better analysis of the energy consumption using their system, and consequently, can be reached a better urban energy planning.

C. Tsai, Y. Xiao, H. Chen, *et al.* [TXC⁺17] proposed a system where is reached the integration between BIM and GIS by using the Revit software as middle-ware. Thereby, the objects in ArchiCAD keep their BIM's information, and we also can use the GIS functions. They used the Revit software to convert the files from IFC format to the supported data by Skyline software. Thus, their system allows querying the spatial information and the information about a specific object through the GIS operating management interface.

2.5.2 BIM-GIS at the Data Level

The BIM-GIS integration at the data level involves the conversion of data formats, linking data, or the extent of the standard formats used in the technologies [ARMN16a].

J. Hwang, C. Hong and H. Choi [HHC13] developed a prototype being a base study on the interoperability between BIM and GIS, taking into account the developments of the future. The authors analysed the available BIM software until their article's date, such as the Revit and ArchiCAD software, and they created a system that provides visualisation of the buildings and the topography to overcome some of the missing functionalities in the nowadays available software. The prototype allows correction of the BIM's coordinates, conversion of the IFC's data, search for an attribute and its respective display and the visualisation in 3D. The system was built using the leading technologies available and then adapting them to the authors' goal.

Systematic Literature Review

In contrast with GIS, is not established the multi-scale modelling in BIM. A. Borrmann, T. H. Kolbe, A. Donaubaue *et al.* [BKD⁺14] proposed extensions to the IFC to incorporate multi-scale representations of shield tunnels. The data is converted to the CityGML format to visualise it through the GIS technology. Hence, it is demanding the LODt conversion to LOD. To accomplish that, the BIM geometric models are converted to the B-Rep representation, allowing their interpretation in GIS. However, in this conversion, complex models can conduct the system into misleading data. The authors identified the risk possibility of deviation introduction in the design phase by not simultaneous apply modifications in all LODs.

A method to convert the IFC entity model into the CityGML surface model was proposed by H. Liu, R. Shi, L. Zhu *et al.* [LSZJ14]. They analysed the IFC architecture and created algorithms to convert the geometry and the coordinates of the IFC model into the CityGML model. The system reads the IFC files, turns the geometric data, converts the coordinates, and generate the surface model of CityGML. Moreover, it allows the 3D visualisation. Thereby, we can have access to the indoors details even through the GIS queries, and the BIM's model also has the spatial information. However, this study does not convert all the BIM's possible geometric representations. Only the Sweet Solid and B-Rep representations were taking into account. Once again, the differences in the geometric representations between BIM and GIS technologies are a barrier to their full integration.

T. Kang and C. Hong [KH15] proposed software that integrates BIM and GIS by converting BIM's IFC to GIS's CityGML and algorithms for LODt to LOD conversion. The system aims an integration to perform a better facilities management. Thus, were developed the system and its components with the facilities management into account. Their proposed method semantically maps the BIM objects to GIS objects. However, is performed the LODt to LOD conversion by a different and separate algorithm. In the end, is read the object by GIS technology. Indeed, they developed a system where is reached the integration. Nevertheless, it loses critical data in their conversion. The conversion made relies on their use case study. For other studies, other converted information could be necessary. Thus, it is required to change the converters so that they can handle different data.

In their study, S. Amirebrahimi, A. Rajabifard, P. Mendis *et al.* [ARMN16a] proposed integration between the two technologies using a new data model for GIS. They converted the IFC's infrastructure information to a database based on their new GIS data model (XML). It is gathered the semantic and geometric information separately and in the end, is combined and stored. The aim of their system is a BIM-GIS integration to evaluate the flood damage in the infrastructures. This kind of integrations requires a lot of time analysing the IFC and CityGML architectures to build a new data model. Furthermore, this system has essential data loss. Indeed, their system is limited to one or two infrastructure analysis in simultaneous. Otherwise, the detailed analysis and the computer resources can crash. However, this system can perform functional analysis in the flood damage that some other methods cannot do.

A BIM-GIS integration to combine traffic noise evaluation in outdoor and indoor environments was proposed by Y. Deng, J. Cheng, and C. Anumba [DCA16a]. In their study, is represented the

environment in a 3D GIS model that also contains the BIM's detailed information. This study comes due to the lacking of noise mapping tools with enough flexibility to changes and different inputs support. The system consists of a bidirectional conversion between IFC and CityGML models. This system allows the geometric transformation of the Swept Solid and Constructive Solid Geometry representation from BIM. Thus, this integration covers practically all the geometric representations from BIM and converts them into GIS surfaces models. Thereby, using this system enables better urban planning taking into account the noise parameters in the infrastructures.

A Linked Data method to provide a semantical connection between BIM and GIS was proposed by S. Vilgertshofer, J. Amann, B. Willenborg *et al.* [VAW⁺17]. The data connection allows querying to both data models. Through this study, the authors demonstrate the possibility of semantically linking the two models to have better results and doing that without data loss. The common conversions, even if they are bidirectional, cannot be executed without data loss. By using link data method is possible to access the data lost in the conversion and have a full understanding of the infrastructures' information. Furthermore, the authors suggest a possible method improvement by implementing machine learning techniques to automate the mapping process.

K. Otori, F. Biljecki, A. Diakit , *et al.* [OBD⁺17] analysed the geometric and topological issues in a BIM and GIS integration. They had found that, at the data level, BIM has geometrical and topological errors, building a barrier to a full integration between the technologies. Due to the all different possible geometric representations on BIM, supports all of them in integration is also a big problem. Furthermore, the BIM's information about an element is way more detailed than in the GIS usually representation. Thus, according to the authors, it is not probable that will be converted all of the BIM's details to GIS. Instead, it is used a generalised version in the integration.

In their paper, G. Boyes, C. Ellul, and D. Irwin [BEI17] highlighted some of the BIM-GIS integration difficulties. They analysed how we can execute 3D spatial queries and how to use them to build a two-way link between GIS and BIM data. The main issue found in the integration of the two technologies is related to the conversion of the Constructive Solid Geometry geometric representation from BIM to the CityGML surfaces model. However, their study only focuses on the geometric conversion. Thereby, the semantic side of the integration was not studied. Therefore, some main issues related to the BIM-GIS were not analysed or considered by the authors, such as the related issues to detailed data conversion from the indoors of the infrastructures.

2.5.3 BIM-GIS at the Process Level

The BIM-GIS integration at process level involves the use of a reference ontology to store and represents the differences. Thus, the data format is not changed in this integration level [LWW⁺17].

In their study, C. Mignard and C. Nicolle [MN14] proposed a semantic extension to the BIM called Urban Information Modelling (UIM). The system bases is an ontology able to handle sources from the IFC and CityGML formats. After the import of the files (IFC or CityGML), the system stores the semantic and the geometric information in two different databases. There, are created the links between the two types of information so that for each object its geometry is associated. The visualisation can be from the 3D viewer or can be in contextual alphanumeric

Systematic Literature Review

trees form. Thereby, this approach brings the modelling of the buildings, of the building's environment, of the public aspects, and of the networks. Thus, their study allows modelling all the city's information.

E. Karan and J. Irizarry [KI15] proposed a semantic web technology system taking into account two different scenarios. Thus, the system aims the topography modelling in BIM technology and locating the BIM's infrastructures in the geographical environment, bringing a BIM-GIS integration to accomplish that. Some standard ontologies were also used, in their study, to perform the query process and the integration of the spatial data process. As authors refer, the semantic web technologies are still a novel technology and, consequently, it takes some time to reach the best way of using this kind of ontologies. An essential characteristic of this study to emphasise is the fact that the system can be applied not only to the pre-construction phase but also to the maintenance and repair processes.

An integration to eliminate significant amounts of efforts required to monitor and analyse the walkability for safe routes to schools was proposed by J. Kim, B. Koo, S. Suh *et al.* [KKSS16]. They use an ontology for BIM and GIS to get a formal representation of the walkability for safe routes. Their ontology's scope was limited according to the required infrastructures to evaluate the walkability. The GIS technology displays the visualisation of the walkability, and BIM performs the walkability evaluation. The case of study used to test the integration had good results, and it proved the BIM-GIS integration can add value to the walkability evaluation. Further developments in this topic can lead people to better walkability conditions, especially the ones with some problems to walk, such as wheelchair people, blind or with eyes problems people, among others.

Y. Deng, J. Cheng and C. Anumba [DCA16b] proposed a reference ontology to map the IFC to CityGML and vice-versa. The system maps CityGML or IFC to the reference ontology and, later, it maps the reference ontology to IFC or CityGML. In their study, they converted the IFC B-Rep and Sweet Solid representation into the CityGML's B-Rep representation. Their integration is one of the best ones found by us regarding interoperability ratio. They achieved a great result in integrating either semantic and geometric information. However, not all the geometric representations allowed in BIM technology can be converted in this integration. Furthermore, were not tested roads and tunnels, only the buildings. Moreover, again a BIM-GIS integration loses essential data.

As shown above, many different approaches were made to achieve the best solution for the integration between BIM and GIS technologies. However, they all lose important data and does not integrate all the available data. Thereby, it becomes clear the difficulty on performing a seamless integration between these two technologies. Apart from this, we can see the many possibilities created by this integration. It is possible to create systems helping the decision-making in the management and maintenance of our infrastructures integrating the BIM and GIS technologies, even now when their integration still loses some information.

2.6 Internet of Things (IoT)

Internet of Things (IoT) aims to provide communication between all sensors and devices, whether they are virtual or real things. The IoT also defines a global network connection with some communication protocols to exchange their information in real-time [Isi15].

The IoT sensors can have a dynamic location when they can move in a certain area, or they have a static location. Furthermore, the location and the specific time information should be used to an easier and better understanding of sensors information. Thus, to overcome the sensors and devices heterogeneity, the use of ontologies is needed [BAS⁺17].

The main challenges to the IoT concept in the construction context are finding fit sensors and devices with secure architectures [TWG⁺17]. Otherwise, people may not trust the system and, consequently, they will not use it.

To have real-time models of our buildings the integration of IoT with BIM is the key. Thanks to that integration we can access the building's element state, and building's spaces state data (rooms, halls, etc.) [Isi15], and do it in real-time.

Creating BIM models of our buildings brings the chance of designing rich energy modules. Moreover, these modules could be used for energy consumption simulation or some another forecast about the building's energy. If we connect the BIM's models with the IoT, we can monitor data such as temperature or energy consumption in real-time. Thus, the building's responsible can check the efficiency of the building regarding environment parameters, and do it in real-time too [BAU⁺17].

Cloud computing is an excellent helper for an IoT system when we are talking about processing the sensors' data [DM12]. Thus, we can use mobile devices to mount our system because the computing power is not on our device but in the cloud. As we know, the mobile devices do not have a powerful computation. Figure 2.3 shows an IoT cloud architecture example.

The smart buildings are a complex Cyber-Physical Systems (CPS). Thus, the buildings have its sensors and control system integrated with themselves. In the IoT context, smart buildings will be one of the most prominent users. They have connections between sensors, devices, and household appliances that, in its turn, communicate with some other apps. However, CPSs are vulnerable to environmental disorders, for instance, interferences and electromagnetic compatibility [Boy05]. Hence, advances in our sensors and CPS as a whole, are needed to overcome these faults.

Beyond that, the building's life-cycle is typically bigger than the operational time of the sensors' software companies. Hence, the software of the sensors becomes out-of-date and can bring many problems, especially in the security of the system. These faults can also lead to security problems in the assets of the building. Regarding all the security, usability and privacy aspects, the user must have access to detailed information about the possible consequences of choosing some option in the IoT devices or the apps [Boy05].

As shown above, the IoT paradigm demands new advances to be able to integrate with other technologies that depend on it. In a BIM-IoT integration, if some disturb happen the information captured can be misleading and conduct into, in the worst case, a disaster.

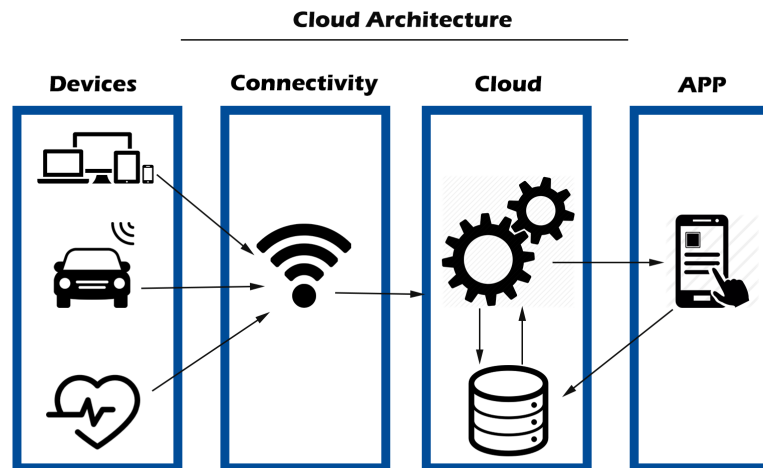


Figure 2.3: IoT cloud architecture. The IoT sensors present in the different devices communicate through a connectivity system like wireless and send data to the cloud. In the cloud, some algorithms translate the received information and communicate with the database and the user's app or service.

2.7 IoT, BIM and GIS Integration

Several studies have been done to integrate BIM and GIS with the IoT technology. In this section, some of these studies will be analysed and the main findings discussed.

U. Isikdag [Isi15] proposed an approach that uses IoT to gather the information about the infrastructure and then uses that information to build the BIM models; finally, the information is exchanged to create the GIS model. The IoT sensors, either virtual or real, provide geometric and semantic information about the infrastructures' and the surroundings state. The GIS API linked to the GIS technology, through messages exchanged with the IoT sensors, will be updated when any change happens in either a building's element or one from the city. Thereby, the visualisation in the system is through the GIS environment.

H. Boyes [Boy05] presented a study on the cybersecurity of the BIM and IoT integration. According to the author, the privacy and security problems have a forthcoming attention by the senior judges in the context of IoT and smart buildings. To achieve a holistic concept of the IoT's integration with our infrastructures the security, privacy, safety, utility, authenticity, and resilience paradigms should be satisfied and taken into account promptly. With the technological advances comes the security and privacy issues related to it. The improvements in the technology cannot bring benefits if these issues are not timely evaluated and overcome. Instead, many possible users will refuse the use of the new technologies.

In their study, C. Li, J. Hong, F. Xue *et al.* [LHX⁺16] proposed BIM-centered strategies to facilitate stakeholder communication and analyse the stakeholder's risk factor in the prefabrication housing production. The study aims to solve risks and enhance communication between

stakeholders. One of the strategies presented by the authors was a framework to map the offshore prefabrication processes allowing capturing the information flow through the offshore processes. Another plan is supporting the decision-making through the data analysis of the data flow diagram. The proposed BIM-centered strategy aims to integrate BIM technology and the IoT sensors with the people and the information flow. Thereby, the system can use real-time data and also help in smarter decision-making.

A GIS-IoT integration to analyse the environmental information in the city and then compare and associate the data with Chronic Obstructive Pulmonary Disease symptoms was proposed by L. Marek, M. Campbell, M. Epton *et al.* [MCE⁺16]. The IoT sensors are used to gather information about the user's movements in the environment, through GPS signal analysis. Moreover, these sensors also collect data about the air pollution, humidity, and temperature. The analysis of the data collected by IoT sensors integrated with the geographical information provide real-time monitoring of the user's life. Thereby, it is possible to better understand how this disease affects people in the city. As IoT sensors can be placed everywhere and smartphones have good GPS signals, the study can be scaled to big cities as well. However, some advances are demanded in the sensors interoperability due to their heterogeneity and resulting difficulty in use.

C. Hu, X. Cheng, X. Xiao *et al.* [HCX⁺17] proposed a GIS-IoT integration to manage the water conservancy facing flood and drought disasters. The system uses IoT sensors to gather data and store them. The type of data collected is information such as water level, rainfall or other meteorological data. Moreover, it is a cloud-based system, allowing computation to be performed in a cloud environment. Thereby, the system can be used in a mobile device with excellent performances. The system can also be used in a desktop PC and access the gathered data visualising it in a map. As the authors stated, the results support the fact that the GIS-IoT integration can bring a smarter way of managing the water levels and its conservancy.

S. Howell, Shaun, Y. Rezgui and T. Beach [HRB17] proposed an ontology aiming to support intelligent water sensing. The system also provides analyses for the optimisation of the water network. The ontology relies on information from BIM and GIS and extends the BIM's Industry Foundation Classes (IFC) with operational concepts. Moreover, it also integrates static data and near real-time data as well. Thereby, as the authors stated, their study encourages the use of the IoT and the semantic web approaches to creating real-time updated models. Hence, the water analysis and the management of the water networks can be monitored more smartly, also having the real-time information in the system modules.

L. Bottaccioli, A. Aliberti, F. Ugliotti *et al.* [BAU⁺17] proposed a BIM-GIS-IoT integration capable of allowing the real-time visualisation of the energy consumption of one infrastructure and the evaluation of the infrastructure performance also in simulations and considering real weather conditions. To overcome the interoperability issues in the IoT sensors, due to their heterogeneity, the authors developed a middleware web services approach. The system is cloud-based software that monitors and models the infrastructure. The IoT sensors and the BIM and GIS databases are integrated. Thus, the information gathered by the IoT sensors can also be used to measure the environmental and energy parameters of the infrastructures. This kind of integration allows

Systematic Literature Review

the infrastructure manager to execute near real-time actions or analyses of energy consumption, temperature, efficiency performance, among others.

R. Baracho, M. Pereira, and M. Almeida [BPA17] analysed the association of ontologies with IoT and BIM in construction. The authors refer that there are a few studies related to the integration of these three technologies. They also state that an ontology could represent the objects and its relationships according to the IoT perspective. The IoT sensors gather and communicate the environment information and the BIM technology allows the 3D infrastructure's modelling using the IoT database. They summarise their research stating that the integration of BIM, IoT and ontologies can bring improvements and novelties in the construction area.

In their study, S. Peng, G. Su, J. Chen *et al.* [PSCD17] propose the use of IoT for real-time monitoring, the use of GIS to provide the spatial information of the hospitals, and the use of BIM to enable a complete digital modelling of hospital buildings. The system aims to detect and analyse the risk at the right time and deal with emergencies in the hospital operation. The BIM and GIS technologies connect to the data collected by the IoT sensors. GIS provides the spatial data and a set of tools for simulation, while BIM builds the hospital model, with all the detailed information about its indoors. Thus, with this integration, it is possible to prioritise protection areas relying on the system's data.

A BIM-IoT integration aiming to analyse the seniors' quality of life was proposed by E. Roh and S. Park [RP17]. Their system measures the status of people's age, using biomarker sensors connected to the IoT and BIM technologies. The BIM data model stores the information gathered by the IoT sensors scattered through the nursing home. Indeed, this method can replace the general quality of life survey used until now. Moreover, it makes this measurement smarter. However, this study focuses only on the indoors analysis in the quality of life evaluation context. Possible integration with GIS could bring the analyses of the outdoors and it completes the measurement method.

R. Nourjou and M. Hashemipour [NH17] proposed a GIS-IoT integration to provide a web map with the information of power outage and correspondent economic loss estimation. Their system provides a tool for decision-making support in cases of disasters or other problems in the energy sector, mapping the energy network and analysing the best way of solving the problems with the minimum economic impact. Moreover, this analysis is in real-time by using IoT sensors. The system uses cloud computing to embed the other system components, such as the Web Feature Service, an autonomous computer program, the spatial database, and the web map service, and to make the system available everywhere.

In their study, J. Teizer, M. Wolf, O. Golovina *et al.* [TWG⁺17] propose a BIM-IoT integration for operation management inside of the infrastructures. During construction, the system provides real-time updated data. Firstly, the infrastructure is planned, since the field conditions are displayed, and then the actual conditions are retrieved. To accomplish that, the system is based on a cloud platform. Furthermore, the system also allows obtaining the localisation of the user through a Bluetooth Low Energy (BLE) simple location tracking developed by the authors. Thus, the system provides a smarter way of construction planning and execution with almost injury-free

construction management.

W. Almobaideen, R. Krayshan, M. Allan *et al.* [AKAS17] propose a GIS-IoT integration to calculate the path with more available medical centres. The study aims to help the tourists choosing their way according to the closest medical centres open in the city. The path distance may not be the shortest one. However, it will have the highest number of medical centres from the starting point to the user's destination point. This study is geared towards people with health problems. However, any person could use it. The system has the user health information gathered through the data obtained by IoT sensors, such as blood pressure, and the GPS signal of the mobile device. This information is used to monitor and calculate the path with the medical centres specialised in treating the user's illness. This study presents many improvements in people's healthcare, especially when they are outside their city and do not know the city they are visiting.

A BIM-IoT integration was proposed by R. Zhong, Y. Peng, F. Xue *et al.* [ZPX⁺17] to connect and share all the information with the stakeholders and also to add more dimensions to the BIM models such as cost estimation and duration schedule. The system relies on the prefabricated construction and gives the desired visibility and traceability to better monitor the status, the progress and the estimated cost of the project in real-time. The IoT sensors are used to gather information about the construction progresses, resource consumption and worker assignments. Thereby, the stakeholders have real-time information about the project status, and they can make decisions according to the project progress.

As shown in this section, the IoT integration can bring many improvements in the management and maintenance of our infrastructures. However, this technology is now taking the first steps, and we have many concerns about its use. The heterogeneity of the IoT sensors brings problems in the applications development side and also in the trustfulness side. Moreover, the privacy issues and also the security ones are some of the biggest concerns about IoT technology. If for some reason the IoT sensors or even the IoT layer store any wrong information about the infrastructures, its monitoring could conduct into misleading decision-making and, in the worst case scenario, could conduct into a disaster or a catastrophe.

2.8 Virtual and Augmented Reality (VR/AR)

VR introduce a user into a completely virtual and computer-generated environment. Nowadays, VR is becoming increasingly popular in the computer games context. Some interfaces can be transitional and be designed to take the combined advantages of VR and AR [SH17].

AR brings the possibility of augmenting our vision, or even images of real scenes, with additional information in real time [LMS⁺17]. That information can be from different kinds of sources, for instance, and in the GIS context, we can add spatial information to the scene and, thanks to that, we can make the planning process in the real place [BBR⁺17]. In a perfect AR system, the user's perception must be that the virtual object appears to become part of the real world. [SH17]. However, an AR system needs accurate locations from its sensors. Otherwise, the user can easily

induct itself into wrong information [BBR⁺17]. The human eye can easily find inconsistencies in the displayed image. If the virtual object isn't in the correct position, or without shadows, or even without a good illumination correspondence, the user will automatically find it and will not trust the system. Each object in the real world has some inherent amount of information, and sometimes the user cannot understand or view that information. An AR system aims to make easier the user access to that information, adding some virtual objects to the real scene [PME⁺17].

Many people think that the overlap between virtual and real elements are the special effects they see in movies. Indeed, the computer graphics techniques practised in movies could be applied to AR systems. However, movies do not have a special requirement of an AR system, the interactivity [SH17].

According to Azuma 1997 survey paper [Azu97], an AR system should have these three components: (1) The system must combine real and virtual, (2) should be interactive in real time, and (3) must be registered in 3D. If some of these are not present in the system, the system, as an AR system, cannot be recognised.

To an AR system, use of marks is needed to recognise the scene or specific objects in the scene. The identification of marks can be made by a marker-based method. Thereby, the user can choose specific marks that will be recognised by the AR system. However, in a large environment, these marks must be enormous and, consequently, this method is not optimal. On the other hand, it is possible to identify some marks in the scene, characterised as being natural marks. Here, the user does not specify a mark different from the environment. The system will analyse the scene and recognise some natural elements with the use of computer vision. This one is a complicated method and usually needs high computational power. Hence, as a mobile application runs with limited computational power, it is difficult to use the natural marks method in the mobile devices [WRM⁺10].

The technological components used in an AR system are the media representation, the interaction device, the feedback display, the trackers, and the computer unit [WLK⁺13]. Figure 2.4 shows an AR system and some of the used components. However, more components could be used as haptic devices and subsystems.

The AR environments were developed and applied firstly to scientific visualisation and the world of games. The technology advances made possible the advances of AR from the marker-based to markerless methods. Hence, AR could be applied to mobile environments. However, some factors do not allow the fast development of the AR systems. The main problem is the need of trackers and sensors of long-range that also has to be accurate in their measurements [WLK⁺13]. Otherwise, the system will display the virtual objects in a wrong position or with its illumination miscalculated.

The evolution of the smart buildings brings new applications to the AR systems. In this case, the AR system aims to the simpler management of the building and of its elements [PME⁺17].

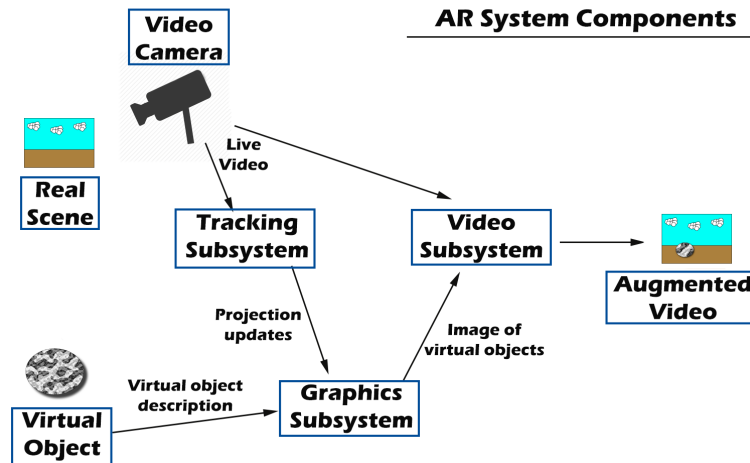


Figure 2.4: AR System Components. The real scene is captured by the video camera and the AR subsystems use this video for tracking purposes, or other ones. After having enough real scene's information the virtual object is overlapped, in the correct place, in the final display.

2.8.1 Computer Vision

The Computer Vision is a key functionality in an Augmented Reality system. The AR system needs to isolate an object among others in the scene. We can intuitively conclude that the recognition of an object in an image or scene is a challenging work to do. As we saw above, the system can use marks or a markerless methodology to recognise the objects. Thereby, in this phase of an AR system, the computer vision algorithms are critical to complete the recognition task.

In the AR area, the computer vision uses the electrical recognition of images provided by sensors, such as cameras, giving information about the users and the surroundings to the AR system [SH17]. However, AR demands real-time approaches, therefore, the computer vision algorithms must allow that, performing in real-time too.

The main purpose of the computer vision is the determination of the pose of some object relative to the camera and in the real world [SH17]. That pose estimation allows for building a virtual camera with the same parameters of the real camera. Therefore, we can align the virtual objects correctly to the real scene and also identify some object that we want. The pose estimation gives us the base to use other algorithms to determine patterns or feature points in the image captured by the camera.

Many algorithms were developed to the object recognition and pose estimation phases. Although they have very good results, some of them are computationally too heavy to be used in the mobile devices. The best example of that is the Scale-Invariant Feature Transform (SIFT) algorithm. It overcomes the principal problems in the object recognition, such as the different object scales. However, the algorithm requires too many computational resources to a mobile device.

2.8.2 Computer Vision Algorithms in Mobile Devices

To overcome the problem of the lower computational power of the mobile systems, we can use computer vision algorithms less computationally heavy. However, other problems arise here. With the use of computationally lighter algorithms, the system will not be capable of resist to some problems in the image recognition. These problems can be related to the different scales of an object, different perspectives, different illuminations, among others. Thereby, the trade-off must be studied to analyse which is the best algorithm to use relying on the system's goal.

E. Rosten and T. Drummond [RD06] proposed the Features from Accelerated Segment Test (FAST) algorithm. This algorithm is especially suitable to use in mobile devices due to its fast performance. However, the algorithm has some issues related to the weak resistance to noisy images or blurry ones.

D. Wagner, G. Reitmayr, A. Mulloni *et al.* [WRM⁺08] presented the PhonySIFT algorithm, a variant of SIFT optimised for mobile devices. This algorithm uses the FAST algorithm applied to a scale estimation of the object. However, here the scale estimation is not a convolution but a pyramid. Furthermore, the authors also present a modification to the Ferns algorithm.

A. Armagan, M. Hirzer, P. Roth *et al.* [AHRL17] proposed the use Convolutional Neural Networks (CNN) to a better pose estimation. A Convolutional Neural Network can be used in many applications, especially on works with images. CNN's applications allow for image classification, image semantic segmentation, object detection in images, among others [Wu17]. In their work, the initial pose estimation can be far away from the best pose estimation. However, in each iteration, the CNN is updating the pose estimation, and it is getting closer and closer to the best estimation. Other developed work in the CNN's area was developed by P. Wang, Q. Hu, Z. Fang *et al.* [WHF⁺17] aiming to the search of complex images in mobile devices.

A cloud-based system for mobile devices to recognise specific marks in the image was proposed by H. Arshad, L. Chun, W. Obeidy *et al.* [ACoy17]. In their study, they tested two algorithms for the recognition phase, the Binary Robust Invariant Scalable Keypoints (BRISK) and Oriented FAST and Rotated BRIEF (ORB) algorithms. The results from the two algorithms were compared to find which is the best one to use. According to the authors, the findings suggest that the BRISK is the algorithm with the best results. The developed study achieved a robust image recognition capable of detecting the marks in noise-free images as well as in dark and partially occluded ones.

E. Charalampaki and A. Malamos [CM17] proposed a methodology for tracking and detection of objects in complex scenes using the mobile device's camera. In their work, they used the Multiple Instance Learning (MIL), Track Learn Detect (TLD), Median Flow (MF), and Kernelized Correlation Filter (KCF) algorithms. They reach great results, but the test scene was a 5*5 football field, testing the recognition of the players. Thereby, as the authors said, to even more complex scenes, it is necessary to overcome some barriers, especially the occlusion-related ones.

Some projects use the Kinect Fusion algorithm to estimate the pose of the camera and then to track it after the recognition. This algorithm uses a Kinect camera, and it iteratively constructs the

scene model. One of these works is the study carried out by L. Nardi, B. Bodin, S. Saeedi *et al.* [NBS⁺17]. Their study aims to reduce the execution time, however, without compromising the results.

G. Pintore, F. Ganovelli, R. Scopigno *et al.* [PGSG17] proposed different methodologies to the reconstruction of the buildings' indoors. Moreover, their approaches use the mobile devices to capture the images. In their studies, they used the Shi-Tomasi Corner Detector algorithm to detect the feature points. After having enough set of feature points, the homography is calculated to estimate the camera's pose.

A system for mobile devices that integrates the GIS and Computer Vision technologies was proposed by M. Serrão, S. Shahrabadi, M. Moreno *et al.* [SSM⁺15]. In their work, they used the Speeded-Up Robust Features (SURF) algorithm to detect the feature points of the scene. The authors pointed out that SURF is not computationally heavy. However, must be confined the space of search in the image to areas of interest instead of the using the whole image. The system could be used on mobile devices. However, tests were carried out on a laptop. Nevertheless, the tests presented good results, and the system works in real-time.

B. Sung and C. Lin [SL17] proposed a system to reconstruct the scene in 3D using a mobile devices' camera instead of laser scanners. To acquire the features points they tested several algorithms such as the SIFT, SURF, ORB, and, the new one, the Accelerated KAZE Features (A-KAZE) algorithm. The results of each algorithm were compared according to the best system performance. However, the results were not good enough. Thereby, the authors tested the system using the FAST and the pyramidal Lucas-Kanade (L-K) algorithms getting here good performance.

Simultaneous Localization and Mapping (SLAM) has been mostly used in the robot localisation field. However, the SLAM techniques are now utilised to obtain the same accuracy and real-time performance in the AR area [GWT⁺17]. SLAM is a technique able to build and generate a map of the environment [KTH16]. Thereby, using the generated map, we can determine the user location, in the AR field. SLAM uses odometry-based methods giving an estimation of motion. Furthermore, this technique reduces the positioning error along the navigation path of the user. As the optical systems provide information with high quality, and by now at a moderate cost, the development of SLAM approaches have been a search target all over the world [BNPM18].

As we saw above, there is any standard computer vision algorithm to use in the mobile devices, or even in the computer systems. Many algorithms were presented here such as SIFT, SURF, FAST, ORB, BRISK, A-KAZE and SLAM. However, none of them has consensual results. Thereby, the best computer vision algorithm is the one that gives us the best performance in a trade-off between the feature matching and the real-time execution. Moreover, the mixing of more than one algorithm, or even the creation of new ones, can be the best solution depending on the system needs.

2.9 AR/VR, IoT, BIM and GIS Integration

Several studies have been done to integrate BIM, GIS and IoT with AR/VR technologies, only using the maximum of three of them in each project. In this section, some of these studies will be analysed and discussed.

A BIM-AR integration was proposed by X. Wang, P. Love, M. Kim, *et al.* [WLK⁺13]. The system aims to enable the visualisation of the hidden information about each construction element or activity through the use of an AR system embedded. Thus, the user can have access to the data in real-time and also give feedback to the BIM's model, updating it. The authors point out six areas where this integration can bring some improvements: (1) interdependence, where usually each worker thinks only on its tasks instead of regarding the other functions; (2) Spatial site layout collision management, where some conflicts can appear due to the workplace's lousy placement; (3) link digital to physical, where the design and planning tasks work essentially with information; (4) project control, where the schedule can grow due to design changes or errors; (5) material tracking and management, where the delivery planning of the material could have problems; and (6) visualization of design during production, where the interface could be challenging to use.

Z. Lv, T. Yin, H. Song *et al.* [LYSC16] proposed a GIS-IoT-WebVR integration aiming to analyse the city and also have a visualisation of the city in a virtual environment. The system can also perform the spatial analysis and the visualisation of the spatial information process in 3D. Furthermore, it provides a consistent external access interface for GIS and VR with access to each model. The ability of monitoring and visualising the status of the city in real-time brings a smarter way to control our cities. The authors developed this study aiming to improve the environmental protection and awareness, the public safety, the management and the maintenance of the infrastructure and through that, they want to make the world a greener place.

M. Breunig, A. Borrmann, E. Rank, *et al.* [BBR⁺17] proposed the development of a collaborative platform and 3D multi-scale modelling to improve the digital 3D planning of the infrastructures through the BIM-GIS integration. They tested the conversion of the CSG geometric representation in their study and designed a universal access module and also a query module to access their different modules of the representations. Furthermore, they also developed a mobile AR system to overlay the model with the information related to it in real-time. This AR embedded system uses image-based localisation and mapping to estimate the pose of the camera and user. Therefore, this kind of systems can help the subway track infrastructures construction providing specialised and specific information about an infrastructure's element in all the phases of the infrastructure life-cycle. Moreover, this approach can be applied not only to the subway infrastructures but all of them.

In their study, E. Patti, A. Mollame, D. Erba *et al.* [PME⁺17] proposed and BIM-IoT-AR integration. Their system is an Android-based application that enables navigating in an AR environment and accesses the BIM-IoT integrated data. There were used QR codes as marks to the AR recognise system, allowing a fast recognise of which infrastructure appears in the scene. They also created a middleware layer to overcome the IoT sensors interoperability issues. Thereby,

it is possible to access the environmental parameters such as temperature and energy consumption and integrate them with the BIM's models. Through this integration, the users can leave the desktop and have local access to the infrastructures elements information by the AR environment. Moreover, the user has a better perception of the environmental parameters of the infrastructure.

G. Luchetti, A. Mancini, M. Sturari *et al.* [LMS⁺17] proposed a GIS-IoT-AR integration to decision-making about unexpected events such as natural disasters. The preventing and the monitoring of the natural accidents are one of the most important performances in the human lives saving context. Their system collects data from the social networks and also analyse it. Important to emphasise that the gathered data is geo-referenced and for their study the social network tested was Twitter. The mobile application allows the visualisation of the information in an AR environment. Thereby, it is possible to get information about an event just pointing the mobile device camera to it. Furthermore, the system also shows analysis of the events either in text or using maps.

As shown above, the AR/VR integration brings many possibilities to smarter management and maintenance of our infrastructures. By displaying the hidden inherent information of the infrastructure's element, it is possible to understand the user tasks better and even to give better feedback to the system. An example of this kind of hidden information is the cost of an element. With the mobile approaches, it is possible to retrieve the information "in loco" and, thanks to the IoT integration, the displayed information is updated in real-time.

2.10 Summary

Many approaches tried the BIM-GIS integration, others BIM-GIS-IoT integration, and others BIM-IoT-AR/VR or GIS-IoT-AR/VR integration. This integration allows for new kinds of data analysis, data collection as well as using them in real-time. Thus, it is possible to monitor all steps of the infrastructure management and maintenance in a smarter way. Moreover, it brings important advances to have cities with more sustainability, less energy consumption, infrastructures crash prediction, and so on, which also leads to better global sustainability.

After analysing all studies, we can select the best ones regarding the road networks infrastructures management and maintenance. In this context, the most relevant are the ones related to the infrastructures' monitoring by providing the internal and external parameters to a smarter use. Table 2.3 summarises the best ones identifying the technologies used and also showing the relationship between the technologies used and the development year of the integration. Over the years, we have a shift to the integration of the four technologies. However, that only will be achieved in the next few years.

Table 2.1 and Table 2.2 presents a summary of the features developed in the analyzed studies. Moreover, these tables present the features in a broad way, where one of them can correspond to one or more features aggregated. Table 2.1 presents the features ID and its description while Table 2.2 presents the category of each feature.

Table 2.1: Description of the Developed Features using the Discussed Technologies.

Feature ID	Feature Name	Feature Description
F01	Conflict Detection among Professions	The construction simulation allows the detection of conflict among the different professions and can store the actual data of the monitoring. Conflict detection can be very helpful in construction, where many professionals from different areas are working together.
F02	Urban Energy Planning	It uses GIS to get the information about the infrastructures and BIM technology to give the energy simulation for each infrastructure. The feature allows the 3D visualization either in GIS or BIM steps, where the user can see the energy consumption, simulation result, among others. It also uses IoT sensors to update the GIS and BIM data.
F03	Searching for an Attribute and its respective Visualization in 3D	Querying for a specific element or attribute to the system returning the BIM and GIS data.
F04	Multi-scale representations of Shield Tunnels	Extensions for the IFC to incorporate multi-scale representations of shield tunnels, not supported by BIM technology in its releases.
F05	Flood Damage Analysis	Gathering GIS, BIM, and IoT data for a better analysis of the flood damage in one infrastructure.
F06	Walkability Evaluation	Significantly reduce the required efforts to monitor and analyze the walkability for safe routes. GIS and BIM data allows for a better evaluation in this scenario.
F07	Life Quality evaluation	The use of BIM, GIS and IoT data to get the environmental parameters such as air quality, humidity and temperature.
F08	Intelligent Water Sensing	It allows for analysis for the optimization of the water network.
F09	Visualization of Hidden Information	It enables the visualization of hidden information about each construction element or activity through the use of an AR system.
F10	Construction Progress Information	It gathers information about the construction progress, the material consumption and the worker assignments. The stakeholders have real-time information about the project status and they can perform decision-making according to the project progress.

Table 2.2: Features Categorization. "X" means they belong to that category.

ID	Category									
	Data Type		Structured Data		Framework					Energy Consumption
	Static	Dynamic	Yes	No	Construction Maintenance	Road Management	Water Levels Management			
F01	X		X		X		X			
F02		X	X						X	
F03	X		X		X		X			
F04	X		X		X		X			
F05		X	X		X		X		X	
F06	X		X				X			
F07		X	X		X		X			
F08		X	X		X		X		X	
F09		X	X		X		X			
F10		X	X		X		X			

Systematic Literature Review

Table 2.3: The most successful integrations with the highest relevance in the infrastructures management and maintenance with the possibility of being applied to the road networks infrastructures management and maintenance. "X" means they used that technology.

Related Work	Year	BIM	GIS	IoT	AR/VR
C. Mignard and C. Nicolle [MN14]	2014	X	X		
A. Borrmann, T. H. Kolbe, A. Donaubauer <i>et al.</i> [BKD ⁺ 14]	2014	X	X		
Y. Deng, J. Cheng, and C. Anumba [DCA16a]	2016	X	X		
Y. Deng, J. Cheng and C. Anumba [DCA16b]	2016	X	X		
L. Marek, M. Campbell, M. Epton <i>et al.</i> [MCE ⁺ 16]	2016	X	X		
S. Yamamura, L. Fan, and Y. Suzuki [YFS17]	2017	X	X		
L. Bottaccioli, A. Aliberti, F. Ugliotti <i>et al.</i> [BAU ⁺ 17]	2017	X	X	X	
R. Zhong, Y. Peng, F. Xue <i>et al.</i> [ZPX ⁺ 17]	2017	X	X	X	
M. Breunig, A. Borrmann, E. Rank, <i>et al.</i> [BBR ⁺ 17]	2017	X	X		X
E. Patti, A. Mollame, D. Erba <i>et al.</i> [PME ⁺ 17]	2017	X		X	X
G. Luchetti, A. Mancini, M. Sturari <i>et al.</i> [LMS ⁺ 17]	2017		X	X	X

The evolution of our cities into the concept of smart cities is now starting. We see the first IoT sensors across the city to better monitoring and management as well. However, we are far away from reaching the smart cities holistic concept, where the city is fully automated, and its monitoring, management, and maintenance are based on sensors across all of the infrastructures in the city.

With the smart cities comes the smart buildings and also the smart infrastructures. These improve their management and maintenance through the use of the sensors as well, resulting in greener infrastructures. Furthermore, the infrastructures' reading can also avoid disasters in many different cases, either regarding the infrastructure's indoors or the surrounding environment.

To achieve the smart cities goals we need to improve our technologies. Some works and studies with the same goal were analysed in this chapter. It has been shown the different approaches to overcome the lack of the smart infrastructures mainly relying on the BIM and GIS integration. It becomes clear that this integration is challenging to do, especially if we want to do a seamless integration. Some different studies tried to overcome the main challenges, but no one reaches an integration without losing important information. Unidirectional and bidirectional methodologies were used and also geometric and semantic ones. These methodologies were applied at the application level, at the data level, and at the process level. However, was not achieved a seamless integration between the BIM and GIS technologies. Indeed, the seamless BIM-GIS integration is hard to achieve because the two technologies are too different from each other. They have different backgrounds, a different level of development and a different coordinate system. Furthermore, the users of each one are different and want different outputs from the system. Also, they have a different geometric representation which is a big barrier to their integration.

Systematic Literature Review

Other studies tried to implement integration between BIM, GIS, and IoT. This integration has the problems of the BIM-GIS integration and also the ones related to the IoT technology. The IoT sensors are now starting to appear in our lives and cities. However, they come from different industries bringing a heterogeneity of sensors. It means that there is not a standard to build the sensors regarding the way the sensors communicate with each other. Thereby, if we want to use them, we need to create a middleware layer to overcome the communication heterogeneity issues. Another big concern in the IoT paradigm is the privacy and security issues. IoT is a novelty, and the users do not trust it until the system provides the security and the privacy that they want. Notice that, if some IoT sensor was hacked or if it has some problem, the obtained information to the monitoring system could be wrong. It is necessary to inform the user about the IoT problems and overcome these security and privacy problems as fast as possible to leverage the IoT applications in our world. As we seen in the previous studies, the IoT information can bring improvements in the management and maintenance of our infrastructures. Although they can be used for other purposes, these improvements are especially linked with the environmental parameters.

We also analysed some studies aiming at the integration of BIM, GIS, IoT with the AR/VR technology. However, was never made integration between the four technologies. The most common one is the integration using three of the four technologies. By using the AR/VR integration, we can display the hidden inherent information of the infrastructure's element. It can bring a huge improvement in the construction paradigm. Furthermore, using the mobile devices to interact with the AR/VR system the information can be accessed at the infrastructure element's site bringing a fast and better interaction between the user, the system, and the element itself. Moreover, the interaction is in real-time, and the user can see the information overlapped over the infrastructure's element. Indeed, the AR/VR technology brings improvements in the integration. However, we need to improve the AR/VR systems. The tracking and the localisation of the user still be a challenge to overcome in the AR/VR technology. We need better sensors and techniques to reach a perfect system.

There is so much to discover and improve regarding the management and maintenance of our infrastructures. Several works have been done to accomplish that and, in the next few years, more will appear. The management and maintenance of our infrastructures is a new study field, and the smart cities are emerging in our lives. Thus, this is a research way where many developments will be reached. Towards that, those responsible for BIM and GIS are trying to make the interoperability of their technologies more feasible.

We can predict a massive improvement in the management and maintenance of our infrastructures by integrating the four analysed technologies. With a BIM-GIS integration based on the IoT's information, we can have access to real-time information about the indoors and the surrounding environment of our infrastructures. Moreover, using the AR/VR technology, we can see and interact in real-time with the infrastructures parameters and also do it in the infrastructure's site by using the mobile devices applications. Thus, this integration brings an enormous amount of possibilities to our purpose, leveraging us to the smart cities holistic concept. Moreover, any studies towards the integration of the four technologies are steps to a greener and more intelligent

Systematic Literature Review

world and to prevent disasters in our cities.

Chapter 3

Methodological Approach

3.1 Problem Statement

The management and maintenance of road infrastructures require the use of a large amount of information about the maintenance history and the current situation of the infrastructures. This information could be about repair work, the company who makes some element of the infrastructure or even the infrastructure's project. However, nowadays the access to the information is difficult because almost all the information is stored on paper. It is demanding to make smarter the management and maintenance of the road infrastructures. More intelligent monitoring allows accessing the data efficiently and also to predict changes in due time.

The construction and the management of the road infrastructures have many and different stakeholders involved. Furthermore, sharing the information among them could be a big puzzle. Moreover, any new event in the infrastructure is not automatically communicated. We need the human intervention to communicate all the issues related to the management and maintenance of the infrastructures. Thus, it is necessary to build smart systems capable of automating this work.

Some technologies have been developed in the construction area to get 3D models and the specific information about our infrastructures such as energy consumption, costs, among others. In the last few years, the standard technologies used in construction are the BIM and the GIS. As we saw in the Chapter 2, the BIM technology lacks in modelling the surroundings and the GIS technology lacks in modelling the indoors of the infrastructures. Thereby, these two technologies must be integrated to give us a tool to access all the available information about one infrastructure. However, the integration between the two technologies is still a big puzzle for the developers and researchers. The best works towards that integration always fail at something that leads to the loss of essential data. Analysing the biggest problems in their integration, we can point out the interoperability issues resulting from the different technologies' backgrounds and the different ways of representing the data.

The IoT and the VR/AR technologies can also be integrated with BIM and GIS to have the data updated in real-time and also to a better visualisation of the data. One full integration of the four technologies is a powerful tool for the management and maintenance of the road infrastructures.

3.2 Proposed Solution

To overcome the presented issues related to the management and maintenance of the road networks is proposed a system capable of integrating the four technologies mentioned. The system is based on the VR/AR technology to give the user a quick and easy way of getting information about the infrastructures and also doing it in real-time. As concept proof, two applications for mobile devices were developed. The first one, named *Augmented Reality for Management of Road Networks - Place Signage at Location* (ARIMaRN-PSAL), allows the user to place virtual signage in somewhere to better evaluate what is the best position for it. The second one (ARIMaRN) uses the AR to display the information about any infrastructure in real-time, allowing the user to aim its own mobile device's camera to the infrastructure getting the information about it. As we can see, the second app integrates the four technologies and can represent a significant change in the management of the road networks.

3.3 ARIMaRN-PSAL

The application makes available a traffic signal model to put at one location and then evaluate if it is its best position. When the user touches the mobile device's display, the model appears on it, and the user can study the signal position.

3.3.1 Ground Detection

The app must recognise the ground so that the traffic signal model can be placed in the right position according to the user wish. After the ground being identified by the ARIMaRN-PSAL app, it estimates the pose of the mobile device camera and, according to that, the app builds the model in the centre of the identified ground, vertically aligned.

3.3.2 Traffic Signal

The traffic signal model will have the realistic size of a real signal. Converting the scene unities into the real meters is necessary. When the user touches the mobile device's display, the signal model is placed in the middle of the recognised surface as a ground. Afterwards, the user can go around the signal and saw it from different angles and distances. Therefore, the model needs to be tracked in every frame to give the user the impression that the signal does not move.

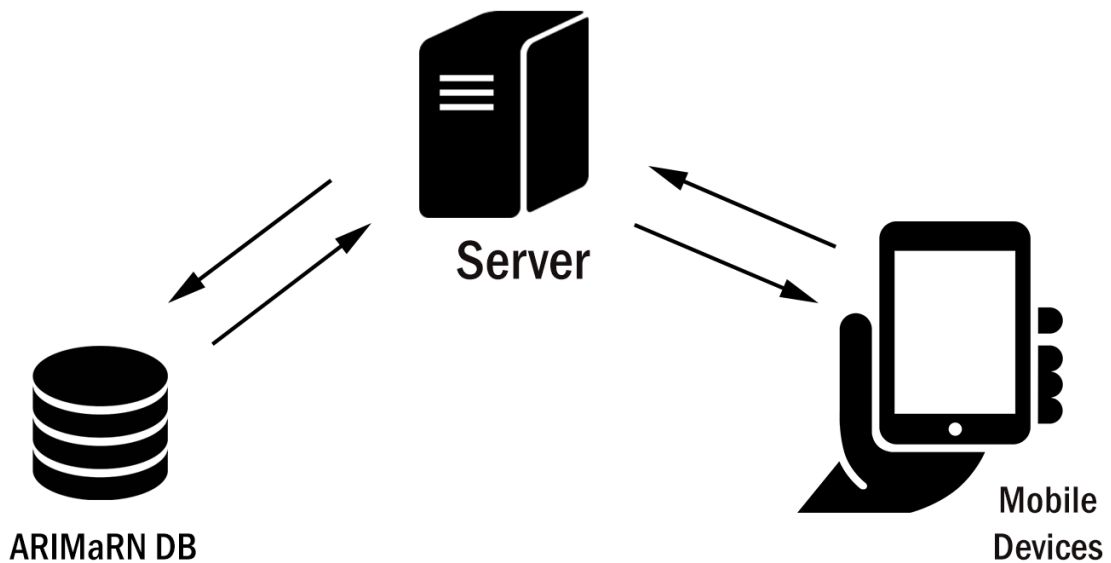


Figure 3.1: ARIMaRN Application Architecture.

3.4 ARIMaRN

3.4.1 The Database

The developed application, as shown in Figure 3.1, is composed of a database with the BIM and GIS data. Here, all information about the infrastructure is stored. Furthermore, these data can be updated in real-time with the information provided by the IoT sensors. Hence, the IoT sensors connected with the infrastructures gather information to complete and update the BIM's and GIS's data.

3.4.1.1 GIS Database

The Armis company has already a database with GIS data. However, this data is only the semantic one, therefore, the geometric representation of the infrastructures is not stored. Moreover, the Armis database also has some tables with BIM data. Thereby, the company's database has a mix of tables with GIS and BIM semantic data. The GIS tables were updated to fulfil the requirements of the test cases. These updates were in the table arguments and also in the data of each one if it was necessary.

3.4.1.2 BIM Database

The BIM database developed is a subset of a big one. Only the needed tables were created and updated. To avoid that the database access could be confusing, the BIM database subset was exported to the Armis database where there were already BIM tables. Thus, the BIM and GIS data are in the same database, but in different tables.

3.4.1.3 IoT

The IoT technology provides us with the updated data from the infrastructures. In the road networks field, we have information about, for instance, the traffic lights and electromagnetic coils. We could obtain this information in real-time and, for now, it is our IoT information. Furthermore, other sensors that already exist in our road networks can be used as IoT sensors. Thus, the IoT data in this project is the data from the traffic lights and electromagnetic coils, such as the colour displayed from a traffic light, the time that this colour will be kept, and the number of cars counts by one electromagnetic coil.

3.4.2 Server

To access the data from a mobile device, at any point in the world, a server was developed. The server connects to the database of GIS and BIM data and sends the requested data to the mobile device. Therefore, we don't need to have the database on our mobile device. They are not powerful regarding the computational resources available. Thus, using the server to connect to the database withdraws the need for a robust device to run the application. We can see the system architecture in 3.1.

3.4.3 AR Visualisation

The visualisation of the information and the interaction with the system is performed by a mobile application that communicates with the server to gather the necessary information and then displays the information to the user. The mobile application sends requests to the server repeatedly waiting sometime between each new request.

When the distance between the user and the infrastructure is between the stated thresholds, the app shows a virtual object, which can be pressed to display its BIM information. To complement the data with more than the GIS data received from the server, the app sends a request to get the BIM data for that infrastructure.

3.5 Application Metrics

To evaluate the two applications is taking into account the AR 's definition by Azuma in 1997 [Azu97]. He stated that an AR system must mix the virtual and real worlds, be interactive in real-time, and the real and virtual objects must be aligned in 3D. Furthermore, in the ARIMaRN app, was evaluated if the displayed information about the infrastructure is correct and if it effectively belongs to the infrastructure in the scene. Moreover, it is analysed the error accumulation on the localisation of the virtual objects.

3.6 Technologies

3.6.1 Global Positioning System Background

The nowadays mobile devices have location detection abilities based on GPS, Global Navigation Satellite System, Galileo and others. However, all the location systems have errors associated with their readings [SL15].

The Global Positioning System (GPS) is broadly used in our daily devices and applications, specially in the outdoor localisation estimate. The GPS was developed by the United States Department of Defence being operational in 1995. This system uses satellite communication to give us our position on the Earth. The satellite sends signals to our GPS receivers with its position and the exact time which the signal was sent. The information from several satellites is necessary to calculate the most accurate coordinates of our location. However, still today the GPS measurements are not accurate enough, especially on some mobile devices. The GPS readings require the view of the minimum number of four satellites [DZY⁺18]. Therefore, in some places, such as the urban and the forest environment, the values of GPS are not very accurate. At this places, the GPS receiver has difficulty to communicate to the expected satellites because of the many different barriers between them.

We also are still facing the problem of the different GPS positions even when the user stands in a fixed position. This phenomenon happens due to many reasons in which the main ones are the random errors in the measurements, the selection of distinct satellites in each measurement, the effects of the atmosphere in the satellites signals [Tho98], and the receiver's hardware and software related errors.

Another issue related to the GPS is the high power consumption. The GPS needs an antenna to communicate with the satellites and is always making network requests in our applications [DZY⁺18]. Thus, the GPS-based applications require much power consuming.

S. Tao, V. Manolopoulos, S. Rodriguez *et al.* [TMRR12] proposed the use of a Kalman filtering to transform the GPS results into dynamic position's estimates. Furthermore, if the values resulting from the Kalman filtering are not between their predefined limits the GPS estimate is discarded. Their goal is to have real-time information to support the development of Intelligent Transportation Systems. Furthermore, they used Motorola and Nokia mobile phones to receive the GPS information. Although they achieved good results for their goal, the accuracy they found necessary is less than 20 meters. Therefore, if we want to have an accurate GPS location, we need to update their algorithms' criteria.

J. Liu, C. Lin and C.Lin [LLL18] developed a study to evaluate which is the GPS best mode to use on Android devices. We have three different GPS modes: high-accuracy, battery-saving, and device-only. According to the authors, the high-accuracy mode uses the GPS receiver of the device, the Wi-Fi and the mobile network to calculate the most precise GPS value. The battery-saving one only uses the Wi-Fi and mobile networks. The device-only uses merely the GPS receiver. The study analysed the GPS modes either in the indoors and the outdoor environments.

Methodological Approach

The results have shown that in outdoor environments the device-only mode gives us the most accurate values. On the other hand, the battery-saving mode is the most accurate in the indoor environments. However, if the user is moving, the most precise mode is the high-accuracy one. Thus, even the GPS mode of the device influences the accuracy of our GPS readings.

J. Raiyn [Rai17] used a map-matching method to improve the estimate vehicle location in road traffic. The author also analysed different strategies for the car location such as the use of GPS, Global Navigation Satellite System (GNSS), Wi-Fi and cellular systems. However, although the GPS values gave them good accuracy, they joint their map matching method to reach more precise values.

On the road surface roughness measuring's field, K. Zang, J. Shen, H. Huang *et al.* [ZSH⁺18] proposed a system based on the GPS and accelerometer sensors of the mobile devices. Their system is mainly to use on bicycles. They use the GPS to record the distance and the travel speed, and the accelerometer to obtain the vertical displacement of the road. According to the authors, the use of the travel speed can lead to most precise results regarding the travel distance than the calculation of it using the GPS coordinates. The same is applied to the GPS altitude measurement that is still very poor regarding its accuracy. Therefore, the authors used the vertical accelerometer changes to detect the potholes and the humps on the road.

To have better results in the location of the autonomous driving vehicles, Y. Cho and H. Lee [CL17] proposed a system that through an adaptive Kalman filter makes the fusion of various sensors. In their work, they used Doppler, tachometer and GPS sensors to a better position estimation. Once again we could saw that the reliability and the accuracy of the GPS receivers demand us to build new algorithms to get more accurate readings.

D. Jwo and S. Wang [JW07] proposed a system based on fuzzy logic and the Strong Tracking Kalman Filter (STKF). Thereby, the authors, instead of using the traditional methods to process the GPS signals in the receivers used their system and compared the results regarding accuracy. According to them, when compared the results of their system with the results of the STKF and the Extended Kalman Filter(EKF) methods we can achieve visible improvements in the navigation accuracy and also in the tacking.

Y. Cho and H. Choi [CC14] proposed a system which uses the Kalman filter and a map-matching technique based on the point-to-curve matching to have more precise GPS values. The Kalman filter purifies the GPS readings and then is applied the Map Matching technique. The GPS filter is based on its associated errors, the atmosphere's and satellite errors and the receiver's hardware and software ones. This error variable is calculated by comparison with the GPS values of the map matching technique.

P. Kaniewski, R. Gil and S Konatowski [KGK16] studied which is the best technique to use on the GPS receivers to achieve the most accurate results. Their study targeted the Kalman filter, ordinary-least-squares (OLS), and weighted-least-squares(WLS). According to the authors and their study results, the use of the Kalman filter brings more accurate results. Therefore, they state that if it is possible to use the Kalman filter, we must use it.

N. Alsubaie, A. Youssef and N. Sheimy [AYES17] proposed a system to achieve more accurate values in the smartphones' geo-referencing. They used an algorithm to extract the orientation of the captured images, and based on that, their system updates the initial values of the Exterior Orientation Parameters (EOP). The authors draw to our attention the fact of their scale measurement, using the steps length of the user, can lead to errors when the step length changes between two compared images.

H. Du, C. Zhang, Q. Ye *et al.* [DZY⁺18] developed a hybrid system for outdoor localisation. Their system uses the GPS signals and the Wi-Fi fingerprints for getting the most accurate locations possible. Their findings point out that their system is more accurate than the widely used GPS method and also it consumes less energy. Furthermore, the use of the crowdsourcing fingerprints revealed useful in the location calculation.

J. How, N. Pohlman and C. Park [HPP02] proposed a system that analyses the GPS Doppler measurements so that it is possible to get a velocity value of a test vehicle with high accuracy. Thereby, we can achieve high precise location values discarding the external and internal influences issues.

H. Song and J. Lee [SL15] developed an algorithm to detect the wrong GPS readings using the moving windows approach. It uses the distance between two GPS locations and the acceleration to evaluate irregular speed values. Therefore, if the speed value is very different from the expected GPS reading, indeed it was biased from the environment and gave us the wrong information. Their algorithm uses the Haversine Formula and some authors-chosen distributions in the calculations. Thus, we can adapt it and implement our version of the algorithm if we want to adjust something on it.

3.6.1.1 Mock Locations

Simulating the GPS readings of other place is nowadays possible using an application to mock our locations. We have several apps for doing it on Android, being the *Fake gps location* application one example [Sak17]. The application allows us to choose the start location and we can use a joystick that appears on the device screen to change our location. The joystick can be used at the same time we use other applications. Moreover, we can specify the velocity that we want to move in every time we push the joystick in the app settings.

To allow our device using the app GPS readings we need to have the developer options enabled and inside of it, we need to open the option *Select mock location app* and choose the app to mock our locations.

This method brings us the possibility of testing the GPS-based applications without the need of walking around, being a handy tool.

3.6.1.2 Mathematics

3.6.1.2.1 Haversine Formula

In a GIS-based system, calculations with coordinates are very used. However, they are not trivial. The latitude and longitude angles form the coordinates on earth. Thereby, using the Haversine Formula, we can calculate the latitude and longitude values according to the spherical angles calculation. As the Haversine formula assumes that Earth is a sphere, instead of an ellipsoid as it really is, calculating large distances can give us inaccurate values. However, the Haversine formula is easier to understand and implement than other better accurate approaches. Furthermore, the Haversine Formula can give us very accurate results in short distances.

The Haversine Formula allows calculating the shortest distance between two points on the sphere's surface, given the two points' latitudes and longitudes. This formula derives from a general formula called the *Law of Haversines*, that relates the angles and the sides of spherical triangles [dBC97].

3.1, 3.2, 3.3, 3.4 and 3.5 equations show how using the haversine formula to calculate the distance between two points on a sphere's surface.

$$hav\left(\frac{d}{r}\right) = hav(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) hav(\lambda_2 - \lambda_1) \quad (3.1)$$

$$hav(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2} \quad (3.2)$$

Solving for d by applying the inverse haversine or by using the arcsine function:

$$d = rhav^{-1}(h) = 2r \arcsin(\sqrt{h}) \quad (3.3)$$

Where h is $hav\left(\frac{d}{r}\right)$, or more explicitly:

$$d = 2r \arcsin\left(\sqrt{hav(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) hav(\lambda_2 - \lambda_1)}\right) \quad (3.4)$$

$$= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \quad (3.5)$$

d is the distance between the two points;

r is the radius of the sphere;

φ_1, φ_2 are point 1's and point 2's latitude, in radians;

λ_1, λ_2 are point 1's and point 2's longitude, in radians.

3.6.1.2.2 Speed Formula

To calculate the value of the speed between two points we can use the 3.6 formula.

$$s = \frac{d}{t} \quad (3.6)$$

Methodological Approach

Where:

s is the speed;

d is the distance between the two points;

t is the time spent;

According to [SL15], the Maximum Speed value when the user is on Ambulation is $3m/s$. N. Carey [Car05] stated that the average walking speed of the male gender is 1.51 m/s. Therefore, values less than 0.5 m/s can be evaluated as GPS reading inaccuracy, and we can assume that the user is stopped.

3.6.1.2.3 Acceleration Formula

To calculate the value of the acceleration between two speed values we can use the 3.7 formula.

$$a = \frac{\Delta_{velocity}}{t} \quad (3.7)$$

Where:

a is the acceleration;

$\Delta_{velocity}$ is the *FinalVelocity* – *InitialVelocity*;

t is the time spent;

The Maximum Acceleration value to a human being is $3.09m/s^2$ according to [LML16]. This value was found by calculating the acceleration of the Usain Bolt when he was starting a race. As he is the fastest human being in the world, we can admit that no one can walk/run with a higher acceleration.

3.6.2 Augmented Reality Frameworks

The AR systems are mostly based on libraries or Software Development Kits (SDK) released by some companies or experts in the AR issues. Based on this two different tools available, we can quickly build new AR applications and do it more efficiently than if we start from the scratch. However, not every library or SDK has the same methods or functionalities to build the new apps. Thereby, evaluating which is the best to use is one of the most important parts to achieve better performance in our applications. As ARKit is exclusive to Apple devices, it will not be analysed its features in this document. Only the main universal SDKs are discussed below.

3.6.2.1 ARCore

Google released the ARCore 1.0 on February 23, 2018. The Google's goal is facing the ARKit from Apple's company in the AR field, where Google was a pioneer. Therefore, ARCore is a new Android SDK bringing the AR functionalities to the Android devices. It is important to emphasise that ARCore is only for Android devices. See Table 3.2. However, although the ARCore brings many possibilities for the AR Android world, in the first release, only 13 different Android devices

models are supported [Gos18]. See Table 3.1. Google only allows the use of ARCore for these 13 models because Google is gathering the information about the device's hardware and software, especially the specifications related to the camera of the devices [Ska17]. Thereby, Google guarantees that, to the allowed models, the tracking is the most accurate possible because they have the ARCore settings adapted to these 13 devices accordingly to their specifications.

The main ARCore embedded functionalities are the position tracking of the devices when it moves and the creation of an own understanding of the world around. In the tracking of the world, the ARCore executes the Concurrent Odometry and Mapping (COM) process to understand the environment. Thereby, the combination of the feature points detection with the use of the device's Inertial Measurement Unit (IMU) is the key to the ARCore tracking, estimating the cameras pose. The ARCore continuously analyses the environment and trying to detect feature points as well as planes. However, as we know, the feature point detection is performed to identify different and unique points in the image. Thereby, desks or other flat surfaces with just one colour may not be detected appropriately. ARCore also gives us the light estimation values bringing a robust feature to build the virtual objects with the correct illumination [Goo18a]. Thus, we can build better virtual objects, adapting their illumination settings to the real ones, leading to a more realistic system. See Table 3.3.

3.6.2.2 Wikitude

Wikitude is also an Augmented Reality SDK. It combines different features such as the Instant Tracking technology (SLAM), Object Recognition and Tracking, Image Recognition and Tracking and Geo-location AR for mobile devices. See Table 3.3. The latest release is the Wikitude version 7.2.1. As we can see, Wikitude is much more mature than ARCore. Furthermore, Wikitude has an API that integrates the ARCore, the ARKit and the Wikitude's SLAM engine. Therefore, the developer is abstracted of specific ways of coding to use in each AR SDK. Wikitude dynamically and automatically will identify the device, and it will choose the best tracking option to use from among the ARCore, ARKit and Wikitude's SLAM [Pau18].

A critical feature in Wikitude's SDK is the Geo-based Services. There are features available to work easily with geo-referenced data, allowing the creation of single or multiple Points of Interest (POI) with which users can interact. See Table 3.2. Although Wikitude supports the geolocation services, the Unity package does not. The Unity package uses the native API of Wikitude, and it does not support the location services yet. If we want the geolocation support, we need to use the JavaScript API of Wikitude.

Wikitude SDK is one of the most used AR SDKs, being the Vuforia and ARKit the first ones.

3.6.2.3 Vuforia

The Vuforia SDK can be used either for Android or Apple mobile devices. See Table 3.2. Its last version is the Vuforia SDK version 7.0.50 released on December 19, 2017. The Vuforia main features are the 2D and 3D Image Targets either markerless or with fiducial markers (last one

named VuMark in Vuforia). Furthermore, Vuforia supports Object Recognition, Cylinder Targets, Multi Targets, User Defined Targets, Cloud Recognition, and Virtual Buttons [Vuf18b]. See Table 3.3. Analysing the features of the Vuforia SDK, we can see it uses and is based on the computer vision to find feature points in the captured image, as expected from an AR SDK. Moreover, the tracking of the feature points is also performed always giving the best position of the virtual object.

Nowadays, Vuforia is practically the most used AR SDK. However, with the recent ARCore SDK release from Google, the market possibly can change favouring the Google company. However, ARCore needs to be compatible with almost all the devices as is now the Vuforia SDK.

3.6.2.4 ARToolKit

Another well known AR SDK is the ARToolKit. On September 20, 2017, was released its last version, the version 5.4. However, this version is not available to download at the ARToolKit official website yet. The available one is the version 5.3.2, released on March 23, 2016 [ART18]. ARToolKit supports iOS, Android, Microsoft Windows, Mac OS X, and Linux platforms making available the SDK to each platform. See Table 3.2.

The main features of the ARToolKit SDK are Robust Tracking, Strong Camera Calibration Support, Simultaneous tracking and Stereo Camera Support, and Multiple Languages Supported [ART15]. See Table 3.3. ARToolKit was practically a pioneer in the AR SDK releases. However, it is losing the fight against the other SDK, getting behind the other ones regarding its use.

3.6.3 Geolocation Frameworks

3.6.3.1 Motive.io

Motive.io is an AR SDK with geolocation services. This SDK aims to the development of location-based games such as Pokemon GO, where the user sees a map and is displayed over the map a character representing the user [Io18b]. See Table 3.2. The Motive.io SDK is mainly a location service, but it also allows the integration of other SDKs, such as Vuforia, Wikitude, among others, to provide the AR experience. See Table 3.3. However, this SDK was on early access and its first full release was only on March 21, 2018. Thereby, like the ARCore SDK, this SDK is doing its first steps, and need many improvements to beat the other ones already in the market.

3.6.3.2 Mapbox

Mapbox is another SDK providing the geolocation services. As Motive.io, Mapbox SDK only provides the location service, and you need to integrate it with another SDK to fulfil the AR requirements. Mapbox offer location features such as maps, search, and navigation [Map18]. See Table 3.3. This SDK can be used to mobile or to web applications. See Table 3.2.

The Mapbox SDK for Unity is a set of tools to create Unity apps with real map data. It converts map resources into game objects and has a graphical interface to work with Unity.

Methodological Approach

Table 3.1: ARCore Android Devices Supported Models [Gos18]

Brand	Model
Google	Pixel, Pixel XL, Pixel 2 and Pixel 2 XL
Samsung	Galaxy S8, S8+, Note8, S7 and S7 edge
LG	V30 and V30+ (Android O only)
ASUS	Zenfone AR
OnePlus	OnePlus 5

Table 3.2: Supported plugins, APIs and IDEs [Wik18, Goo18b, Vuf18c, ART16, Io18a, Map17]

SDK or Library	Supported plugins, APIs and IDEs
ARCore	Android Studio, Unity, Unreal, Android NDK, and Web
Wikitude	Cordova, Titanium, Unity, Xamarin, JavaScript API, Native API, Adobe Air, and Wikitude Ionic 3
Vuforia	Android Studio, Xcode, Visual Studio, and Unity.
ARToolKit	Android Studio, Microsoft Visual Studio, Xcode, GCC, and Unity
Motive.io	Motive Authoring Tool and Unity
Mapbox	Mapbox Studio and Unity

Table 3.3: SDK or Library Main Features. "X" means it support that feature.

SDK or Library	Fidutial Markers	Markerless	Tracking	Light Estimation	Geolocation
ARCore	X	X	X	X	
Wikitude	X	X	X		X (JS only)
Vuforia	X	X	X		
ARToolKit	X	X	X		
Motive.io					X
Mapbox					X

3.7 Tools

3.7.1 Revit Software

Revit software incorporates functionalities for architectural design, Mechanical, Electrical, and Plumbing (MEP) engineering, structural engineering and construction. The Revit tools enable the use of the intelligent and model-based process to plan, to design, building, and manage buildings and infrastructures. Thereby Revit is a multidisciplinary software providing also an integrated cloud-based web service that enables centralised team access to project data. Thus, Revit is a

powerful software able to manage our infrastructures more smartly. The last official release is the version Revit 2018 [Aut17].

Revit allows users to manipulate buildings, assemblies or individual 3D shapes. The software is not a Non-Uniform Rational Basis Spline (NURBS) modeller. Furthermore, it cannot manage the individual polygons of an object except on some specific object types. Revit calls the object's categories as families. They can be divided into three groups: System Families, Loadable Families / Components, and In-Place Families. The software already has different families however sometimes the users are forced to create their own families, or even buy them [con18].

Revit also has a plugin that enables the export and the import of the data to and from databases, respectively. The plugin is called as Revit DB link. However, not every ODBC databases are supported, although the documentation says that it supports ODBC databases [Aut18].

3.7.2 AutoCAD Software

AutoCAD was initially used for the designing of two-dimensional (2D) technical drawing and modelling three-dimensional (3D). In addition to the technical drawings are available several features for viewing in multiple formats. AutoCAD is broadly used in architecture, interior design, civil engineering, mechanical engineering, geographic engineering, electrical engineering and in various other branches of industry. Although its many features, AutoCAD is still mainly used to represent the real-life infrastructures in 2D drawings. In these drawings, the user can describe the mechanical, electrical, and other sections of one infrastructure or set of them. For instance, an AutoCAD 2D drawing can represent a road junction and its infrastructures. The last official release is the version AutoCAD 2018 [Aut16].

3.7.3 QGIS Software

QGIS software is an open-source application that supports editing, viewing, and analysing the geospatial data. We can compose and export graphical maps in QGIS. Furthermore, it supports either raster and vector layers. As QGIS can be integrated with other GIS packages, such as PostGIS, this tool could be very useful in the geospatial data processing. The last released version is QGIS 3.2.2 [QGI18].

3.7.4 Unity

Unity is a multi-platform game engine. It allows the development of games for all platforms and also can be used to develop applications in the VR/AR field. The graphics interface brings a good understanding of what is happening and how our virtual objects are composed. Furthermore, we can make changes to our objects, and even to our script values, just updating some values in the Unity window, or using our mouse. To build the behaviour of our application the Unity engine uses the system of scripts. Thereby, writing a script and associating it with an object changes its behaviour. Thus, Unity uses a simple concept to achieve complex performances. Unity has three different version where the Personal one is free. However, the brand of the Unity always opens at

the beginning of our application in this version. The scripts are usually written in C# or JavaScript. Furthermore, we can use our favourite Integrated Development Environment (IDE) at the writing time. The last release version of Unity is Unity 2017.3.1f1 [Uni18]. In this version, the Vuforia SDK is integrated with Unity natively. Thus, we do not need to import the Vuforia Unity package anymore. We just need to select the Vuforia Support in the Unity install manager.

3.7.5 Microsoft Visual Studio

Microsoft Visual Studio, developed by Microsoft, is an Integrated Development Environment with an emphasis on the C, C++, C# and J# programming languages development. However, it also can be used with other programming languages, such the ones in the web development area [Mic17c]. Microsoft Visual Studio is very used as the IDE to the development of the Unity Scripts mainly due to its C# support.

The last released versions of Microsoft Visual Studio are the Community 2017, Professional 2017, and Enterprise 2017. The Enterprise version has full support for all the visual studio features. The Community version does not support the majority of testing tools and Cross-platform Development. Finally, the Professional version is in the middle of the two discussed ones in terms of features supported [Mic17a].

3.7.6 Android Studio

Android Studio is an IDE for the Android operating system. It was build based on the IntelliJ IDEA software, built by JetBrains, and it is designed to the Android development. We can use it on Windows, macOS or Linux. It mostly uses Grable as the build tool, and it provides us with a Virtual Device emulator which allows us the running and the debugging of the applications [And18].

3.7.6.1 Monitor Tool (Android SDK)

Debugging an android application, especially the ones created on Unity engine, can be tricky work. However, we can use the available Android SDK tools to help us. Nevertheless, using this tools, we need to have our Android device connected to a computer. Therefore, using the Monitor tool, located at the path SDK/tools, we can create a filter in the LogCat tab to show us only the Unity logs. These have the tag "Unity", so we use that tag to filter the logs.

3.7.6.2 Gradle

Gradle is a build automation system. It is based on Apache Maven, 3.7.8.2, and adds a domain-specific language (DSL) rather than the XML used by Apache Maven. Furthermore, it uses a directed acyclic graph to determine in which order the tasks can run [Gra18]. Gradle is the Android Studio build automation system.

3.7.7 Databases

Today, several database technologies are available in the market. Every one of them has different ways of working, but practically all can perform our desired tasks.

Taking into account the use of BIM and GIS, a relational database is necessary to fulfil the requirements. We can export either BIM or GIS data to a relational database, where we can find the information about any element.

3.7.7.1 PostgreSQL Database

In the Geographic Information Systems area, one of the most used relational databases is the PostgreSQL, alongside with the spatial database extender PostGIS. The PostGIS extender gives the PostgreSQL database the ability to handle location queries [Pos15]. Thus, if we export the GIS data to a PostgreSQL database with the PostGIS extender, the data can be accessed through database queries. Moreover, if we open the GIS tables in the QGIS software 3.7.3, we can see the elements in the map.

On the other hand, a BIM database is also a relational database. So we can export the BIM database to a PostgreSQL one. As BIM does not have spatial attributes, we do not use the PostGIS extender here. Furthermore, we can build just one PostgreSQL database with both data from BIM and GIS and access them. We access the tables' data through queries. Thus, if we named the tables differently, we can have the two relational schemas in just one database.

3.7.7.2 Microsoft SQL Server Database

Microsoft SQL Server is also a relational database management system, and it was developed by Microsoft. It has several different editions to fulfil the users' demands [Mic17b]. However, in the GIS field, The Microsoft SQL Server is not a standard to use. On the other hand, the BIM database can be easily represented by a SQL Server database.

Although SQL Server can handle very different types of the data, PostgreSQL brings us more features to do it.

3.7.7.3 DBeaver

DBeaver is a software that allows us to have access to the SQL Client databases through an interface. Thereby, we can see the UML design of a database and better understand its relationships, constraints and tables. DBeaver supports all popular databases, such as MySQL, PostgreSQL, MariaDB, SQLite, Oracle, DB2, SQL Server, Sybase, MS Access, Teradata, Firebird, Derby, and others [DBe]. Thus, we can use this software to analyse better our database and have a full understanding of it.

3.7.8 Server

A RESTful system seems to be the best solution to use in a system which connects to multiple devices. It is based on representation, state, and transfer concepts. In the representation concept, the data is a model showing the way the resources are distributed. The server and the clients know the representation. The state concept requires that all states have to be provided with a request. The transfer concept allows the exchange of the representations and states between the client and the server [TBDH14]. Therefore, connecting the server to a database, we can have access to the database worldwide, through the server.

3.7.8.1 Spring Boot framework

Spring Boot is a Java-based framework able to run RESTful servers. This framework is a well-known one mainly due to its features. For instance, with simple annotations in our code spring boot can generate queries to our database, without the need of their extensive writing by us, using other frameworks to it. Thereby, we just create interfaces and within them methods with predefined names [Spr18]. Afterwards, we only need to call the methods, and the Spring Boot framework will generate the code to do the queries for us using, for instance, the Hibernate framework. Thereby, Spring Boot allows us to have easy integration with other Java frameworks such as JPA/Hibernate ORM, among others. With all these capabilities Spring Boot is the state of the art Web Model View Controller (MVC) framework for the creation of web applications [WSL⁺18].

3.7.8.2 Apache Maven

Maven is a build automation tool, and it is mainly used for Java projects. The Java libraries are downloaded by Maven without the user interference. We just specify the maven dependencies in the XML file named *pom* and the tool does the work for us [Pro18]. Moreover, when we migrate our project to another computer the Maven tool downloads all the necessary libraries and builds the project automatically. Therefore, the user never needs to download, copy, and paste libraries in the correct directory. As we can see, the Maven tool it's advantageous in the Java projects development.

3.7.8.3 Hibernate framework

In the Java programming development, Hibernate ORM is an object-relational mapping tool that provides us with a framework for mapping the objects to a relational database. It generates database calls, and the developer does not need to make the handling and object conversion manually [Hib18].

Hibernate is also an implementation of the Java Persistence API specification. Thus, it can be used in environments that support the JPA. Moreover, Hibernate is also very used with the Spring Boot framework.

3.7.8.4 JPA framework

The Java Persistence API (JPA) is a Java application programming interface specification that manages the relational data in applications using Java. In the Spring Boot projects, it is used to communicate between the Spring Boot framework and the Hibernate one. Thereby, the JPA does the necessary translation so that the two other frameworks can perform their task without know each other. The JPA works here as an intermediate that manages the data from each side [Ora18].

To give the instructions to JPA the user need to use Java annotations such as *@Entity*, *@Column*, *@Table*, among others. These annotations set the Java class as a JPA Entity to access the database, and they give it the hints to know where the user wants to save the data from the database inside of the Java class, and which table the user wants to access.

3.7.8.5 Thymeleaf

Thymeleaf is a Java template engine that can work in web and also non-web environments. Thymeleaf allows displaying the HTML files correctly in browsers. It has even modules to integrate with the Spring Boot Framework being very likely to use in the HTML web development [Thy18].

3.7.9 Eclipse

Eclipse provides primarily IDEs for almost every language. The most well-known ones are the Java IDE, C/C++, JavaScript and PHP IDEs. It was made by Java programming, and it is an open-source development software. The main programming language supported is Java, the other ones are mostly supported by adding new plugins to the IDE. Furthermore, several tools can be added to the Eclipse IDE such as GUI builders, and tools for testing, modelling, among others. Nowadays, Eclipse is practically the most used Java IDE in the world. The last version released was Eclipse OXYGEN [Ecl18].

3.8 Summary

In short, the ARIMaRN application aims to allow access to the information about the road infrastructures. To accomplish that, the system uses a server with a database based on the BIM and GIS data, updated by the IoT sensors information. Through a mobile device application, the system can present the information about any infrastructure using the AR technology, displaying the information received from the server.

The app provides an integration of the BIM, GIS, IoT, and VR/AR technologies. Thereby, we have smarter management and maintenance of the road infrastructures. Furthermore, the app can provide access to information about infrastructures status such as the temperature, the energy consumption, the last performed revision, the status of degradation, among others. Hence, we can have smarter decision-making using the system to analyse the infrastructures.

Methodological Approach

The ARIMaRN-PSAL app allows us to place a traffic signal in one position and then analyse it regarding the best location. Therefore, we can also better manage our road infrastructure by seeing them in the local through the application.

When we are talking about developing a VR/AR application, the name Unity always comes out as the engine to be used. The different available tools in Unity allows us to build VR/AR application efficiently and it supports all platforms. Moreover, practically every SDK is supported by Unity giving an interface easy to use. Alongside with Unity name also comes the Microsoft Visual Studio IDE. The Visual Studio support for the C# programming language is a helpful tool. Furthermore, the majority of the Unity apps developers uses it.

When we are in the databases field, regarding the four technologies implementation, the PostgreSQL with the PostGIS extender is almost a standard in this developing area, mainly because it can support spatial or location queries and also can be the storage of the BIM data. If we are facing an application with a big amount of data from either BIM and GIS, using two separated PostgreSQL database is recommended. On the other hand, if the tracking of the data can be easily made, we can use the same database and just name differently the tables that have GIS and BIM data. However, this two methods can be a discussion target regarding which will have a better performance in the data access.

To enable the access to the database the use of a RESTful server is one of the best ways to do it. When we are talking about the mobile application, the RESTful servers make perfect sense because they can provide the information worldwide and efficiently. Moreover, the Spring Boot Java-based framework can provide us with a fast way of developing a RESTful server. If we use the Maven tool alongside with it, we can design and development more efficiently the server.

Thus, using a RESTful server connected to a PostgreSQL database, we can send the necessary information to the AR location-based application and build the virtual scene overlapping the real world scene. If we update the database with the IoT sensors, we can achieve the BIM-GIS-IoT-VR/AR integration successfully.

The SDKs are a massive help in the AR field. They allow us to do the upper-layer code, instead of implementing the computer vision algorithms to recognise the features points in the image. Thereby, we can use the SDK API just to call a function that does all the computer vision work for us and we only need to think about how our application will work apart from the recognising step. Obviously, if it is required, we can change or implement our approach to the recognising stage, or to other one provided by the SDK.

Some SDK were analysed in this Chapter, looking for an integration between the BIM, GIS, IoT, and VR/AR technologies the use of an AR SDK with geolocation services leverage us regarding development's time consumption. Developing a geolocation service requires its time, and it is not an easy task. However, the geolocation services in the AR field still need to improve. Today's ones do not have accurate calculations, and the virtual objects are often displayed in the wrong place. Even if the distance from the virtual object to the right position is not too long, it can mislead the user or the application itself.

Methodological Approach

Furthermore, as this kind of features are now appearing and the AR SDKs are also evolving, the majority of the mobile devices do not support them. Thereby, the available devices to test the developed applications can be a huge restriction to the development in the AR geolocation field.

Another approach is the use of an SDK with geolocation services alongside with an AR SDK. The combination of the two SDK can give us good results too. Nevertheless, we cannot forget the problems with the geolocation accuracy and supported devices.

The use of the GPS is mandatory for this kind of integration since we are talking about GIS-based applications. The GPS is widely used yet it still needs more improvements in its readings. The atmosphere, hardware and software, and the unavailability of the necessary satellites to communicate are barriers to accurate GPS values. If we are in the middle of a city with big building or a forest the GPS receiver does not give us precise results. Thereby, algorithms to improve the accuracy of the GPS and to provide us with more steady values are the next step of research.

To overcome some of the GPS problems, especially in the application's test phase, we can use the mock location technique. Therefore, we have steady and precise GPS readings to test specific features of our app. Thus, the mock locations technique can be useful for testing and debugging, mainly because we do not need to go outside our office.

Methodological Approach

Chapter 4

Implementation

This chapter describes with more detail the implementation of the proposed system. The architecture, the technologies, and all steps performed to achieve the integration between the BIM, GIS, IoT, and VR/AR technologies are analyzed and discussed here.

4.1 ARIMaRN-PSAL

Wikitude and ARCore are not supported by my OP3, as we saw at 3.1 and A.5. Thus, it was tried to use Vuforia SDK this time. In addition to the fact that it is one of the most used SDK in the world, Vuforia's last release brings us the SLAM feature too. Furthermore, Vuforia is supported by many more devices than the others SDKs.

4.1.1 Unity Setup

As the used Unity version has the Vuforia SDK natively, it is not needed to import the Vuforia package to the Unity project. After the setup configurations, we need to put the Vuforia's licence to build the app.

Before that, a Vuforia *ARCamera* game object was added to the scene. Now, the *ARCamera* was selected and configured to use Vuforia and SLAM. In the *Device Tracker* option it was checked the *Track Device Pose* option and inside of it, in the *Tracking mode*, it was chosen *Positional*. Now the camera settings for Vuforia are completed.

4.1.2 Scene

In the scene, it was added the Vuforia *Plane Finder* as a child of my *ARCamera*, and then the Vuforia *Ground Plane Stage*. Now the SLAM setup is done. However, if we want to put a model to appear when we tap the device's screen, it will not have the behaviour we want. The default behaviour in Vuforia is that every time you touch the screen a new clone of the model will be added to the scene. That is not the case for the application. It is wanted to place the traffic signal

Implementation

just one time. Thereby, Vuforia documentation gives us a script [Vuf18a]. This script changes the Vuforia SLAM behaviour to the one wanted. When we tap the screen, the model appears if we tap it again the model appears at the other place I touched. Using it, the app does not create clones.

4.1.3 Scripts

Thus, it was created a new Script with the name of the Vuforia Script Class, *DeployStageOnce*, with the pasted the code from Vuforia Documentation into it. Now, in the *Plane Finder* game object, we need to remove the *Content Positioning Behaviour Script* component. The next step is the addition of the *DeployStageOnce* Script as a component of the *Plane Finder*. Therefore, we replaced the default script with the Vuforia Documentation one.

In the added script, we have an empty *Anchor Stage*. Thereby, it was dragged the *Ground Plane Find* game object into it. After that, the same model of the traffic light used in the Wikitude approach (A.4.1) was added to the scene as a child of the *Ground Plane Stage*.

Now, we need to enable Unity to call the function in our script that puts the model on the screen. Thereby, it was selected the *Plane Finder* game object and, in the *Inspector* tab, it was open the *Plane Finder Behaviour Script* drop-down. There, it was changed the *Mode* to *INTERACTIVE*. After, it was dragged the *Plane Finder* game object to the empty box inside the *Advanced* option. In the drop-down at the right, it was chosen the *DeployStageOnce* script, and then the *OnInteractiveHitTest* function. This function allows us to put the model on the screen wherever we touch it, as we can see at Listing 4.1.

Listing 4.1: Vuforia OnInteractiveHitTest Function

```
1 public void OnInteractiveHitTest (HitTestResult result) {
2     if (result == null || AnchorStage == null) {
3         Debug.LogWarning("Hit test is invalid or AnchorStage not set");
4         return; }
5     var anchor = _deviceTracker.CreatePlaneAnchor (Guid.NewGuid().ToString()
6         , result);
7     if (anchor != null) {
8         AnchorStage.transform.parent = anchor.transform;
9         AnchorStage.transform.localPosition = Vector3.zero;
10        AnchorStage.transform.localRotation = Quaternion.identity;
11        AnchorStage.SetActive (true); }
12    if (_previousAnchor != null) {
13        Destroy(_previousAnchor); }
14    _previousAnchor = anchor;
15 }
```

4.1.4 Application's Build and Run

Now, the application is complete, and it is necessary to test it. It was connected the OP3 to the computer and chosen the Build and Run option of Unity. The application built successfully. In

Implementation

the OP3 it was given the necessary permissions to the app start, and it opened. The camera was working, and it is an improvement compared to the Wikitude approach.

However, when we touch the screen, no model appears. Now, we have a different problem to deal with. However, in the same page of the Vuforia Documentation where they give us the script, they talk about a way of simulating the use of the application in a mobile device in case the device cannot support SLAM. As Wikitude SLAM feature did not run in the OP3, maybe the smartphone cannot run the SLAM feature of any SDK, at least by now. Thereby, it was followed the Vuforia Documentation to simulate the running of the app. And it is quite simple. We just need to use our computer with an external camera, and a fiducial marker given by Vuforia. It can be a webcam or a more powerful one. The fiducial marker is automatically recognised for the SLAM tracking as the ground surface. This feature is already deployed and the marker is in *Assets/Editor/Vuforia/ForPrint/Emulator/* folder of the Vuforia package.

Therefore, we just need to print out the fiducial mark and run the application on Unity. Now we point out to the fiducial marker, and when we click with the mouse on the Unity screen, the model appears correctly. Thus, the app is correct and completed. The OP3 device seems to not support any SLAM tracking by now.

4.2 ARIMaRN

4.2.1 Architecture

The ARIMaRN application is based on a centralised architecture, where the mobile devices communicate with the server to get the necessary information. The server sends the information required after the querying to the database. The application on the mobile device manages the received data and creates the AR environment, as we can see at Figure 4.1.

The database is a PostgreSQL one with the extender PostGIS to handle the GIS data. Therefore, the same database has BIM and GIS data in different accessible tables. It was used the Java Spring Boot framework for building the RESTful server to connect to the mobile device and the database. Then the GPS side was developed using the Unity engine tools. Moreover, Unity, alongside Vuforia SDK, is the base of the AR environment.

4.2.2 Database

As we saw in Chapters 2 and 3, to store the GIS and BIM data we need a relational database. The Porto City Hall made available to Armis a database (PDB) of information about some Porto city's road infrastructures. That database was stored in paper, excel files, and AutoCAD files. From that files was created a GIS database using an algorithm to read the information from excel file and insert the data into the GIS tables. Some non-GIS data exists in the files given by the Porto City Hall. Thereby, we can convert that info to a BIM database.

Implementation

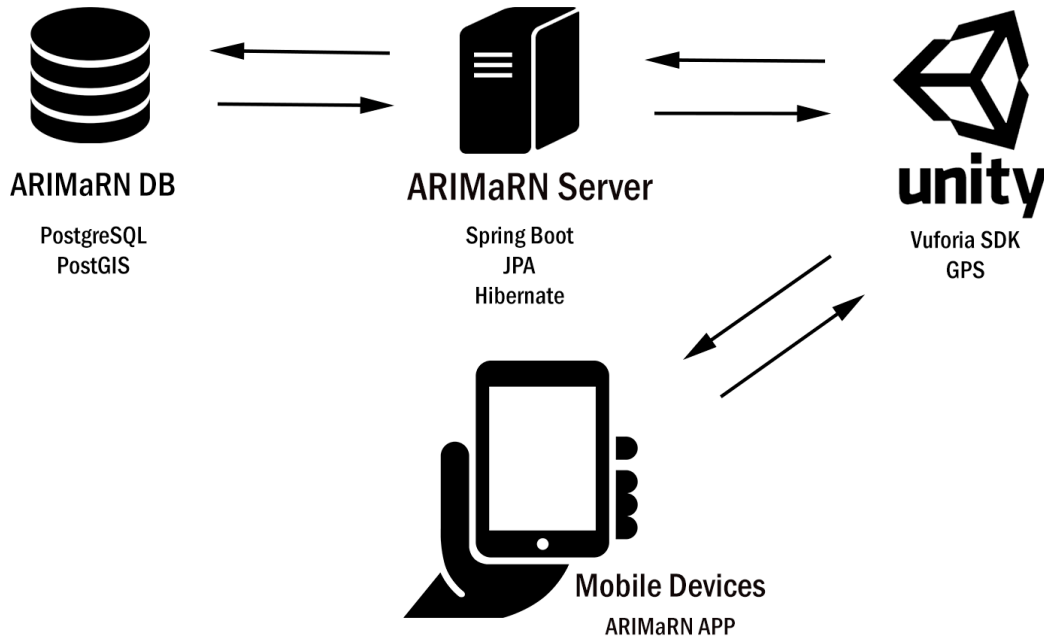


Figure 4.1: ARIMaRN Application Detailed Architecture.

The data is mostly related to road junctions, where we have data from the signal lights, crosswalks, and electromagnetic coils. However, the PDB does not have information about every road junctions, or infrastructures.

4.2.2.1 BIM Database

A study on how is built a BIM database and also how it works was the first step to create my own BIM database. The Revit software was used to create a BIM database automatically. The first try was the exporting of a project sample from the Revit samples. Using the simple export of Revit, that exports the data to an ODBC database, a PostgreSQL BIM database was created. To export the data, it is necessary the configuration of the ODBC database as the database we want. In my case, the configuration leads to a PostgreSQL database.

As this export had good results, it was tried to export the information of a traffic light model to the PostgreSQL database [War07]. First, it was necessary to convert the Sketchup file into a file supported by Revit. Using the Revit option, the conversion of the Sketchup file to a FormIt format one was made. After that, again using the Revit options, it was imported the FormIt format file to an RVT one. Thereby, the Revit software could open the traffic signal model. Afterwards, the same operation to export the model data into the PostgreSQL database was made. However, this time, the database only has information about the default tables resulting from the Revit exportation. The main one is the *Materials* table where are stored all the materials available in the Revit software to use in the models.

Implementation

Trying to understand the reason why the database is empty, it was decided to use the Revit DB link plugin to export the model's data to the database. However, this time we are facing an error. The Revit plugin could not export adequately to the PostgreSQL database for some unknown reason. As we saw in Chapter 3 Section 3.7.1, the Revit documentation claims that the Revit DB link plugin can export and import data to ODBC databases. Thereby, it was installed the Microsoft SQL Server database to test if the plugin works.

After the SQL Server installation, it was tried to export the traffic signal model again. This time no error occurred and the model was exported successfully. Analysing the resulting database, we face the same problem that we had before. The database is empty, except the tables related to the default settings of Revit, such as the *Materials* table. Thus, the problem was not the way we export the model.

To test whether the problem was from Revit or the semaphore model, it was created two walls around the signal and gave them materials. Now, as in the first time, the model was exported to the PostgreSQL database. Analysing it again, the walls are now in the *Walls* table with their relationships with the *Materials* table through the *WallTypes* table. Thus, we can conclude that the problem is the way the traffic light signal was designed in the Revit software. However, we could not export successfully any other traffic light's free model. The other ones available for free can be opened in the Revit software, but the Export to ODBC option is not available to them [Rev17b, CAD15].

Therefore, the attempts to understand how a traffic light BIM database is composed were forced to stop. Thus, it was analysed the resulting database of a sample project from Revit, named *rac_advanced_sample_project.rvt* [Rev17a]. To analyse it, it was used the DBeaver software to have an overview of the database by visualising its UML. See the UML cut image here A.1. Here we can found that in the middle of the 228 database tables the relationships between them are not trivial, and not related to a model table. We have relationships between the available materials' table and the used materials one, and also between the materials used and which component used it. However, we do not have a table, for instance, named *House* that have relationships with the elements of a house. Therefore, the BIM tables from Revit are related to each other but not to one main table that designates who are the elements belonging to each model. If we want to have more than one Revit model exported into one database, afterwards of the inserting of the data in the tables we cannot know which element belongs to the house A or the house X. Furthermore, the Revit exportation tool deletes the database and create a new one always we export to it. Thereby, it is necessary to export to different databases to not override the data. Only after that, we can merge the two databases.

Facing these adversities, and as BIM still have not much support for other infrastructures than buildings, it was chosen the use of one subset of the Revit BIM database as the BIM database. However, in the new database, the data will be related to the infrastructure. Thus, we can access any element and know to whom it belongs. Thereby, it was created a BIM database subset where we can find information about the traffic lights, crosswalks, and electromagnetic coils.

The BIM database subset is composed of nine tables: *arm_tl_controller*; *arm_tl_group*, *arm*

Implementation

_tl_plan, *arm_tl_step_group*, *arm_tl_step*, *bim_intersection*, *bim_crosswalks*, *bim_espiras*, and *bim_materials*. See the UML diagram here 4.2 and here A.2. The *bim_materials* table was filled with the data of the *Materials* table from the Revit export. Thereby, the Revit materials are available to use in the database. Moreover, it was added one new material. This one was named "Crosswalk Material", and it is the material of a standard crosswalk.

The *arm_tl_controller*, *arm_tl_group*, *arm_tl_plan*, *arm_tl_step_group*, and *arm_tl_step* tables have the information about the traffic lights as they are from the PDB. In the respective order, we have the information about the controller that coordinates a group of traffic lights, the information about the traffic lights group, the data about the plan of steps used in the group, the info about a step group, and the information about a specific step. The step gives us the information about which light will be displayed in the traffic light, for instance, the red light. A road junction has a set of traffic lights, here designated as a group. Each group of traffic lights has a group of steps to coordinate with the others. This group of steps is composed of three individual steps corresponding to the three different colours available, regarding a simple traffic light. The whole group of traffic lights is controlled by one unique controller to change the steps according to the group of steps assigned to the group. Thereby, for instance, when some traffic lights are green, the others are red, according to the plan of steps.

The traffic lights tables already were in the PDB, see 4.2.2.2. Therefore, I put my *bim_materials* table in the PDB. Moreover, it was added the *bim_intersection*, *bim_crosswalks*, and *bim_espiras* tables too. The *bim_intersection* table says which is the number of the road junction according to the Porto City Hall designation. The *bim_crosswalks* table represents the crosswalks on the road and has a relationship with the *bim_materials* table to give us the material of the element. The *bim_espiras* table represents the electromagnetic coils on the road and also has a relationship with the materials table. Thus, we can store information in this BIM database subset about the traffic lights, the crosswalks and the electromagnetic coils of a road junction. Moreover, the controllers, the steps and the intersections info are also available.

The tables whose names start with "*bim_*" have constraints to ensure the good use of the database. For instance, the *intersection_id* attribute in the *bim_crosswalks* and *bim_espiras* tables cannot have an id not available in the *bim_intersection* table, as we can see at Figure 4.2.

4.2.2.2 GIS Database

The GIS database is based mainly on the PDB. Firstly, it was studied the tables and was seen the representation of them in the QGIS software. Here, it was used the *planet_osm_point* table because it has the information about the location of the controller of the traffic lights in each crossroad junction. After analysing it, we can conclude that this table would be not very useful because it is necessary to know the traffic light post position and not only the controller of the all group of traffic signals. Thus, it was decided to update the PDB database with more tables and update some of the existing ones.

The *gis_espiras* and the *gis_crosswalks* tables were added. Moreover, the *arm_traffic_light* table was modified. The column named *coordinates* was added to it. This column represents the

Implementation

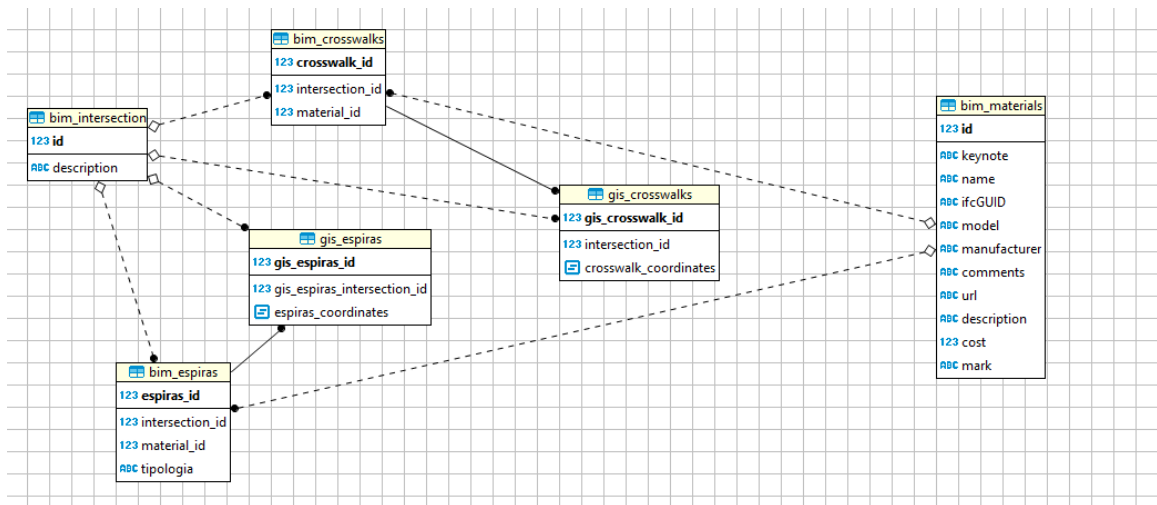


Figure 4.2: BIM and GIS Added Tables.

position of the traffic light post in the geometry type. Thus, the coordinates of the traffic lights posts whose controller is the controller of the road junction we want to test must be inserted in this table. Thus, now we have the location of all the traffic lights posts instead of only the position of the controller, as the Armis database just had.

To update the *arm_traffic_light* table with the coordinate values, the user must use the query in the 4.2 listing.

Listing 4.2: Update Query. This query updates the values of the coordinate column of the traffic_light_id = 1 to the inserted ones.

```

1 update arm_traffic_light set coordinates = ST_SetSRID(ST_MakePoint(
    latitude_value, longitude_value),4326) where traffic_light_id = 1
  
```

The *gis_espiras* table has the information about the electromagnetic coils' coordinates, id, and its intersection id. The *gis_crosswalks* gives us the data related to the crosswalks such as the id, intersection id, and the crosswalk coordinates. The coordinates of the controllers are given by the *planet_osm_point* table. See the UML here 4.2. However, we just need to know the location of the traffic light because it is related to the controller id. Thus, the *planet_osm_point* table will not be very used.

Once again, the *gis_espiras* and the *gis_crosswalks* tables need to be updated with data about each one. They were created but they are empty. So to test the database we need to insert data into these two tables.

The tables whose names start with "gis" have constraints to ensure the good use of the database.

After the designing of the databases and the creation of the necessary tables, the BIM and GIS databases were merged into only one. Thus, in the new database, we have the GIS database received from Armis company and a subset of a BIM database. Furthermore, it was updated

Implementation

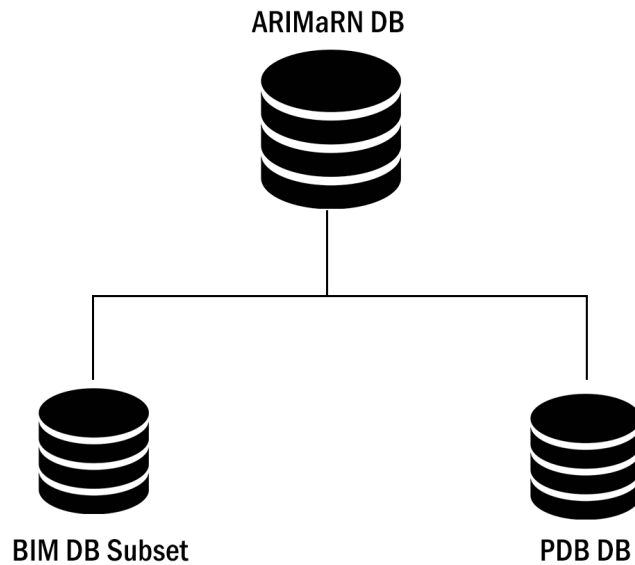


Figure 4.3: Database Architecture. The PDB was updated with new tables, data and columns, the BIM database subset created, and then the two databases were merged into the ARIMaRN database.

the GIS database to satisfy the requirements. It was chosen to have the BIM and GIS databases together because the Armis PDB already have a mix of tables of BIM and GIS data. Thereby, it is less confusing than if it was put a new database just for BIM data and the GIS database still had BIM data. See image [4.3](#)

It is also important to point out that the GIS tables created have constraints so that the identifiers of the GIS tables are the same as the ones in the BIM tables. Thereby, the BIM and GIS crosswalks and espiras tables have constraints using a foreign key for each other identifier. Thus, for instance, we cannot create crosswalks with different ids, and also the BIM identifiers are related to the GIS ones, allowing intuitive access to the database.

4.2.3 Server

The developed system uses a RESTful Server to exchange the information between the database and the clients. The chosen framework for the server development was the Spring Boot Java-based one, analysed in [3.7.8](#).

4.2.3.1 Server Setup

To create the RESTful server it was created a Maven project in the Eclipse IDE. Afterwards, the Spring Boot Maven dependencies that are needed and the ones to connect to the PostgreSQL database were added. To accomplish that it was added the dependencies to the *pom.xml* file [A.1](#). Here, the database dependencies are the ones related to the Hibernate and JPA. Furthermore, we need to create an *application.properties* file in the *src/main/resources* path and specify

Implementation

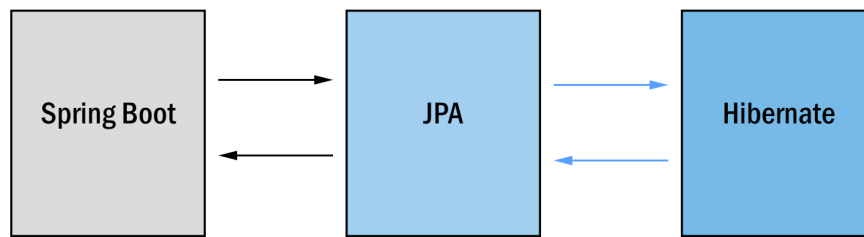


Figure 4.4: Spring Boot, JPA, and Hibernate Architecture.

the database settings, as shown at 4.3. In this file, it was specified the database to use and its credentials, the JPA and Hibernate options and also the Thymeleaf settings.

Listing 4.3: application.properties text file

```
1 # Disable Thymeleaf caching for hot-swapping of templates
2 spring.thymeleaf.cache = false
3 # Database connection settings
4 spring.datasource.url=jdbc:postgresql://192.168.1.28/arimar
5 spring.datasource.username=joelcarneiro
6 spring.datasource.password=123
7 # Show or not log for each sql query
8 spring.jpa.show-sql = true
9 # needed for Location domain class
10 spring.jpa.properties.hibernate.dialect=org.hibernate.spatial.dialect.postgis.
    PostgisDialect
11 # JPA
12 spring.jpa.properties.hibernate.id.new_generator_mappings = false
13 spring.jpa.properties.hibernate.format_sql = true
14 spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.
    PhysicalNamingStrategyStandardImpl
```

Accordingly to my settings, the Spring Boot framework interacts with the JPA framework and it uses Hibernate to generate the queries. Then, the Hibernate send the data to JPA, and it translates the information to Spring Boot as we can see at Figure 4.4. Thereby, Spring Boot can efficiently perform queries to the database, and we do not need to implement this database access from scratch. After these configurations, the setup for the RESTful server was done.

4.2.3.2 Web Languages

Following the setup, it was started the development of the server itself. First, it was created a homepage as an interface to the server. To use web languages in Spring Boot, we need to create a folder named *templates*, and other one called *static*. Both of them need to be at the *src/main/resources* path. After of the folders' creation, it was put the HTML file in the template folder and the CSS and JavaScript files in the static folder. To not take much time on this task it was used a bootstrap template and adjusted the HTML file inside of it to fit my project [Boo]. Moreover,

Implementation

to load the files inside the static folder, it was used the Thymeleaf template engine. Thereby, we need to use the special way of Thymeleaf to call the files as shown on [4.4](#)

Listing 4.4: application.properties text file

```
1 <!-- Bootstrap core CSS -->
2 <link th:href="@{css/bootstrap/vendor/bootstrap/css/bootstrap.min.css}" rel="
  stylesheet"></link>
3 <!-- Plugin CSS -->
4 <link th:href="@{css/bootstrap/device-mockups/device-mockups.min.css}" rel="
  stylesheet"></link>
5 <!-- Custom styles for this template -->
6 <link th:href="@{css/bootstrap/css/new-age.min.css}" rel="stylesheet"></link>
```

4.2.3.3 Database Entities

The connection to the database was the next step. To do it it was used the annotation `@Entity` in each of my Java classes to tell the JPA that each of these classes is a database JPA Entity. One Java class as a JPA Entity for each table that is wanted to access was created. The Entity to access the `arm_traffic_light` table can be seen at [4.5](#).

The `@Entity` indicates to the JPA that the Java Class is an entity to access the database info. The `@Table` specifies the tables to which the JPA must access. If there is no table annotation, the JPA assumes the name of the Java class as the name of the table from which it will get the data. The `@ID` annotation gives the information that this column is the primary key of the table. Moreover, the `@ID` annotation is also followed by the annotation `@GeneratedValue` to indicate that the ID should be generated automatically. If we want to insert data through our Entities the `@GeneratedValue` is essential, otherwise, the database will have problems with it. The `@NotNull` one ensures that the value cannot be null. Finally, the `@Column` annotation states the table's column to access and then stores it in the instance variable declared right below it.

Now, we have our Java class as a JPA Entity. Moreover, if we want to have access to the values of the objects created with data received we need to declare ordinarily the Java class constructor and the getters and setters for the instance variables. Thereby, we can have access to the data of each object created with the database info. Anytime we access the database we receive a row of the table as one object. So, if the query will return more than one row of the table, we will receive a list of objects from our query, each one as an object from our JPA Entity Java Class. Thus, we can manage the object information through the getters and setters or even other methods created in the Java Class.

Listing 4.5: Java Entity created to the `planet_osm_point` table. (Partially complete here.)

```
1 @Entity
2 @Table(name = "arm_traffic_light")
3 public class Arm_traffic_light {
4     @Id
5     @GeneratedValue
```

Implementation

```
6 @NotNull
7 @Column(name = "traffic_light_id")
8 private int traffic_light_id;
9 @Column(name = "type")
10 private Integer type;
11 @Column(name = "tl_controller_id")
12 private int tl_controller_id;
13 @Column(name = "coordinates")
14 private Point coordinates;
```

4.2.3.4 Repositories

Hereafter, it is necessary to create a Java interface to access the database by queries. Specifying only the Entities is not enough for JPA to access the database. It was created an interface for each Entity Class. Each Java Interface is declared, and it extends the *org.springframework.data.repository.CrudRepository*, as we can see at Listing 4.8. The *@Repository* and *@Transactional* annotations state to use the java Interface as a repository, and also they say that the Interface will be used for database transactions. Thereby, JPA can now use the Java Interface to the database access methods.

Extending the *CrudRepository*, and using the two mentioned annotations, allows us to make queries without explicitly write them. Using the supported keywords, we can create a query using only the name of the instance variables as parameters of our method. Moreover, the name of the method must use the JPA Repositories convention. For instance, we use the *FindBy* word followed by the instance variable name starting with upper case, as we can see in the 4.6 example. Then JPA will translate the method into a query as we can see at Listing 4.7.

Listing 4.6: Using *CrudRepository* to create Queries to the *planet_osm_point* table.

```
1 @Transactional
2 @Repository
3 public interface GisRepository extends CrudRepository<Points, Serializable> {
4     List<Points> findByAmenity(String amenity);
5     List<Points> findByHighway(String highway);}
```

Listing 4.7: Query generated by JPA.

```
1 SELECT * FROM Points p where p.amenity = ?1
```

However, we can also specify the queries using the *@Query* annotation, as we can see in the 4.8 example. Using this annotation, we simply write the query we want to do and then we must specify the method we want to call to execute the query. The parameters of our method are related to the query ones through the *?1*, *?2*, *?3*, etc, parameters specified in the query. If the query has only one parameter, we use the *?1* in the place we want to use the value of the method parameter. If the query has more than one parameter, we use *?2*, *?3*, etc, accordingly to the order of the method parameters.

Implementation

Listing 4.8: Java Repository Interface for the *arm_traffic_light* table.

```
1 @Transactional
2 @Repository
3 public interface Arm_traffic_light_repository extends CrudRepository<
    Arm_traffic_light, Serializable> {
4     @Query(value = "SELECT * FROM arm_traffic_light WHERE ST_Twisting(
        coordinates, ST_SetSRID(ST_Point(?,?), 4326), 0.0003)", nativeQuery =
        true)
5     List<Arm_traffic_light> findInfraAround(double coordx, double coordy);}
```

The ARIMaRN application needs to receive information about the infrastructures around one coordinate. Thereby, when the user is walking, the app is asking for the nearby infrastructures. To fulfil this request, it was developed a query to gather the data from the database where its coordinates are inside a sphere with about 30 meters of radius, around the user. Thereby, the query is performed for all the tables with the GIS data. The 4.8 listing shows us the query to get the info about the traffic signals around the user. More two similar queries were developed to get the info about the crosswalks and the electromagnetic coils around. Furthermore, as it was extended the *CrudRepository* in every one of my repositories, we can make queries for each column of the tables.

4.2.3.5 Controllers

The database access is ready after we develop the JPA Entities and Repositories. To send the data from the server, we need to create the controllers. The *@RestController* annotation specifies to the Spring Boot framework the Java Class that will be a rest controller. Thereby, the Spring Boot framework automatically allows us to handle JSON without more indications. To create a GET request, we need the *@RequestMapping* annotation alongside with the parameters we want. An example of one of my available GET requests can be seen at 4.9. Here, we specify the URL path in the value parameter. Furthermore, we need to state which request method we want to use. As we want to share information and not store it, we only created GET requests to my server. Then, right below the *@RequestMapping* annotation, we define the Java function that calls the data we need to return in the URL path. In the function declaration we use the *@RequestParam* annotation to identify the name to use in the URL, that gives the parameter value, and also its type declaration. As we can see in 4.9 the parameter *id* is defined as an integer and the name in the URL will be the "*id*" string declared there too. Thus, to access this URL successfully, we need to write the path: */gis/find/gis-crosswalks?id=1*, after the server IP and port.

Listing 4.9: GET request example.

```
1 @RequestMapping(value="/gis/find/gis-crosswalks", method = RequestMethod.GET)
2 public Object findByGisCrosswalks(@RequestParam("id") int id) {
3     GisCrosswalks object = crosswalkRepo.findOne(id);
4     if(object == null) {
5         return "null";
```

Implementation

```
6     }
7     DTO dtoObject = new DTO();
8     dtoObject.setGis_crosswalks_id(object.getGis_crosswalks_id());
9     dtoObject.setGis_crosswalks_intersection_id(object.getIntersection_id());
10    CoordinateSequence pos = object.getCrosswalk_coordinates().
        getCoordinateSequence();
11    dtoObject.setGis_crosswalk_coordinatesX(pos.getCoordinate(0).x);
12    dtoObject.setGis_crosswalk_coordinatesY(pos.getCoordinate(0).y);
13    return dtoObject;}
```

4.2.3.6 Data Transfer Object

The DTO Class is the Class of the Data Transfer Object. Here, the Java Class defines the instance variables to build an object with information about all the possible data to send in a request. All the columns of the ARIMaRN database tables have a corresponding instance variable. The DTO object is created and then filled with some given data. As each query just gives information about a table, the DTO object instance variables related to the table stores the data from the query 4.9. The other ones remain with the default values given when we call the DTO constructor, as we can see at Listing 4.10. Thereby, when someone sends a GET request to the server, they always see the same object structure. However, according to the GET request, they will receive the data in the respective object variables. Thus, the external requests do not have direct access to the database Entities, giving one layer of security to the server.

Listing 4.10: Empty DTO JSON

```
1 { "osm_id":0, "amenity":null, "highway":null, "tags":null, "wayX":0.0, "wayY":0.0, "
    tl_Controller_controller_id":0, "cmp_longersection_id":0, "tl_group_id":0, "
    tl_Group_controller_id":0, "group_value":0, "tl_plan_id":0, "
    tl_Plan_controller_id":0, "plan_value":0, "duration":0, "tl_step_group_id":0, "
    tl_step_id":0, "tl_Step_Group_group_id":0, "tl_Step_step_id":0, "
    tl_Step_controller_id":0, "step_value":0, "max_step_time":0, "traffic_light_id"
    ":0, "feu":0, "tys":0, "type":null, "tl_Traffic_controller_id":0, "coordinatesX"
    ":0.0, "coordinatesY":0.0, "gis_crosswalks_id":0, "
    gis_crosswalks_intersection_id":0, "gis_crosswalk_coordinatesX":0.0, "
    gis_crosswalk_coordinatesY":0.0, "bim_crosswalks_id":0, "
    bim_crosswalks_intersection_id":0, "bim_crosswalks_material_id":0, "
    bim_intersection_id":0, "bim_intersection_description":null, "gis_espiras_id"
    ":0, "gis_espiras_intersection_id":0, "gis_espiras_coordinatesX":0.0, "
    gis_espiras_coordinatesY":0.0, "bim_espiras_id":0, "
    bim_espiras_intersection_id":0, "bim_espiras_material_id":0, "
    bim_espiras_tipologia":null}
```

4.2.3.7 Home Page Controller

The *@RestController* annotation does not allow us to use HTML and the correspondent static files. Therefore, it was created the *HomePageController* Java Class to handle the homepage HTML file

Implementation

and its dependencies. Here, it was used the `@Controller` annotation instead. The annotation tells the Spring Boot Framework that the Java Class is a web controller, instead of and RESTful one. Thus, now we can handle the HTML template files using the GET request option of the `@RequestMapping` annotation. Then we just need to *return* the HTML file name, without its extension, and the web-page is fully working. We can saw the controller at [4.11](#).

Listing 4.11: Homepage Controller

```
1 @Controller
2 @Component
3 public class HomeController {
4     @RequestMapping(value = "/", method = RequestMethod.GET)
5     public String homepage() {
6         return "homepage";}
```

4.2.3.8 Main Application

After all the coding, without the main class, we cannot run the server. To identify it we use the `@SpringBootApplication` annotation. Therefore, the Spring Boot framework knows which is the main class. The framework searches as default for other Spring Boot components in the package where we have the main application. As it was divided the Entities, Repositories, and Controllers into different packages out of the package where the main application is, we need to give instructions to the framework to search the other packages for our Entities, Repositories and Controllers. The Java classes and Interfaces were divided into packages by the data that they access. Thereby, the BIM entities, Interfaces and the Controller are in the `bim` and `bimDao` packages. The GIS ones are in the `gis` and `dao` packages. Moreover, the main package has the DTO Java class, the Homepage Controller and the main application.

As we can saw at [4.12](#), we have more three different annotations. The `@EntityScan` one is used to tell Spring Boot to search for the Entities in the declared packages. The `@EnableJpaRepositories` tells to scan for the Repositories. Finally, the `@ComponentScan` scans for the `@Component` annotation. This annotation is used in the Controllers Java Classes so that Spring Boot could find them, as we can see at [Listing 4.11](#).

Listing 4.12: Main Application

```
1 @SpringBootApplication
2 @EntityScan(basePackages = {"gis", "bim", "main", "dao"}) // to scan for
   @entity in the specified packages
3 @ComponentScan(basePackages = {"gis", "bim", "main", "dao" } ) // to scan for
   @component in the specified packages
4 @EnableJpaRepositories({"dao", "bimDao"}) // to scan for @repository in the
   specified packages
5 public class Application{
6     public static void main(String[] args) throws Exception{
7         ApplicationContext ctx = SpringApplication.run(Application.class, args);}
```


Implementation

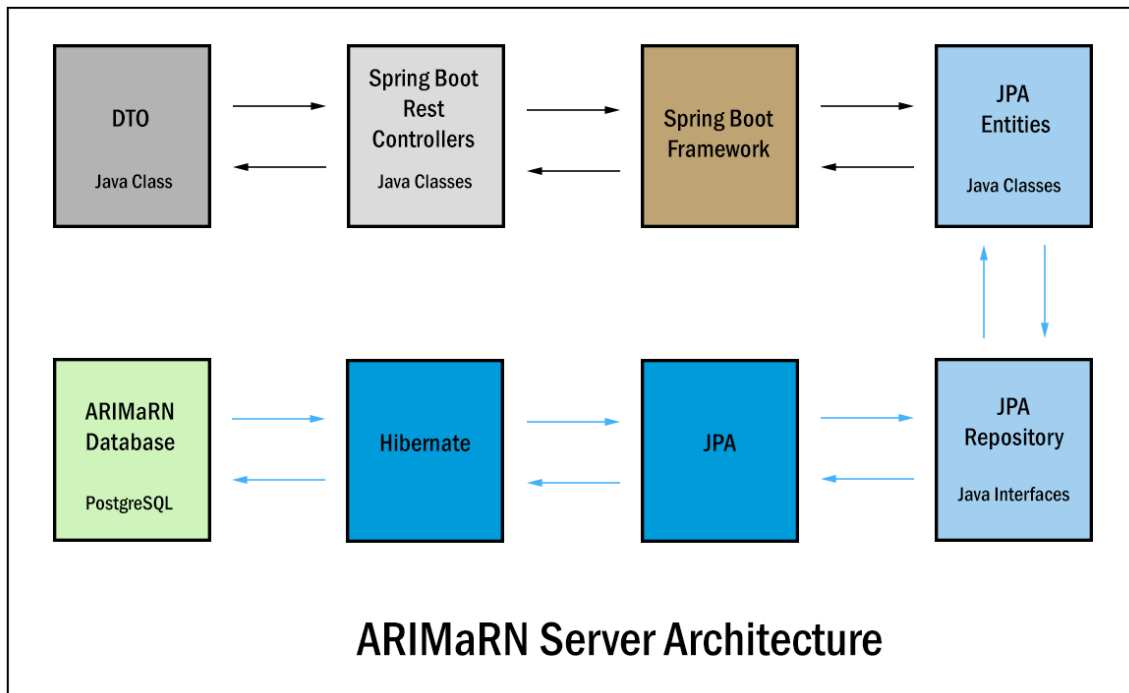


Figure 4.5: ARIMaRN Server Architecture.

4.2.3.9 Server Architecture

As we can see at Figure 4.5, the ARIMaRN server uses the JPA Java Interfaces Repositories to communicate with JPA and Hibernate frameworks. They aim to withdraw the data corresponding to the requested queries. Then, the JPA Repositories use the JPA Entities Java Classes to store the table rows, of the resulting queries, as a JPA Entity object. The access to the database is triggered by the Rest Controllers that tell Spring Boot framework to do it when they receive a request. After the gathering of the database data, the Rest Controllers set the data to send as DTOs objects and return them in the URL as a JSON array.

4.2.4 AR Integration

To integrate the VR/AR technologies with BIM GIS and IoT, it was tried several approaches. The AR SDKs have devices restrictions that influence the use of one of them to the detriment of the other. See also A.5.

Vuforia SDK was used to build the virtual objects and overlap them on the real scene. As Vuforia comes now integrated into the Unity engine, we do not need to import any external package. Therefore, we can insert the AR Camera game object in place of the *Main Camera* that comes as default in Unity. The *AR Camera* has some settings that we need to change in the Inspector tab of Unity. There, we need to press the *Open Vuforia Configuration* button. Here, we set our *Licence Key* to use the Vuforia SDK. As we do not need to use the fiducial marks, we can deselect

Implementation

all the *Databases*. In the *Device Tracker* dropdown, we select the *Track Device Pose* option. Then we change the *Tracking Mode* to *ROTATIONAL* and the *Model Correction Mode* to *HANDHELD*. Thereby, the AR Camera uses the orientation of the mobile device, and if we look around in the real world with our device, the virtual camera also rotates in the same direction.

The GIS-BIM database that was built has information about the traffic lights, crosswalks and electromagnetic coils. However, that information is only the semantic one. Thus, the geometric representation of the infrastructures in the database needs to be built. Therefore, it was required to develop the virtual objects representing them. As the traffic lights have two main types, were searched and found two different free models that we can use in Unity. One model of the standard traffic lights that stand in the margin of the road [Bru17] and one model for the light signals that start in the margin and cross until the middle of the road[Tur12].

After the download of the models, they were imported to Unity. Then, they were added as the game objects of the *BuildScene* script. As the traffic lights have more sensors and more information about them in the database, it was given more emphasis on them by using a better representing model. Furthermore, it was built a cube representation of the crosswalks and the electromagnetic coils. The crosswalk's cubes have the green colour, and the electromagnetic coil's cubes have the yellow one. As with the traffic lights models, the crosswalk's and the electromagnetic coil's models were added as game objects in my *BuildScene* script.

4.2.4.0.1 *BuildScene* Script

The *BuildScene* script is the responsible for the creation or the destruction of the virtual objects. Using the *Update()* function of Unity scripts it is verified in every frame if the objects could be created, destroyed, or kept. This verification is according to the distance between the user and the virtual object. As the GPS have associated errors, the closer we are to the virtual object higher will be the visualisation of the error. If we are more distant from the virtual object, the angle of deviation is smaller and gives us a more accurate viewing, as we can see at Figure 4.6.

According to that, it was created two thresholds, the maximum and the minimum of distance between the user's position and the virtual object. Thereby, it was chosen 5 meters as the minimum distance to see the virtual object and 20 meters as the maximum. The *BuildScene* script is consequently based on these two thresholds to build the virtual objects. The Script loop through the list of received infrastructures from the server. If the distance from the user to the virtual object is inside the range of the two limits, the virtual object is created according to its type (standard traffic light, traffic light that crosses half the road, electromagnetic coil, or crosswalk). If the distance is out of the range, the virtual object is not built. Moreover, if the distance is out of the range but the virtual object representing that infrastructure has been already created it will be destroyed. To a better understanding see Figure 4.7.

On the traffic lights case, it was also used the orientation received from the server to rotate the model according to its real direction. Thus, the traffic lights models are oriented in the virtual world as in the real one.

Implementation

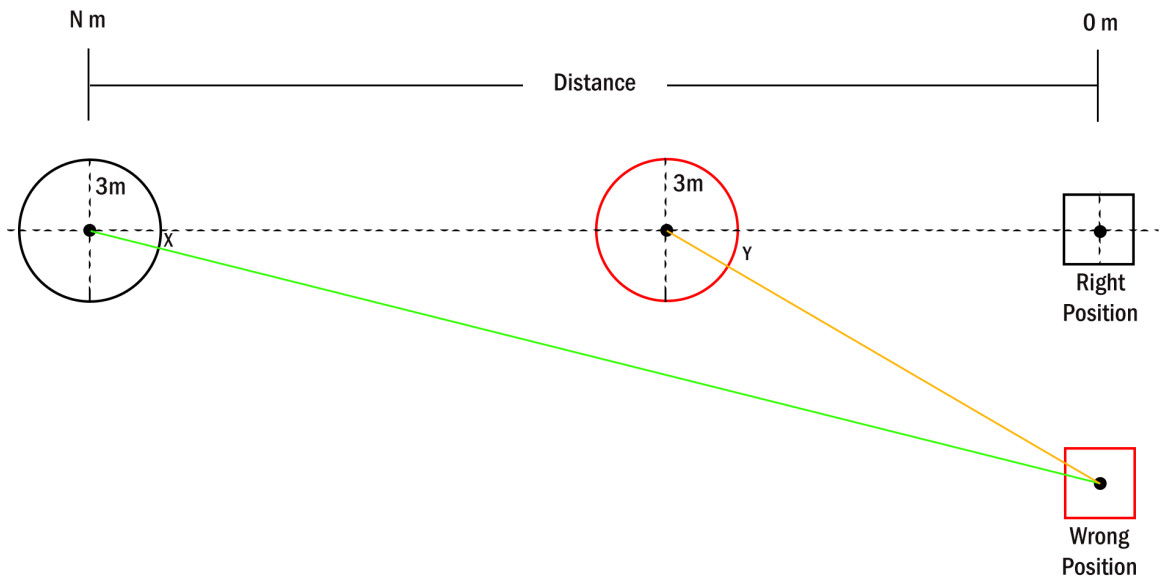


Figure 4.6: Viewing of the Angle's deviation according to the GPS Associated Error.

As we can see, it is demanding that the GPS readings are the most accurate possible. Otherwise, the resulting distance can lead the algorithm to build the wrong virtual objects. Thus, the location service is essential in my application. If it receives GPS readings too far from the real place of the user, the virtual scene will not be according to the expected.

4.2.4.0.2 *AugmentedScript* Script

The infrastructure models have an associated script to them so that when we press an infrastructure in the application, we can see a panel with the BIM data about the pressed infrastructure. The *AugmentedScript* script uses the *OnMouseDown()* function to the Unity application recognise the touch of the user on the infrastructure's virtual object. This script has an associated Unity canvas for each type of infrastructure to display the BIM data. The traffic light canvas' background is blue, the crosswalk's one is green, and the electromagnetic coil's one is yellow.

When the user presses one virtual object, the *AugmentedScript* script executes a request to the server according to the touched infrastructure type. Then the canvas is displayed with the text: "*Waiting For BIM Data. Connecting to Server..*". If the server returns the BIM data correctly, the canvas text is updated with all the BIM data regarding that infrastructure. If the server does not give us anything, the canvas is destroyed after 5 seconds, and the user can use the app again. If we have an error as the return of the server the canvas is immediately destroyed. To a better understanding see Figure 4.7.

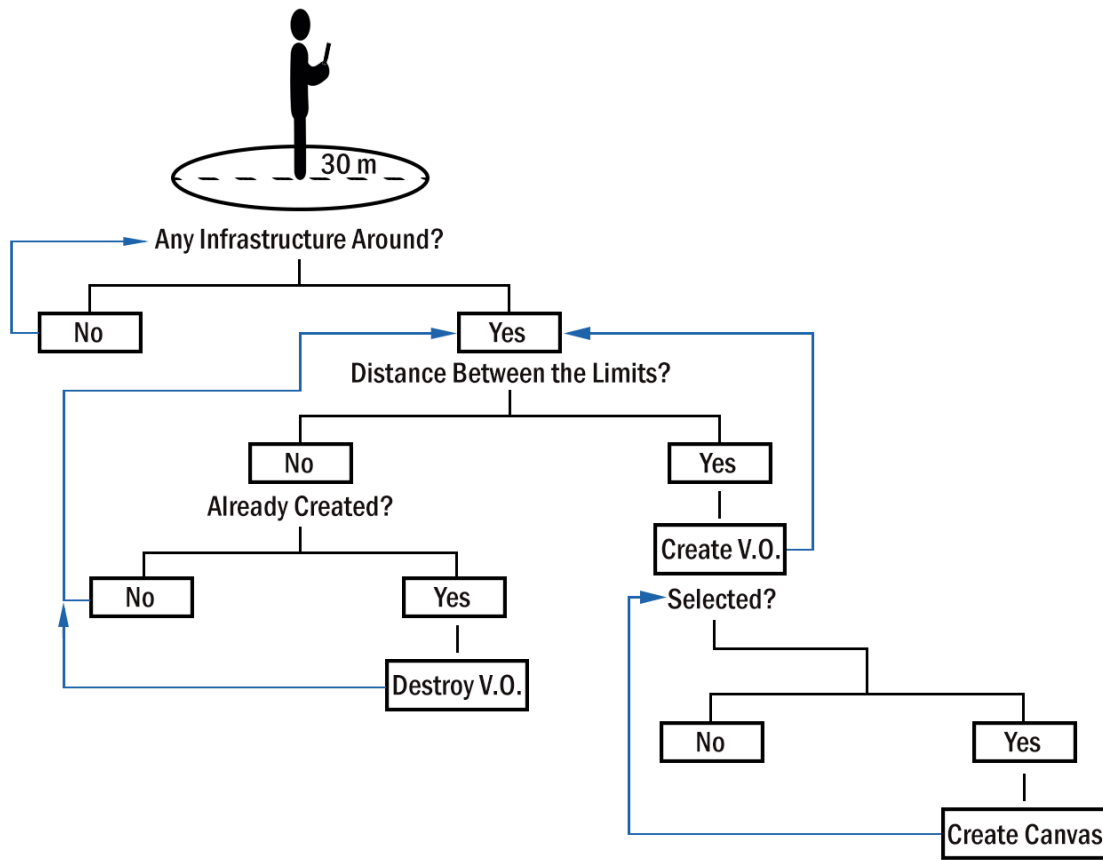


Figure 4.7: ARIMaRN Main Application.

4.2.4.1 Server Connection

To acquire the infrastructures around the user, it was developed the *ServerConnection* script. The script has functions to make the server requests according to the necessary information. Each function uses the current GPS reading to request the server for the infrastructures around. As we know, the queries performed in the server return the infrastructures in a radius of about 30 meters around the input coordinates. Thus, when the *ServerConnection* script sends one request to the server, it returns the infrastructures within a 30 meters radius, according to the infrastructure type requested.

The script calls the functions to get the traffic lights, crosswalks and the electromagnetic coils on every 10 seconds. Then, according to the server data, it saves the infrastructures.

The functions to get the different types of infrastructures are based on the same principle. First, according to the type of the infrastructure, we make the connection to the respective server URL, giving the necessary parameters. If it returns the GIS data, it was used the *JsonHandler* Script to translate the JSON object from the server to an object of my script. The *JsonHandler* Script has two classes. The first one deals with the JSON string from the server and transforms it into an array of the type of the *ServerObject* class. The second one is the *ServerObject* class where it was

Implementation

declared the instance variables needed to store the data received from the server. Thus, using these two class methods, we can transform the JSON from the server into usable objects.

After the translation of the data into Server Objects, the function calls the *AddOrNot()* function to verify if we already received that infrastructure and stored it. If we do not have that object already, the *canAdd boolean* becomes *true*, and the object is added to the list of received objects of the BuildScene Script. The method is the same on the three different types of request. Thereby, if we already have the information about that infrastructure, as it is no need to store it again, the application use the last record of it. From now on the virtual scene is built according to the BuildScene script thresholds.

4.2.4.2 Location Service Approach

The Vuforia SDK only give us the AR capabilities. Hence, it was needed to develop a location service to fulfil the application requirements. To do it, there were developed three different scripts: *GPS*, *LocationHandler*, and *UtilsMath*. The location functions are divided among the three scripts to a better structure of the code. They are developed based on the Unity GPS abilities and some mathematics formulas.

4.2.4.2.1 GPS Script

The GPS script is responsible for acquiring the GPS readings and filtering them to return the best accurate values possible. In the ARIMaRN application, it is necessary the use of precise GPS readings. If we have latitude and longitude values far from the real user position, the virtual scene is created based on wrong assumptions. Thereby, the accumulated error is even more significant, resulting in the placing of the virtual objects in the wrong position.

As we saw in the 3.6.1, the GPS readings still have accuracy problems, especially on the mobile devices. On the One Plus 3, the best accuracy values found in an open space, without obstacles, were about 3 meters in both vertical and horizontal accuracy. However, the trees and the buildings quickly change these values to 10 or more meters. Thus, we are forced to develop an algorithm for filtering the estimated wrong GPS readings. For this reason, it was used the Moving Window approach as the H. Song and J. Lee used in their study [SL15].

My Moving Window algorithm uses the distance between the last GPS reading and the new one to calculate the speed and the acceleration. When we start the application, the first GPS readings always start very far from the correct position of the user. Thereby, the first five different readings are filtered according to the distance. If the distance between the last and the new GPS reading if more than 10 meters the last GPS reading is discarded and the new one is used. Moreover, the first ten different GPS readings are used to build the Moving Window average values such as the speed average, acceleration average, distance average, latitude and longitude variation average, and the *timestamp* average.

As we saw at 3.6.1, the Maximum Acceleration value of a runner is $3.09m/s^2$, and the Maximum Speed value for Ambulation is $3m/s$. According to this values, it was defined the maximum

Implementation

speed and acceleration values on my algorithm. If the resulting speed or acceleration values are more than the thresholds, the GPS reading is discarded and it is used the average between the last and the new GPS readings to give a more approximated value. The same is calculated for the timestamp value to maintain the time in accordance to the GPS readings. Then, the average latitude and longitude values are used as the position of the user.

In the same line of thinking, it was defined the minimum speed and acceleration values to a person in motion. If the speed of the instant t_1 is 0.5 m/s and in the instant t_2 it is, for instance, 1m/s, both below the average, and the time difference is 2 seconds, we have an acceleration of $0.25m/s^2$. Therefore we can assume the minimum acceleration as $0.25m/s^2$ as our threshold. As we saw at 3.6.1, we can use the 0.5m/s value as the minimum admissible speed value. If the resulting values are less than these thresholds is assumed that the user is fixed at one position. Therefore, the GPS reading is again the last one since the user did not move. The speed and acceleration averages are also constraints in the algorithm. If the speed value is less than the maximum threshold value but higher than four times the speed average value, the algorithm discards the new GPS reading and uses the average between it and the last one recorded. The same process is applied to the acceleration values. Therefore, the system performs with more smooth values. Otherwise, sudden movements of the users will have undesirable effects on the AR experience, such as shaking virtual objects.

If the new GPS reading passes the distance, speed and acceleration constraints, the algorithm updates the average values using the new reading and the latitude and longitude values are used as the position of the user.

The *GPS* script also updates the camera position according to the Moving Window algorithm resulting GPS value. To translate the latitude and longitude values into the three dimensions' Cartesian coordinate system, it was used a formula developed for me, declared on the Location-Handler script 4.2.4.2.2. This formula is also used for the translation of the infrastructures' position.

Another essential feature of the GPS script is the rotation of the camera according to the compass value. The compass value is based on the geographic north pole since the magnetic one is always changing its location over the years. The latitude and longitude mapping into the Cartesian dimensions is performed according to the north orientation. Thus, the objects are built oriented to the Cartesian representing of the north. To ensure that the user can rotate their device's camera, and consequently the virtual camera, and the virtual objects still have the correct orientation the mobile's compass value is used. When the application starts, it is calculated the average value of the compass on ten readings. Then the virtual camera, initially oriented to the north, is rotated to the same orientation of the real camera using the average compass value. Thereby, the user can turn its device in any direction, and the AR camera rotates in accordance to it because it was chosen the *ROTATIONAL* setting on it. This *ROTATIONAL* option uses the gyroscope of the device to rotate the virtual camera. Thus, the virtual objects appear in the right place although they are always built according to the north orientation, as we can see at Figure 4.8.

Implementation

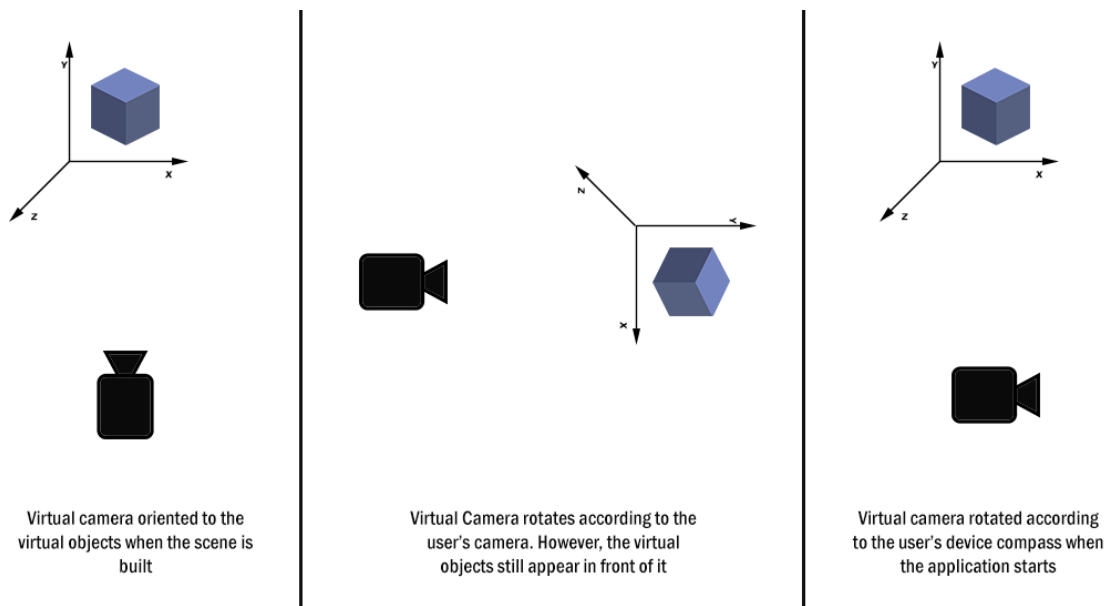


Figure 4.8: Virtual Camera Rotation According to the Mobile Device's Compass.

4.2.4.2.2 *LocationHandler* Script

The *LocationHandler* script has the functions to calculate distances using the Haversine Formula and also the coordinates conversion functions to the Unity's Cartesian referential. The distance functions are used to return the distance to a infrastructure to the user's position. Thereby, we can use the distance function in the BuildScene script to evaluate if the virtual object is drawn or if it is necessary to destroy it. The other distance function is used in the Moving Window algorithm to calculate the distance between two GPS readings, also based on the Haversine Formula.

The other two available functions in this script are related to the mapping of the GPS coordinates to the Unity's Cartesian referential. As we know, if we have a short distance on the earth surface we can discard the earth's radius on the mapping calculus. Therefore, we can draw a square within which we have the places we want to map and find a constant to use. If we choose the middle point of the latitude and longitude values on the square's edge, as we can see at Figure 4.9, we can find a constant that gives us the maximum precision in the middle of the square and trustful approximate values inside the square. Nevertheless, this is only possible with short distances regarding the earth curvature. Otherwise, we need to use the earth's radius in the equation. Using the Unity engine unities, where it was chosen that one meter of the real world corresponds to 0.1 unity of the Unity engine, we can find the same distance in the virtual world, making the mapping possible.

Calculating the latitude and longitude differences between the new coordinate and the C known point, we know that if this difference value is multiplied by one constant, it is equal to the distance between the two coordinates points, as we can see in the equations 4.1 and 4.3. Thereby, the latitude and longitude constants are the distance between the two points multiplied by the latitude

Implementation

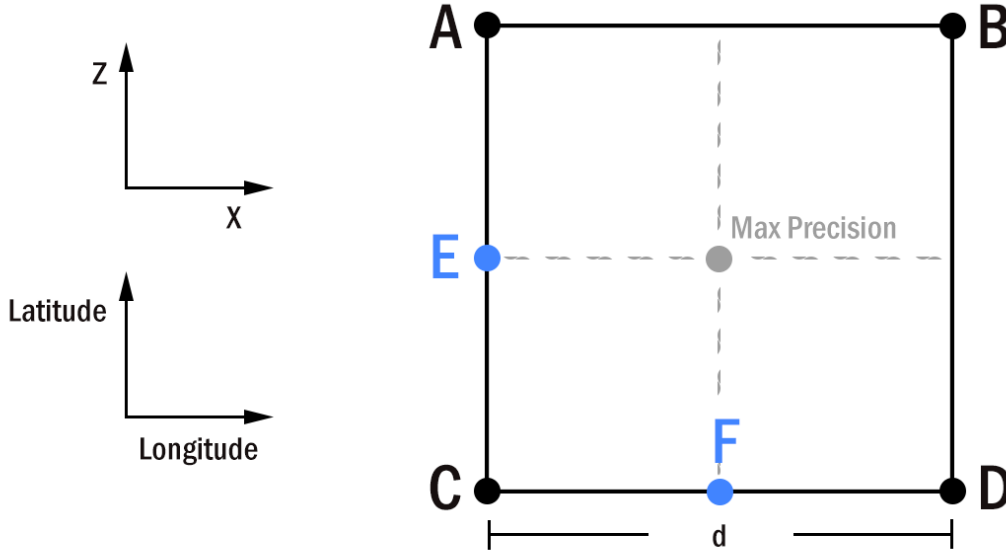


Figure 4.9: Mapping the Real Coordinates to the Unity world.

or longitude difference, as we can see in the equations 4.2 and 4.4.

Thus, we created two functions based on the present equations to map the latitude and longitude values to the Unity representation. The virtual objects and the AR Camera are placed according to this mapping. Thereby, the virtual objects can appear at the right place.

The altitude value of the GPS readings is not used to map into the yy axis because the value has an associated error that can lead to a poor AR experience. The most common altitude error is at least 10 meters. Therefore, the virtual objects would appear above the height of our heads. With this issue in mind, it was chosen to use a standard value to the yy axis on the Unity engine. The virtual camera is placed with the Y value of 2.8 and the virtual objects with -18. These values take into account the virtual camera frustum to bring a realistic experience to the user. The size of the virtual objects also influences the Y value of themselves and the virtual camera directly. Thereby, after some tests, the values that have a better fit are the mentioned above.

$$\Delta_{Longitude} = \|P_{C_{Longitude}} - P_{X_{Longitude}}\| \Leftrightarrow \Delta_{Longitude} * K_{Longitude} = \frac{d}{2} \quad (4.1)$$

$$\Leftrightarrow K_{Longitude} = \left(\frac{d}{2}\right) / \Delta_{Longitude} \quad (4.2)$$

$$\Delta_{Latitude} = \|P_{C_{Latitude}} - P_{Y_{Latitude}}\| \Leftrightarrow \Delta_{Latitude} * K_{Latitude} = \frac{d}{2} \quad (4.3)$$

Implementation

$$\langle \Rightarrow \rangle K_{Latitude} = \left(\frac{d}{2}\right) / \Delta_{Latitude} \quad (4.4)$$

4.2.4.2.3 *UtilsMath* Script

The *UtilsMath* script has two functions on it. One is the Haversine Formula, and the other allows to create a file with the logs of the Moving Window Algorithm to a better debugging of it. The Haversine Formula function is used by the *LocationHandler* functions that return the distances.

4.3 Summary

The ARIMaRN-PSAL application can be tested without more work on it. We already have a model of road signage to test and analyse. On the other side, the ARIMaRN app needs database data to work successfully. The app connects with the server to get information regarding the infrastructures around the user's location. Thus, it is necessary to fill the database with data related to the road infrastructures.

At this point, we can point out the achieved success of doing the integration of the AR, BIM, and GIS technologies. However, the application does not include the IoT technology for now. Nevertheless, the IoT system functionality on the project is to update the database with real-time data. Thus, the ARIMaRN application will perform the same way when the IoT technology is updating the database. The main difference is the fact of the data is up to date. Therefore, we can say the app can be able to integrate the four technologies, even though it is now using only three of them. Thus, the chosen tools and the developed algorithms for this integration had shown being capable of overcoming the interoperability lacks and issues between the technologies.

Implementation

Chapter 5

Case Study

The ARIMaRN application aims to be the concept of proof of the integration between the BIM, GIS, IoT and AR/VR technologies. To test the reliability of the developed system was chosen a road junction. Thereby, the road junction selected is in a near location so that it was possible to test the ARIMaRN application quickly. Due to that, was chosen the road junction in the Rua Nossa Senhora de Fátima of Porto (RNSFP).

5.1 Database

Although the PDB has information about the RNSFP crossroad, some data was inserted, and other ones were updated so that the information is complete.

5.1.1 BIM Database

The BIM data concerning to the *bim_intersection*, *bim_crosswalks*, *bim_materials*, and *bim_espiras* tables were inserted based on the road junction characteristics. The *bim_materials* table was filled with the same data that we can find on the Revit materials table. Furthermore, a specific material regarding the crosswalk material was inserted. On the *bim_intersection* table was added the data of the RNSFP intersection. The id, according to the Porto City Hall data, and the description that is the name of the road joint: "Cruzamento Rua de Nossa Senhora de Fátima". The *bim_crosswalks* table was filled with the four crosswalks that there are on the road, with the intersection identifier 1. The same was performed regarding the *bim_espiras* table. The electromagnetic coils were inserted according to the AutoCAD document, given by the Porto City Hall, about the RNSFP junction, as we can see at Figure 5.1. For a more in-depth analysis and a better understanding we can saw the BIM tables at A.3.1.

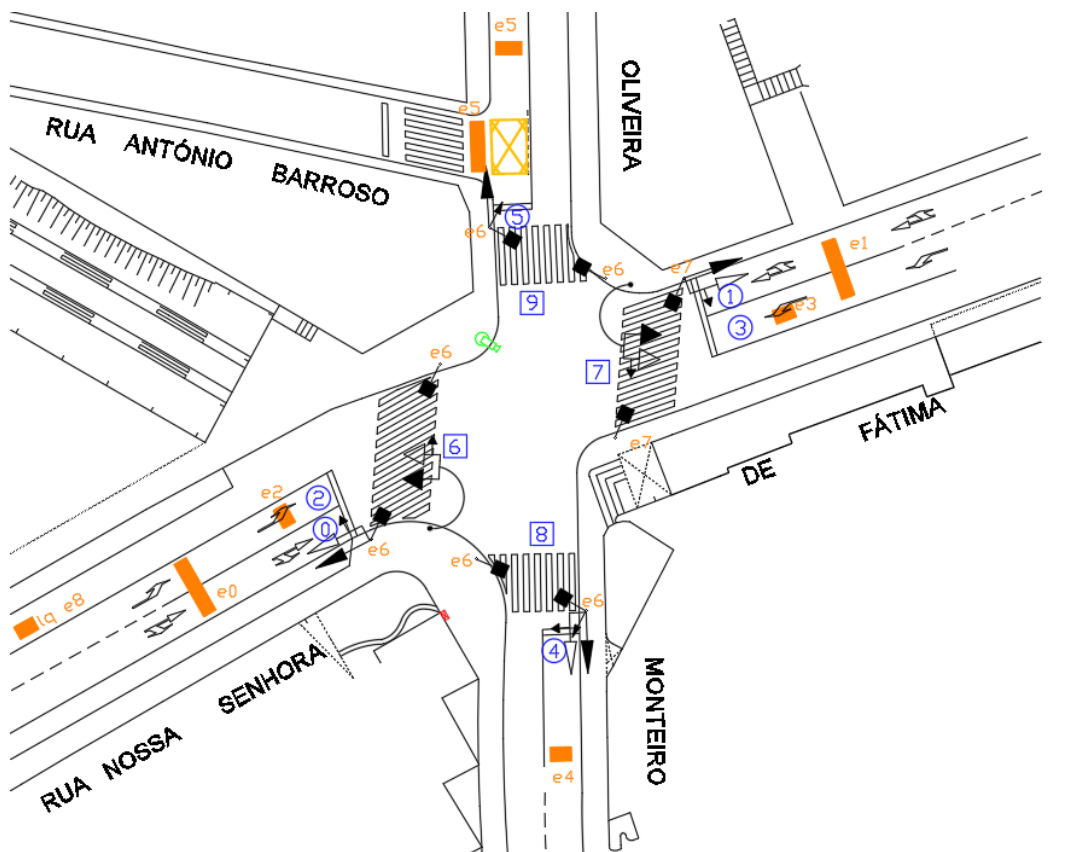


Figure 5.1: RNSFP's AutoCAD File.

5.1.2 GIS Database

The GIS tables already had data. Therefore, some updates were performed to make available the necessary data for the ARIMaRN application. In the *arm_traffic_light* table, it was added the coordinates and the orientation of the traffic lights belonging to the RNSFP junction. The coordinates were based on the Google Maps satellite view. Then it was selected the place where the traffic light was and saved that coordinates. The orientation of the traffic lights was based on the AutoCAD file about the RNSFP junction. However, as the file do not have the angles of the direction, it was drawn a vertical line and calculated the angles between the traffic light orientations and the vertical line (north oriented). This calculus was performed using the AutoCAD *Dimension* tool, as we can see at Figure 5.2.

The *gis_espiras* and the *gis_crosswalks* tables were also updated with the data according to the values found on Google maps, such as on the traffic lights case.

5.1.3 Coordinates' Mapping

To map the infrastructures location to the Unity engine's virtual world's position there was used the approach detailed at 4.2.4.2.2. There was chosen the four vertices of the square so that the

Case Study

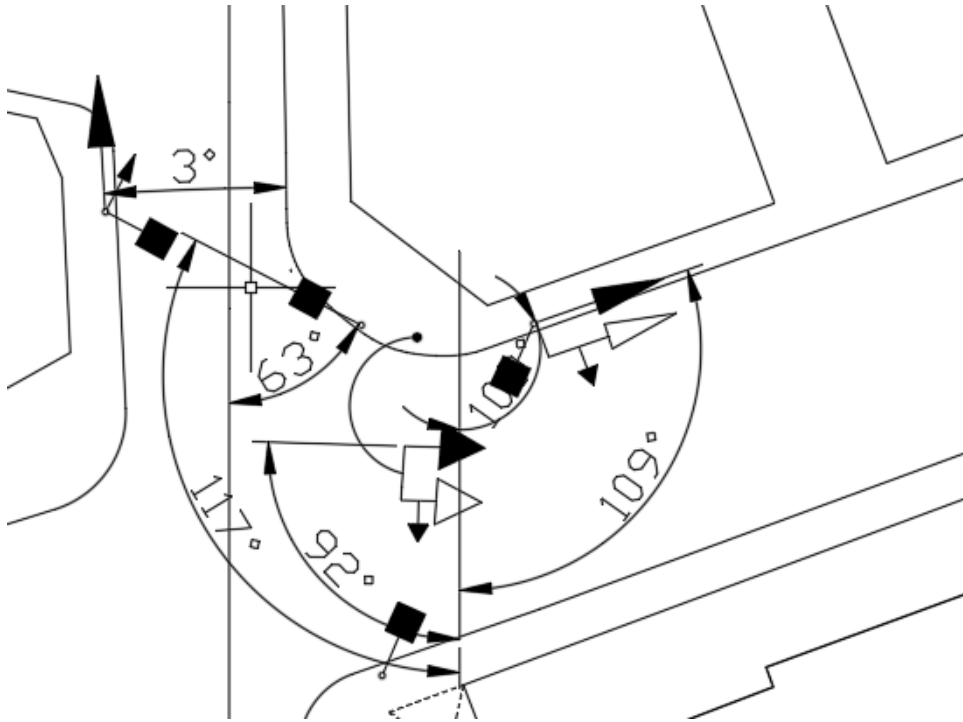


Figure 5.2: AutoCAD *Dimension* Tool Orientation Angles of RNSFP.

RNSFP junction is within it. The four square's points and the two middle points of it are listed below with their coordinates. The distances from the *C* point to the *E* and *F* points are also present below.

$$P_A(41.160225, -8.625065),$$

$$P_B(41.160225, -8.623970),$$

$$P_C(41.159400, -8.625065),$$

$$P_D(41.159400, -8.623970),$$

$$P_E(41.159813, -8.625065).$$

$$P_F(41.159400, -8.624518),$$

$$|CF| = 45.8m$$

$$|CE| = 45.86m$$

Then, using the middle points of the longitude and latitude was calculated the constant for each one that allows the mapping into Unity engine unities as we can see in the following equations, 5.1 to 5.8, and at the 5.3 image.

Case Study

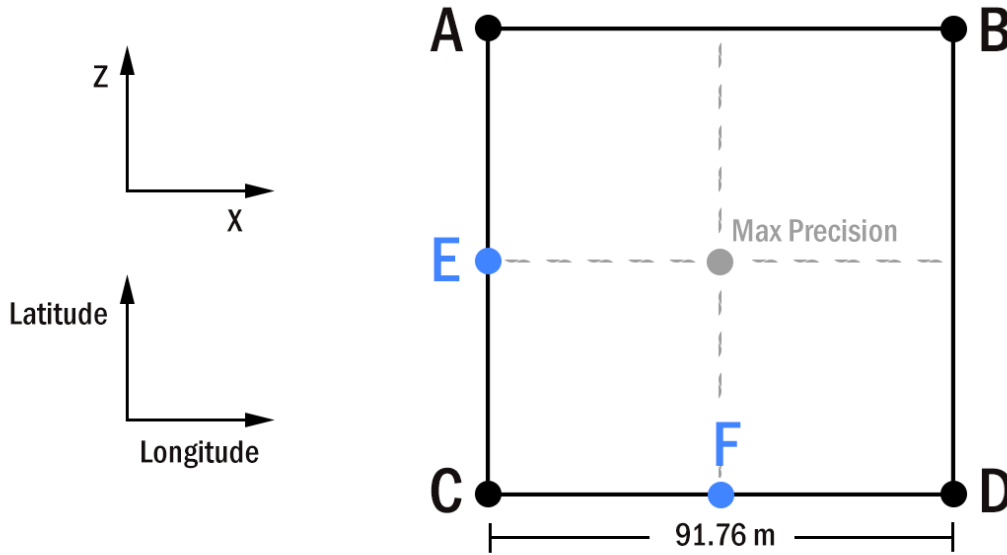


Figure 5.3: Mapping the Real Coordinates to the Unity World - Case of Study Values.

$$\Delta_{Longitude} = \|P_{C_{Longitude}} - P_{X_{Longitude}}\| = \|8.625065 - 8.624518\| \quad (5.1)$$

$$\Delta_{Longitude} = 0.000547 \quad (5.2)$$

$$\Delta_{Longitude} * K_{Longitude} = 45.8 \Leftrightarrow K_{Longitude} = \frac{45.8}{\Delta_{Longitude}} \quad (5.3)$$

$$K_{Longitude} = 83729.433272 \quad (5.4)$$

$$\Delta_{Latitude} = \|P_{C_{Latitude}} - P_{Y_{Latitude}}\| = \|41.159400 - 41.159813\| \quad (5.5)$$

$$\Delta_{Latitude} = 0.000413 \quad (5.6)$$

$$\Delta_{Latitude} * K_{Latitude} = 45.86 \Leftrightarrow K_{Latitude} = \frac{45.86}{\Delta_{Latitude}} \quad (5.7)$$

$$K_{Latitude} = 111041.1622276 \quad (5.8)$$

Therefore, the C# functions related to the location mapping were updated with the C point's latitude and longitude values, and with the latitude and longitude constants found in the last equations, 5.1 to 5.8, as we can see at the 5.1 listing.

Case Study

Listing 5.1: C# Longitude and Latitude Mapping - Case of Study

```
1 public float LongitudeToX(double lon) {
2     float deltaLongitude = Mathf.Abs(((float)lon - (-8.625065f))) * 83729.44f;
3     return deltaLongitude * 10f;
4
5 public float LatitudeToZ(double lat) {
6     float deltaLatitude = Mathf.Abs(((float)lat - 41.159400f)) * 111041.16f;
7     return deltaLatitude * 10f;
8 }
```

5.2 Results

5.2.1 ARIMaRN-PSAL

The ARIMaRN-PSAL application allows to place virtual signage at a location chosen by the user. Therefore, the user can evaluate if the real signage could be placed there according to the signage restrictions.

As we saw at 4.1.4, my One Plus 3 cannot be able to run the Instant Tracking required on this application. However, we can test the app using the Unity engine on a computer and a fiducial mark provided by the Vuforia SDK. The tracking automatically recognises the fiducial mark and uses it as if it is the plane surface that the instant tracking identified. Thereby, using this method, it was tested the application. When we point the camera to the fiducial mark and click on the screen (mouse click) the traffic light's model appears on the screen and, consequently, we can analyse it. The user can get closer and far from the object and walk around it, evaluating if that position is the optimal one, as we can see at Figure 5.4.

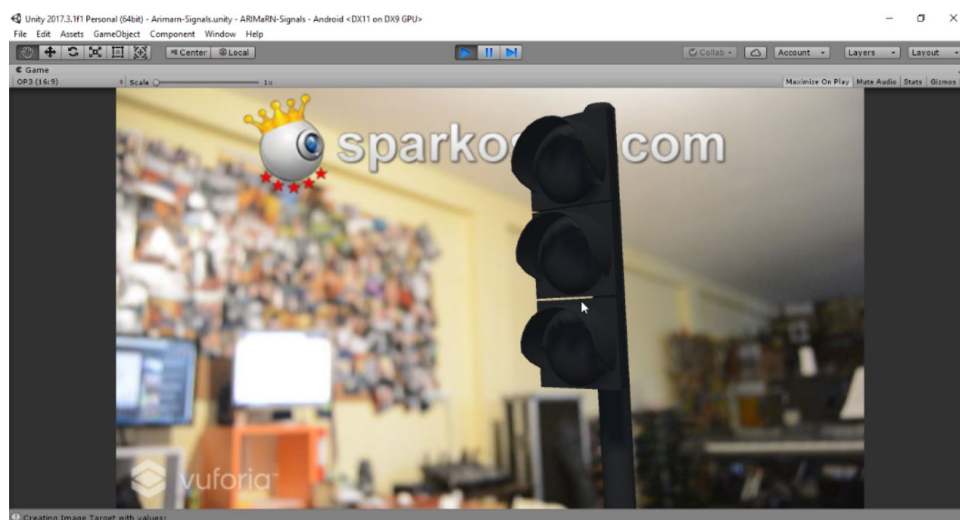


Figure 5.4: Print of the ARIMaRN-PSAL simulation using the fiducial mark on the ground.

Case Study

5.2.2 ARIMaRN

5.2.2.1 Using Mock Locations

Using the *Fake GPS* Android application to mock the location of the mobile device, there was made some tests to the ARIMaRN application.

The first one was if the application receives the info correctly from the server. Using the *Log-Cat* tab in the *Monitor* tool of the Android SDK is possible to see the logs of the Unity application by using a filter. Thereby, it was run the application, and the Unity's logs were analysed to verify if the server connection returned the correct info.

L...	Time	PID	TID	Application	Tag	Text
I	05-02 14:52:4...	28491	28544	?	Unity	Crosswalk ID: 1 intersection: 1 coordx: 41.15982 coordy: -8.624811
I	05-02 14:52:4...	28491	28544	?	Unity	
I	05-02 14:52:4...	28491	28544	?	Unity	(Filename: /Users/builduser/buildslave/unity/build/artifacts/genera @ ted/common/runtime/DebugBindings.gen.cpp Line: 51)
I	05-02 14:52:4...	28491	28544	?	Unity	Added crosswalk 1 to the objectList
I	05-02 14:52:4...	28491	28544	?	Unity	
I	05-02 14:52:4...	28491	28544	?	Unity	(Filename: /Users/builduser/buildslave/unity/build/artifacts/genera @ ted/common/runtime/DebugBindings.gen.cpp Line: 51)
I	05-02 14:52:4...	28491	28544	?	Unity	Crosswalk ID: 2 intersection: 1 coordx: 41.15999 coordy: -8.624674
I	05-02 14:52:4...	28491	28544	?	Unity	
I	05-02 14:52:4...	28491	28544	?	Unity	(Filename: /Users/builduser/buildslave/unity/build/artifacts/genera @ ted/common/runtime/DebugBindings.gen.cpp Line: 51)
I	05-02 14:52:4...	28491	28544	?	Unity	Added crosswalk 2 to the objectList
I	05-02 14:52:4...	28491	28544	?	Unity	
I	05-02 14:52:4...	28491	28544	?	Unity	(Filename: /Users/builduser/buildslave/unity/build/artifacts/genera @ ted/common/runtime/DebugBindings.gen.cpp Line: 51)
I	05-02 14:52:4...	28491	28544	?	Unity	Crosswalk ID: 3 intersection: 1 coordx: 41.1599 coordy: -8.624555
I	05-02 14:52:4...	28491	28544	?	Unity	
I	05-02 14:52:4...	28491	28544	?	Unity	(Filename: /Users/builduser/buildslave/unity/build/artifacts/genera @ ted/common/runtime/DebugBindings.gen.cpp Line: 51)
I	05-02 14:52:4...	28491	28544	?	Unity	Added crosswalk 3 to the objectList

Figure 5.5: Server Data Logs.

As we can saw at the 5.5 Figure, the server data was received correctly and successfully saved by the ARIMaRN application.

To test the mapping and scene build the associated logs were analysed. If the scene is being built correctly, using the mock locations, the application must create the virtual objects referring to the expected infrastructures, being the infrastructures in a distance between 5 and 20 meters. The first mock localisation used was (41.1597801, -8.6246951). According to that, the expected infrastructures that should be built are the traffic lights 1 and 7, the crosswalks 1 and 3, and the electromagnetic coil 2. As we can see in the 5.6 image, there were the created ones. Moreover, when we use the Fake GPS application's joystick, to change the user's current position, the camera and the scene changes too, according to the new user's localisation. Thus, the virtual objects are created and destroyed correctly, as we can see at Figure 5.7.

To test if we can see the BIM data when we touch the virtual infrastructure, it was touched, and the respective canvas appeared with the info about the infrastructure, Figure 5.8. Comparing the canvas info about the infrastructure with the data on the database we can conclude that the data are displayed correctly and is related to the pressed down infrastructure too.

Case Study

```

-----
ted/common/runtime/DebugBindings.gen.cpp Line: 51
W 05-02 15:16:4... 2434 3249 ? Unity Server Connection [Creating V.O.]: Traffic Id: 1
W 05-02 15:16:4... 2434 3249 ? Unity
W 05-02 15:16:4... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
I 05-02 15:16:4... 2434 3249 ? Unity [Augmented Script]: Traffic Light Orientation: 244
I 05-02 15:16:4... 2434 3249 ? Unity
I 05-02 15:16:4... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
W 05-02 15:16:4... 2434 3249 ? Unity Server Connection [Creating V.O.]: Traffic Id: 7
W 05-02 15:16:4... 2434 3249 ? Unity
W 05-02 15:16:4... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
I 05-02 15:16:4... 2434 3249 ? Unity [Augmented Script]: Traffic Light Orientation: 178
I 05-02 15:16:4... 2434 3249 ? Unity
I 05-02 15:16:4... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
W 05-02 15:16:4... 2434 3249 ? Unity Server Connection [Creating V.O.]: Espiras Id: 2
W 05-02 15:16:4... 2434 3249 ? Unity
W 05-02 15:16:4... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
W 05-02 15:16:4... 2434 3249 ? Unity Server Connection [Creating V.O.]: Crosswalk Id: 1
W 05-02 15:16:4... 2434 3249 ? Unity
W 05-02 15:16:4... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
W 05-02 15:16:4... 2434 3249 ? Unity Server Connection [Creating V.O.]: Crosswalk Id: 3
W 05-02 15:16:4... 2434 3249 ? Unity
W 05-02 15:16:4... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
I 05-02 15:16:4... 2434 3249 ? Unity Espira ID: 3 intersection: 1 coordx: 41.15992 coordy: -8.624467
I 05-02 15:16:4... 2434 3249 ? Unity

```

Figure 5.6: Creating Scene Logs.

```

E 05-02 15:19:0... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
E 05-02 15:19:0... 2434 3249 ? Unity [GPS Script]: Camera moved to (x - z): 32.73875 - 49.13625
E 05-02 15:19:0... 2434 3249 ? Unity
E 05-02 15:19:0... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
W 05-02 15:19:0... 2434 3249 ? Unity Server Connection [Creating V.O.]: Espiras Id: 3
W 05-02 15:19:0... 2434 3249 ? Unity
W 05-02 15:19:0... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
E 05-02 15:19:0... 2434 3249 ? Unity [GPS Script]: Camera moved to (x - z): 32.73875 - 50.40702
E 05-02 15:19:0... 2434 3249 ? Unity
E 05-02 15:19:0... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
W 05-02 15:19:0... 2434 3249 ? Unity Server Connection [Disabling V.O.]: 20.35 meters - Espiras Id: 2
W 05-02 15:19:0... 2434 3249 ? Unity
W 05-02 15:19:0... 2434 3249 ? Unity (Filename: /Users/builduser/buildslave/unity/build/artifacts/genera
ted/common/runtime/DebugBindings.gen.cpp Line: 51)
E 05-02 15:19:0... 2434 3249 ? Unity Server Connection [Disabling V.O.]: 20.35 meters - Espiras Id: 2

```

Figure 5.7: Moving Camera and Changing scene according to user's Localisation.

The Moving Window Algorithm was also tested using the mock locations. The algorithm was capable of detecting correctly when the user stopped, and when it is going too fast for a walk. Therefore, the algorithm filtered the GPS readings and gave more accurate values to use.

After all the tests performed, using the mock locations, the ARIMaRN application worked as expected. The virtual world was correctly created as well as the selection of one infrastructure to get its BIM data. Moreover, the application changes the virtual scene according to the user's position in the real world. The developed Moving Window Algorithm is also capable of return more accurate GPS readings, resulting in a more smooth visualisation of the virtual scene. With very distinct GPS readings the virtual scene would be shaking constantly since the virtual camera changes its position according to the GPS readings.

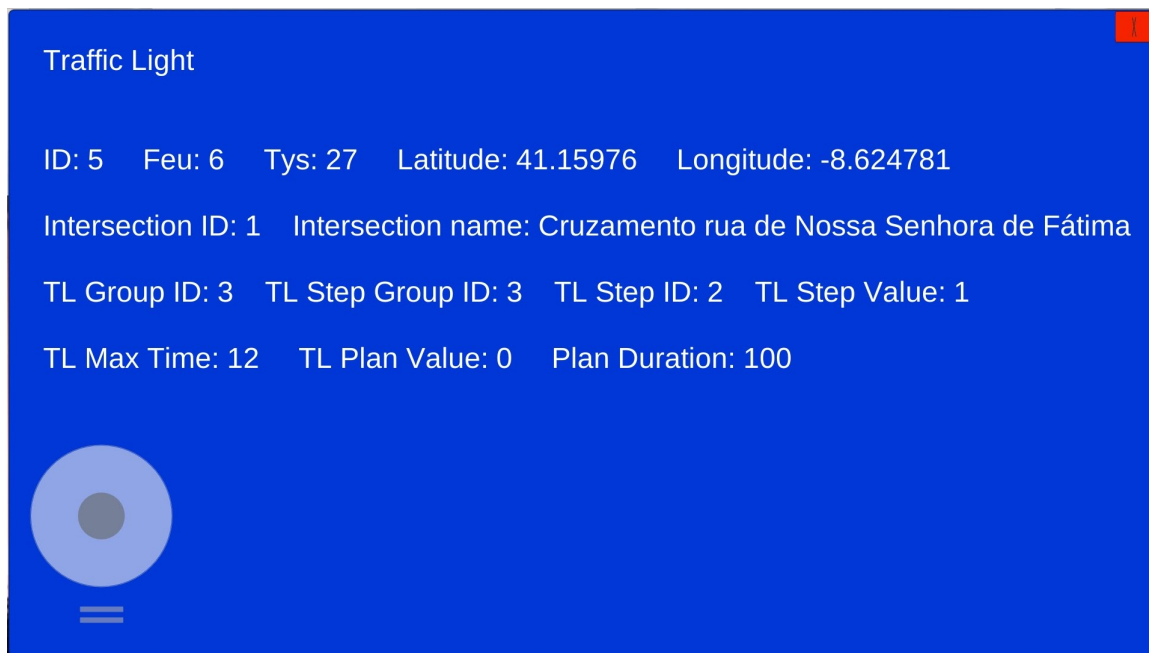


Figure 5.8: Traffic Light BIM data Information - ARIMaRN Application.

5.2.2.2 Real World Performance

After the ARIMaRN application passed successfully through the mock locations tests, the app was tested at the RNSFP road junction. We already know that the application can communicate with the server and get the GIS and BIM data to build the virtual infrastructures and also the information canvas of each one. The coordinates mapping and the GPS algorithm are now the focus of the testing.

When the application was opened in the real world place, the app started and connected to the server as expected. Therefore, the virtual scene is created and pressing the infrastructures it opens the information screen.

Moving around across the street we could see that the infrastructures appear and disappear according to the distance thresholds as expected. Using the Pythagorean theorem, we can calculate the coordinates mapping error. Choosing two points of the square, we can calculate the distance and calculate the same distance using latitude and longitude coordinates. To verify the error near of the edge, we can use the PC and the new $PE(41.160050, -8625000)$ points. Using the latitude and longitude coordinates the distance result is 72.28 meters. Now calculating using the 5.1 to 5.8 equations, the result is 72.38 meters. As we can see, the mapping has a precision of 10 centimetres even near of the edge, being precise in the middle of the square. This error value is a really small error since we do not take into account the earth radius on the calculus.

All the developed algorithms worked fine until now. The more problematic one is the GPS algorithm that influences the viewing of the AR scene. If the application gets amiss GPS readings, the AR scene will be displayed at wrong places. To evaluate the GPS algorithm was developed a

Case Study

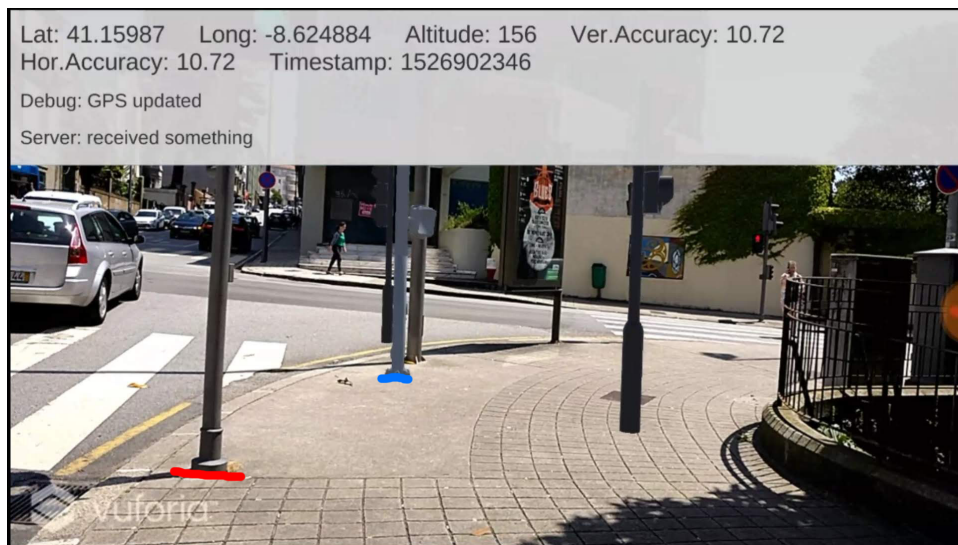


Figure 5.9: Distance error. The red line is the real traffic light location. The blue line is the virtual traffic light position.



Figure 5.10: Traffic light orientation error between the upper traffic light and the virtual one.

canvas that gives the user the information about the run-time GPS info, as we can see at Figure 5.9.

Moving around again to test the GPS values, we can find out that with the One Plus 3 device, the best values regarding the accuracy have an error of at least 4 meters. Moreover, the most common GPS readings have between 8 to 15 meters of error. Such significant values of error can mislead the developed Moving Window algorithm for GPS. Another problem with the GPS readings is the fact that the coordinates sometimes do not change even when we are walking for five or more meters. Walking without new GPS readings leads to a state where the virtual camera

Case Study

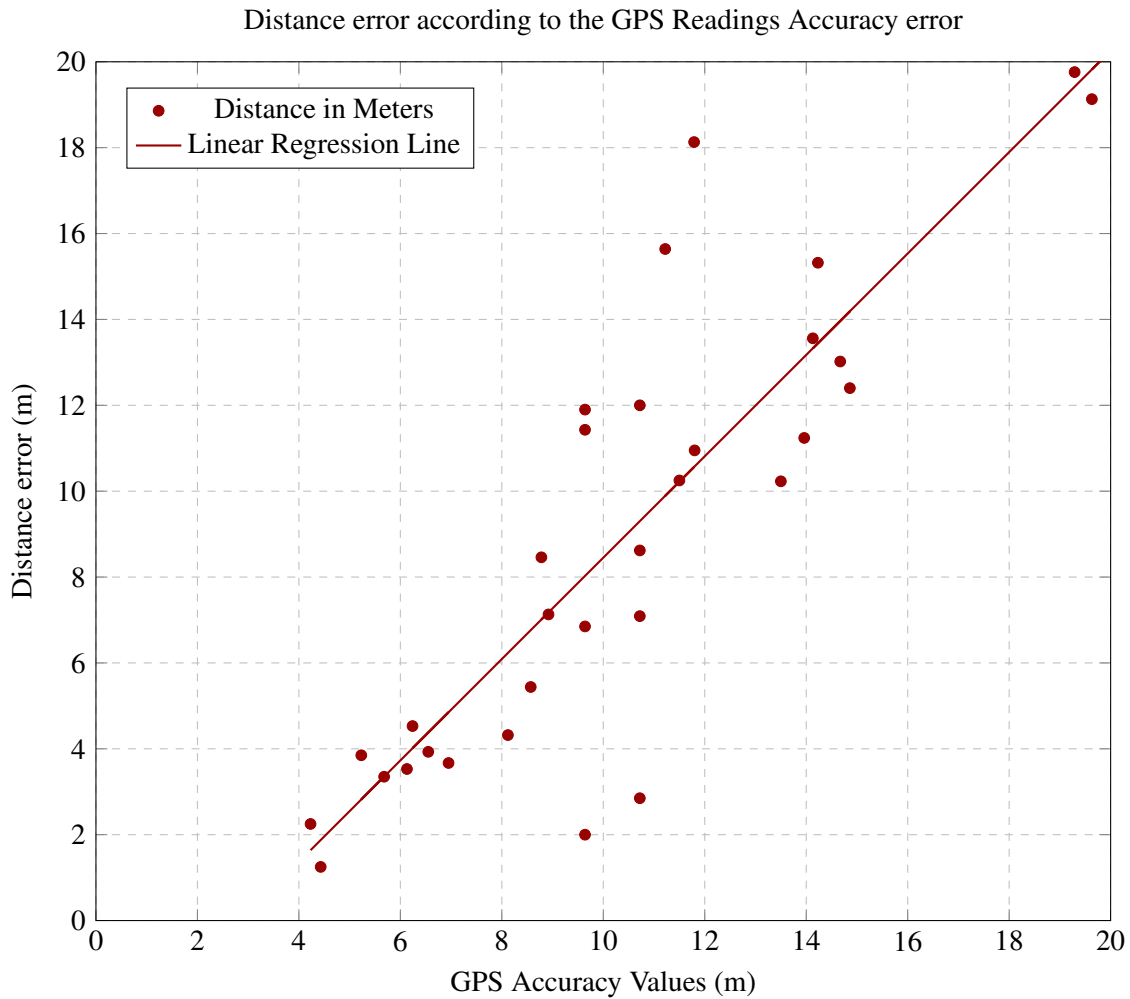


Figure 5.11: The plot shows the accuracy influence of the GPS readings on ARIMaRN application results.

does not translate the real camera movements, and consequently, the virtual objects will not be displayed at the right place. Furthermore, as the virtual camera does not move correctly, the user has the sensation that the virtual objects are walking with him as if stuck to the screen.

When the GPS readings are changing the developed Moving Window algorithm attenuates the GPS inaccurate values giving a more smooth run of the application regarding the AR visualisation. Thereby, the virtual objects do not jump on the screen as they would jump if it were only used the GPS readings without any filtering.

The Graph 5.11 shows the influences of the GPS readings on the ARIMaRN application results. If the accuracy of the GPS values is less than 8 meters of error, the ARIMaRN application can give us relatively good results. However, with more inaccurate values of GPS, the app will show a poor AR experience. Analysing the graph we have another proof that the GPS readings are the biggest influencers in the application. As we can see, with higher values of the GPS accuracy error we have more imprecise results on the app, and also more distance error between the real

and the virtual infrastructure. Therefore, we need precise values to have a perfect AR experience.

Despite the GPS problems, after moving around to update the GPS values, the application shows the virtual objects with 1 to 4 meters of error within a circumference with centre at the real infrastructure position, and the orientation of the traffic lights with errors of 10 to 50 degrees, Figures 5.9 and 5.10. These values are influenced by the accumulated error of two calculus, the mapping to Cartesian coordinates and the GPS readings. Moreover, the orientation of the traffic light relies on the device's compass that also can give us inaccurate values.

It is important to point out that the application has high power consumption. The app uses the GPS and the internet connection, to better GPS results and to connect to the server. Thereby, as the GPS and internet connection consumes lots of energy, the ARIMaRN application consumes a lot of energy to build the scene.

5.3 Summary

The ARIMaRN-PSAL application was able to create the virtual signage correctly. It also lets evaluating the signal's position by allowing the user to look around it and change the distance to the virtual signal. This app only has one signal model has proof of concept. However, the addition of more models is a quick task. Thereby, we can state that the placing of signage at a location as a successful feature of our application. Moreover, this feature can be included alongside with other ones, without the need for an app for each feature, but only if the application supports the Instant Tracking.

The ARIMaRN app shows us good results on the integration of the BIM, GIS and AR technologies. The IoT integration was not developed due to external factors such as the Porto City Hall does not have enough sensors to fill the database, and I do not have any financial investment in the project. However, we can assume that the integration of the four technologies can be performed because the IoT task is to update the database. All the other steps are done.

Despite the encouraging results, the app can be improved. We have error accumulation in the coordinates mapping, GPS readings, and compass tasks. An improved mapping, for instance using the earth radius on the calculus, gives us more accurate positions. Improving the acquire of the compass values can provide us with the correct orientation of the infrastructures. However, the biggest problem is the GPS values. These values are profoundly influenced by external and internal factors. As we saw, the earth's atmosphere, the building around, the software and hardware errors, among others, mislead the app through the GPS readings, and it affects the AR visualisation.

If the GPS readings are not precise or even do not change for a while when we are walking, the virtual camera does not behave itself like the real camera. Hence, the app gives the user a poor AR experience and results since the virtual camera changes its position according to the GPS readings and the coordinates mapping algorithm.

In short, in addition to the IoT integration, the main focus of improvement is the GPS algorithm to get the most accurate values possible.

Case Study

Chapter 6

Conclusions

The BIM-GIS integration is a hard task to do, especially when we are talking about doing a seamless integration. Different methodologies were tried, such as unidirectional and bidirectional, geometric and semantic ones. Moreover, these methodologies were applied at different levels, like at the application level, at the data level, and at the process level. Despite all the efforts, a seamless integration between the BIM and GIS technologies was not reached in the studies of the performed research. The two technologies have different backgrounds, a different level of development and a different coordinate system, among others differences. Furthermore, even the users of each one are different, and they expect different outcomes of the system. Moreover, the two technologies have a different geometric representation resulting in a significant barrier to their integration.

Crossing the data, and having the data up-to-date with the IoT sensors, provides vast information about the infrastructures, leading to the prediction of events. Therefore, we can use that information to prepare the infrastructures for possible future events. Moreover, we can perform the routine inspections of the infrastructures using the information gathered by the sensors. Thus, we can develop algorithms to accomplish this task without the interference of the human being on the system or without the need for humans going to the infrastructure site. However, more factors have to be studied. Especially in the beginning, the IoT sensors could lead to some wrong information. The system needs to overcome the interoperability, privacy and security issues to have trustful data.

The use of AR / VR technology provides the visualisation of the hidden information about the elements of the infrastructure. Furthermore, through the use of the mobile devices, the data can be exchanged at the site of the infrastructure, by interacting with the system through the Augmented Reality. The interaction with the system is performed in real-time, and the users can see the updated information of the infrastructure superimposed on the real scene.

In this document was presented the work done towards a BIM-GIS-IoT-VR/AR integration aiming to make smarter the management and the maintenance of the road networks' infrastructures. As we also saw that the work made so far in the construction, management, and maintenance

Conclusions

of the infrastructures is having a shift to integrate more than just the BIM and the GIS technologies. The IoT and the VR/AR are relevant and practical technologies to integrate with BIM and GIS resulting in a better analysis of the road infrastructures, or to all infrastructures if we apply broadly the concept. Thereby, we can have better decision-making based on the analyses provided by our integration system. We are talking about analyses of energy consumption, temperature, degradation status, water levels, noise evaluation, air quality, among other possibilities.

The integration of the four technologies can be used for many purposes. The two developed applications explore the virtual signage positioning and the use of AR to display in real-time the information about the infrastructures. However, other different systems could be developed based on this integration.

The ARIMaRN-PSAL app revealed itself as a powerful tool in the management of the road networks' infrastructures. The developed concept of proof allows us to test with a traffic light model the best location to put the new signal. Extending the application with other signals' models, we can study the placement of the road signage at the infrastructure's site by viewing the virtual model of the signal. Therefore, we can avoid situations of wrong placement that lead to several daily problems and a future replacement of the signal's location. By avoiding such kind of problems, we can save significant amounts of money on the road management. For instance, there is no need to send another team to replace the location of the signals, that leads to more efficient work. Thereby, the teams can do their job fastly, and they can cover the whole city with fewer delays.

The other developed application was the ARIMaRN one. This app aims to integrate the four mentioned technologies. However, the IoT integration was not be developed yet. Nevertheless, the IoT role on the full integration is to update the database's BIM and GIS data. Therefore, the system can easily integrate the IoT technology and it is prepared to handle it. The application allows the users to visualise the virtual models of the infrastructures and interact with them by showing the BIM data relating to each one. The AR experience is performed in real-time and also at the infrastructures location. This kind of interaction leads us to smarter management and maintenance of the road networks. The City Hall employees can use the ARIMaRN application when walking outside of the office, on the infrastructures location. Therefore, there is no need of analysing paper documents to understand the state of the infrastructures such as different forms to report a maintenance job. The information is gathered in only one place. Moreover, the use of the AR experience allows the quick identification of the infrastructure and, consequently, a fast way of getting the data about it. Therefore, the app integrates the BIM, GIS, and AR technologies successfully. The GIS data is used to get the location and the orientation of the infrastructures. The BIM data of a chosen infrastructure is displayed to the user after touching on a virtual infrastructure. The models and the interaction with them is performed by the AR, using the Vuforia SDK. The ARIMaRN application can also be extended to handle different types of information both BIM and GIS.

The developed applications pointed out the main problems of AR system, mainly on the tracking and the geolocation. The Instant Tracking is a relatively novel feature of the AR SDK, and it is not supported by all the mobile device. Therefore, the development of new apps based on that

Conclusions

feature only can be used for a small portion of the population. This kind of barriers sometimes makes the developers give up of creating apps with the unsupported features. However, the AR technology is still evolving and giving signals of being one of the leading future technologies. Thereby, we can see more applications being created in the AR field, even the ones that use the Instant tracking feature.

The most lacking feature of the AR SDK to point out is the geolocation. Nowadays, almost every single person owns a mobile device and uses it to check their position on the world. We have an almost infinite number of applications that rely on the geolocation services, and they allow the users to do several different tasks and also get information about it. Therefore, the majority of the people are used to have geolocation-based applications in their daily lives. Thus, it is required that the AR technology can integrate these geolocation services to present new solutions for the users. We saw the Wikitude SDK that has the geolocation services. However, it does not work correctly on every device, and even when it works the geolocation have some problems and some associated errors. The label or virtual object never stay at the right position, being always near of it.

We can also try to use two different SDKs to integrate the AR and the geolocation services. However, once again we have associated errors on the geolocation side and that errors propagate themselves to the AR technology, giving a poor experience to the user. Moreover, the AR is not the only one that is evolving and getting crucial steps to a convincing performance. The geolocation SDKs are also very few, and they need to improve themselves. In short, both geolocation services and augmented reality needs to improve creating the bases for the development of new geolocation-based AR applications. Otherwise, we will continue with poor results, and consequently, the users will not trust our applications.

The development of the ARIMaRN application shows the influences of the GPS readings on our apps. Many different external and internal factors influence the GPS values. The hardware and software of the receiver, the atmosphere, the buildings, and the trees are the main influencers. Therefore, we have inaccurate GPS values for the walking people in a city. Moreover, sometimes the GPS values do not change even if the user is walking for five to ten meters. These GPS readings mislead the GPS-based apps and most of the times give bad performances and experiences to the users. Once again, we have a technology that needs to be improved. This time is the GPS one. Once we have more precise GPS readings, the error does not propagate itself through the other tasks of the apps. More accurate values lead to better performances and user's experiences.

6.1 Main Contributions

6.1.1 ARIMaRN-PSAL

According to the results of the tests applied to the ARIMaRN-PSAL application, the Instant Tracking feature reveals to be capable of maintaining the virtual signal at the right place when the user is evaluating the best position. However, if the user points the device's camera in a way that the plane

Conclusions

surface is out of view, the signal's model could get out of the user's chosen position and mislead the user's study. Thereby, the user must have some precautions when it is on the analysing time.

With some improvements on the Instant Tracking feature, the developed application can be even more precise, leading to better evaluations. Hence, the app shows to be capable of solving the initial problem with good results, with the option of upgrading it to more accurate performances.

6.1.2 ARIMaRN

Analysing the app results, we can point out a good outcome even with the weak GPS readings. The ARIMaRN application is capable of showing the virtual objects within a circumference with a radius of 1 to 4 meters, being the centre the real infrastructure position. Moreover, the orientation of the traffic lights has an error of 10 to 50 degrees. These high values of accuracy, when the average of the received GPS values error are at least 8 meters, shows us that the app is capable of doing an excellent mapping of the real coordinates to the virtual ones and also that the developed GPS algorithm helps to have more steady and precise values. However, although the strong results, the app has an error accumulation through its different tasks. The mapping between the real and virtual coordinates has an associated error that is increased by the poor GPS readings. Thereby, if the system receives very inaccurate GPS values, the application can be misleading and the AR experience is not trustable, accurate, and possibly it will not be aligned in 3D as it is supposed to be. The compass values to calculate the user's orientation is another possible way of accumulating error. Different devices will have distinct compass values even if they are oriented to the same compass degree in real life. As we can see, the available technologies, especially the GPS and compass, still need improvements to get more accurate results. Otherwise, we need to develop more sophisticated algorithms to overcome the inaccurate readings' influences in our applications.

The ARIMaRN application has high power consumption. We know that the GPS functionality is high costly regarding the power consumption. Moreover, the mobile internet connection used to communicate with the server also costs lots of energy. Therefore, as the app uses two functionalities with high power consumption, the application itself consumes a lot of energy to give us the AR experience.

Thus, the developed ARIMaRN app can solve the initial problem with great results, also with the possibility of improvements in its algorithms to achieve even better performances. Nevertheless, the app does not integrate the IoT technology. The IoT integration needs to be developed, and it does not need to be a feature of the ARIMaRN app. The developed application uses the GIS and BIM data stored on the database through the server connection. Thereby, the IoT integration only needs to update the database to achieve the integration of the four technologies. The IoT feature can be a different app, and then the ARIMaRN application uses the real-time updated database for its purposes. Therefore, the system can be categorised as working at the application level, since we are integrating two different applications to achieve only one goal. The primary sensors available in today's road networks are the traffic lights controllers. Many other sensors must be inserted on the road networks' infrastructures to make possible to use them as an IoT system. Sensors of temperature, water levels, pressure, proximity, chemical, smoke, gas, air quality, among others,

Conclusions

can give us the information to automatic decision-making that leads to better management and maintenance of our city's roads. The ARIMaRN application is one step towards that main goal. With it, we can have access to the infrastructure's info and act in conformity. More applications and systems are necessary to build robust decision-making programs capable of predict potential disasters before their occurrence and to have a refined management and maintenance system of not just our road networks but also the entire city. Thereby, the ARIMaRN app has vital importance as one of the first steps to achieve the holistic concept of the smart cities. This project is the beginning of the study on the most suitable technologies to use in the future, exploring their limitations and their advantages. The BIM, GIS, IoT, VR/AR technologies revealed to be the bases of the near future. They are the more robust ones to handle with our cities infrastructures and help us to have more sustainable cities. Hence, more sustainable cities bring us a greener world that also leads to a better quality of life. Therefore, studying and working towards the use of these four technologies' integration becomes crucial in the smart cities and green solutions research.

6.2 Further Development

6.2.1 ARIMaRN-PSAL

The ARIMaRN-PSAL app can be extended with more signal models. The developed concept of proof only has available one standard traffic light model available online for free. Furthermore, we can try to use the GIS geometric representation as the signal's model. However, to achieve that it is necessary the export of the GIS representation to the Unity engine supported prefabs format. Another way can be the use of another engine to enable the fast development of the AR experience that can read the GIS model's format. In addition to that, the Instant tracking algorithms must be improved to provide better and more accurate augmented reality experiences. Better 3D alignment and positioning of the virtual objects offer more engaging, useful, and entertaining apps for the users.

6.2.2 ARIMaRN

ARIMaRN is the app that aims to integrate four different technologies. Therefore, as we still have problems with each one of them, using all together accumulates the errors. The IoT task on the system, even knowing that the IoT integration is not yet developed, can mislead the database info. If we have for instance issues related to the security of the IoT sensors, the system can update the BIM and GIS data with wrong information. Thus, it is necessary to build a robust IoT system that resists the external influences, and also to internal ones such as sensors' malfunctioning.

The use of GPS-based applications can be a tough task as we saw above. New algorithms to overcome the GPS issues must be developed on both GPS receivers' software and the applications' software. However, this task is not trivial.

The developed server allows the communication between the app and the database to get the requested data. However, the server does not allow for insertion of new data into the database

Conclusions

tables. Allowing the addition of information through the ARIMaRN application is a feature that could help on the updating of the data related to each infrastructure. The IoT sensors could not be able to report some specific status, and the user could insert them using the app easily and quickly. Thus, the insertion of data about one infrastructure is a helpful feature to develop in the future.

In the same line of thinking as in the ARIMaRN-PSAL app, we can try to use the GIS models of the road infrastructures instead of the used ones on the ARIMaRN application. Once again we need to find a solution to the successful conversion of the GIS geometric representation to the supported formats of the Unity engine. Achieving that conversion leads to an even better AR experience. It is important to point out that the representation to use must be the GIS one instead of the BIM's. The BIM representation is in a virtual world and does not represent the surroundings correctly. On the other side, the GIS representation is the indicated one to use.

As we can conclude, the ARIMaRN application was developed aiming the use of it when the user is walking. The GPS algorithm will discard values that are not between the thresholds stated as walking standard values. Therefore, if the user is using its car, the app will have a poor performance. Thereby, allowing the possibility of other ways of users' moving could be well seen for some of them.

The developed mapping of the real coordinates to the virtual ones can only be used on short distances since it does not take into account some influences such the earth radius. Thus, including some new parameters on the coordinates mapping could lead to a more accurate virtual map.

The virtual objects of the application do not have the illumination according to the real scene. They are built using the default light of the Unity engine. Therefore, the shadows and the less illuminated points of the object are not according to the real infrastructure. However, this is a demanding feature on the use of computational power. As the application aims to mobile devices, it is arguable if that kind of "details" should be present or not. The primary goal is to identify the infrastructure and get data about it. Probably the different illumination is not an obstacle in that recognition. Nevertheless, the feature can be developed to improve the ARIMaRN performance and the user experience.

In short, the two developed applications revealed to be an essential step to reach the smart cities holistic concept, especially concerning the management and maintenance of the road networks. Moreover, the ARIMaRN application shows that it is possible to integrate the BIM, GIS, IoT, and the VR/AR technologies in the same project. Some improvements are needed on each of the developed apps. However, the results of both showed a good performance even on adverse conditions. These two apps are ones of the first steps to smarter management and maintenance of our city's infrastructures. With this kind of steps, we are moving towards a more intelligent and greener world that, thanks to that, is also more sustainable.

6.3 Future Research Lines

Nevertheless, we can apply the BIM-GIS-IoT-VR/AR integration for other purposes. On the construction of buildings, we can give a powerful tool for each of the different type of workers. The

Conclusions

workers need to analyse the project of the building to build it according to its project specifications. If the workers have an app allowing the visualisation of the required information through an augmented reality environment, the access to the data is more accessible and quicker. Moreover, the app could allow the insertion of the steps already done by the workers, giving full information about the development status of the building. Therefore, we also can apply the same concept to the infrastructures' construction, having access to all the data related to each infrastructure. Thereby, as we already saw, we can analyse the real-time data and use it in decision-making systems that can prevent the waste of some materials, problems to have all the operators working at the same time, among other related issues. We can also add the full virtual reality environment to give access to the infrastructures data without leaving the chair. Hence, for instance, the stakeholders could see the development status of the infrastructures in real-time and understand better any emerging problem. This VR environment could also allow the city hall employees the evaluation of the city's infrastructures during the construction phase without the risk of a work accident.

We are facing more frequently some natural phenomena, especially related to the weather. Heavy rains and strong winds can have catastrophic effects on our cities. Thereby, we can use the integration of the four technologies to build systems for monitoring the water levels in the city. The system can also allow the simulation of water levels problems. Thereby, we can analyse the best ways to evacuate the city and build more powerful outflows in the most problematic areas.

The pipelines of public water, electricity, and natural gas are generally in the underground. The access to them is difficult because we have asphalt over practically all of their path. Nowadays, the maintenance work is performed using paper documents to find the location of the tubes. Using the four technologies integration, we can have a system that displays the path of all the pipelines over the ground. Thereby, the city hall employees only need to use the app, and they will have a feedback image that helps in not to cut any undesirable tube and also they will have all the information related to the pipes, such as the flux inside of them, the length, the degradation status, among others.

One broad use of the four technologies integration could be the development of an app for the citizens. The app should allow the reporting of infrastructure issues by the users. The citizens could point their mobile devices to one infrastructure, and by touching it, they can report some malfunction or another problem related to the infrastructure. Especially in the first days of the IoT systems' implementing and use, the reporting of the issues by the citizens could be a critical support to the establishment of updated BIM and GIS databases with the data about the infrastructures.

Thus, the integration of the BIM, GIS, IoT, and VR/AR technologies is a powerful tool in the management and maintenance of the city's infrastructures. With systems using it, we can achieve more efficiency on the city's maintenance and management work. Moreover, we can apply the integration of the four technologies in the most vital problems of the cities leveraging them to more sustainability.

Conclusions

References

- [ACoy17] Haslina Arshad, Lam Meng Chun, Waqas Khalid Obeidy, and Tan Siok Yee. An Efficient Cloud based Image Target Recognition SDK for Mobile Applications. *International Journal on Advanced Science, Engineering and Information Technology*, 7(2):496–502, 2017.
- [AHRL17] Anil Armagan, Martin Hirzer, Peter M. Roth, and Vincent Lepetit. Learning to Align Semantic Segmentation and 2.5D Maps for Geolocalization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3425–3432, 2017.
- [AKAS17] Wesam Almobaideen, Rand Krayshan, Mamoon Allan, and Maha Saadeh. Internet of Things: Geographical Routing based on healthcare centers vicinity for mobile smart tourism destination. *Technological Forecasting and Social Change*, 123(April):342–350, 2017.
- [And18] Android Studio. Download Android Studio and SDK Tools | Android Studio, 2018. <https://developer.android.com/studio/index.html>, visited on 2018-03-01.
- [Ara08] Yusuf Arayici. Towards building information modelling for existing structures. *Structural Survey*, 26(3):210–222, 2008.
- [ARMN16a] Sam Amirebrahimi, Abbas Rajabifard, Priyan Mendis, and Tuan Ngo. A BIM-GIS integration method in support of the assessment and 3D visualisation of flood damage to a building. *Journal of Spatial Science*, 61(2):317–350, 2016.
- [ARMN16b] Sam Amirebrahimi, Abbas Rajabifard, Priyan Mendis, and Tuan Ngo. A framework for a microscale flood damage assessment and visualization for a building using BIM-GIS integration. *International Journal of Digital Earth*, 9(4):363–386, 2016.
- [ART15] ARToolkit. ARToolKit Documentation, 2015. <https://artoolkit.org/documentation/https://artoolkit.org/documentation/{#}Configuration>, visited on 2018-03-03.
- [ART16] ARToolkit. Building ARToolKit from Source [ARToolkit], 2016. <https://www.artoolkit.org/documentation/doku.php?id=8{ }Advanced{ }Topics:build{ }artoolkit>, visited on 2018-03-03.
- [ART18] ARToolkit. Download the ARToolKit Augmented Reality SDK | ARToolKit.org, 2018. <https://artoolkit.org/download-artoolkit-sdk>, visited on 2018-03-03.

REFERENCES

- [Aut16] Autodesk. AutoCAD for Mac & Windows, 2016. <https://www.autodesk.pt/products/autocad/overview>, visited on 2018-02-16.
- [Aut17] Autodesk. Família Revit - Software BIM - Autodesk, 2017. <https://www.autodesk.pt/products/revit-family/overview>, visited on 2018-02-16.
- [Aut18] Autodesk. Revit DB Link | Revit Products | Autodesk Knowledge Network, 2018. <https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/Revit-AddIns/files/GUID-97BF753E-F628-40BE-8757-0CA8304BFA37-htm.html>, visited on 2018-02-16.
- [AYES17] Naif M. Alsubaie, Ahmed A. Youssef, and Naser El-Sheimy. Improving the accuracy of direct geo-referencing of smartphone-based mobile mapping systems using relative orientation and scene geometric constraints. *Sensors (Switzerland)*, 17(10), 2017.
- [Azu97] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoper. Virtual Environ.*, 6(4):355–385, August 1997.
- [Ban11] V. K. Bansal. Use of GIS and Topology in the Identification and Resolution of Space Conflicts. *Journal of Computing in Civil Engineering*, 25(2):159–171, 2011.
- [BAS⁺17] Garvita Bajaj, Rachit Agarwal, Pushpendra Singh, Nikolaos Georgantas, and Valerie Issarny. A study of existing Ontologies in the IoT-domain. *CoRR*, abs/1707.00112, 2017.
- [BAU⁺17] Lorenzo Bottaccioli, Alessandro Aliberti, Francesca Ugliotti, Edoardo Patti, Anna Osello, Enrico Macii, and Andrea Acquaviva. Building Energy Modelling and Monitoring by Integration of IoT Devices and Building Information Models. *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, pages 914–922, 2017.
- [BBR⁺17] Martin Breunig, André Borrmann, Ernst Rank, Stefan Hinz, Thomas Kolbe, Matthäus Schilcher, Ralf Peter Mundani, Javier Ramos Jubierre, Matthias Flurl, Andreas Thomsen, Andreas Donaubaue, Yang Ji, Steffen Urban, Simon Laun, Simon Vilgertshofer, Bruno Willenborg, Mathias Menninghaus, Horst Steuer, Sven Wursthorn, Jens Leitloff, Mulhim Al-Doori, and Nima Mazroobsemnani. Collaborative multi-scale 3D city and infrastructure modeling and simulation. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, volume 42, pages 341–352, 2017.
- [BEI17] G. A. Boyes, C. Ellul, and D. Irwin. Exploring bim for operational integrated asset management – a preliminary study utilising real-world infrastructure data. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 4, pages 49–56, 2017.
- [Bil13] Filip Biljecki. *The concept of level of detail in 3D city models*, volume II. Section GIS technology, 2013.

REFERENCES

- [BKD⁺14] A. Borrmann, T. H. Kolbe, A. Donaubauer, H. Steuer, J. R. Jubierre, and M. Flurl. Multi-scale geometric-semantic modeling of shield tunnels for GIS and BIM applications. *Computer-Aided Civil and Infrastructure Engineering*, 30:263–281, 2014.
- [BNPM18] Ana Rita Gaspar B, Alexandra Nunes, Andry Pinto, and Anibal Matos. Comparative Study of Visual Odometry and SLAM Techniques. *ROBOT 2017: Third Iberian Robotics Conference*, 694, 2018.
- [Boo] Bootstrap. New Age - One Page Bootstrap Theme - Start Bootstrap. <https://startbootstrap.com/template-overviews/new-age/>, visited on 2018-02-25.
- [Boy05] H. Boyes. Security, privacy, and the built environment. *IT Professional*, 17(3):25 – 31, 2015/05/. security;privacy;built environment;building information modeling;BIM;digital asset management;Internet of Things;cybersecurity;wireless technologies;.
- [BPA17] Renata M.A. Baracho, Mário L. Pereira, and Maurício B. Almeida. Ontologia, Internet das Coisas e Modelagem da Informação da Construção (BIM): Estudo Exploratório e a Inter-relação entre as Tecnologias. In *CEUR Workshop Proceedings*, volume 1908, pages 141–146, 2017.
- [Bru17] Daniel Bruck. traffic light + animation by Daniel Brück - download 3D model - Sketchfab, 2017. <https://sketchfab.com/models/30f361478c8143fbb463dcf9b2a0cb61>, visited on 2018-03-09.
- [CAD15] CADBLOCKSfree. Traffic lights revit model - CADblocksfree -CAD blocks free, 2015. <https://www.cadblocksfree.com/en/traffic-lights.html>, visited on 2018-02-19.
- [Car05] Nick Carey. Establishing Pedestrian Walking Speeds. *Portland State University ITE Student Chapter*, page 4, 2005.
- [CC14] Youngwan Cho and Hyunkyoo Choi. Accuracy Enhancement of Position Estimation using Adaptive Kalman Filter and Map Matching. *International Journal of Control and Automation*, 7(7):167–178, 2014.
- [CDA15] Jack C P Cheng, Yichuan Deng, and Chimay Anumba. Mapping Schema Semi-Automatically Utilizing Linguistic and Text Mining Techniques. *Journal of Information Technology in Construction*, 20:193–212, 2015.
- [CL17] Youngwan Cho and Heejin Lee. A Position and Velocity Estimation Using Multifarious and Multiple Sensor Fusion. *International Journal of Fuzzy Logic and Intelligent Systems 2017*, 17(2):121–128, 2017.
- [CM17] Eirini Charalampaki and Athanasios G. Malamos. Tracking football players with a conventional mobile device camera. *Proceedings of the 21st Pan-Hellenic Conference on Informatics - PCI 2017*, pages 1–2, 2017.
- [con18] Wikipedia contributors. Autodesk revit — wikipedia, the free encyclopedia, 2018. https://en.wikipedia.org/w/index.php?title=Autodesk_Revit&oldid=830620528, visited on 2018-02-16.

REFERENCES

- [dBC97] Josef de Mendoza y Rios, Joseph Banks, and H Cavendish. Recherches Sur Les Principaux Problèmes de l’Astronomie Nautique. *Philosophical Transactions of the Royal Society of London*, 87:43–122, 1797.
- [DBe] DBeaver. DBeaver | Free Universal SQL Client. <https://dbeaver.jkiss.org/>, visited on 2018-02-17.
- [DCA16a] Yichuan Deng, Jack C.P. Cheng, and Chimay Anumba. A framework for 3D traffic noise mapping using data from BIM and GIS integration. *Structure and Infrastructure Engineering*, 12(10):1267–1280, 2016.
- [DCA16b] Yichuan Deng, Jack C.P. Cheng, and Chimay Anumba. Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. *Automation in Construction*, 67:1–21, 2016.
- [DM12] Charalampos Doukas and Ilias Maglogiannis. Bringing IoT and cloud computing towards pervasive healthcare. *Proceedings - 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2012*, pages 922–926, 2012.
- [DN10] Carrie S. Dossick and Gina Neff. Organizational Divisions in BIM-Enabled Commercial Construction. *Journal of Construction Engineering and Management*, 136(4):459–467, 2010.
- [DZY⁺18] Hongwei Du, Chen Zhang, Qiang Ye, Wen Xu, Patricia Lilian Kibenge, and Kang Yao. A hybrid outdoor localization scheme with high-position accuracy and low-power consumption. *Eurasip Journal on Wireless Communications and Networking*, 2018(1), 2018.
- [Ecl18] Eclipse. Eclipse desktop & web IDEs, 2018. <https://www.eclipse.org/ide/>, visited on 2018-02-26.
- [GHG16] Özgür Göçer, Ying Hua, and Kenan Göçer. A BIM-GIS integrated pre-retrofit model for building data mapping. *Building Simulation*, 9(5):513–527, 2016.
- [GKNH12] Gerhard Gröger, Thomas Kolbe, Claus Nagel, and Karl-Heinz Häfele. OGC City Geography Markup Language (CityGML) En-coding Standard. *Ogc*, pages 1 – 344, 2012.
- [Goo18a] Google. ARCore Overview | ARCore | Google Developers, 2018. <https://developers.google.com/ar/discover/>, visited on 2018-03-01.
- [Goo18b] Google. Quickstart for AR on the Web | ARCore | Google Developers, 2018. <https://developers.google.com/ar/develop/web/quickstart>, visited on 2018-03-02.
- [Gos18] Anuj Gosalia. Announcing ARCore 1.0 and new updates to Google Lens, 2018. <https://www.blog.google/products/google-vr/announcing-arcore-10-and-new-updates-google-lens/>, visited on 2018-03-01.
- [GP12] Gerhard Gröger and Lutz Plümer. CityGML - Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:12–33, 2012.

REFERENCES

- [Gra18] Gradle Build Tool. Gradle Build Tool, 2018. <https://gradle.org/>, visited on 2018-03-01.
- [GWT⁺17] Q.H. Gao, T.R. Wan, W. Tang, L. Chen, and K.B. Zhang. An Improved Augmented Reality Registration Method Based on Visual SLAM. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10345 LNCS:1–9, 2017.
- [HCX⁺17] Chengfang Hu, Xuejun Cheng, Xiao Xiao, Zeqiang Chen, Zhaohui Wang, Jian Xu, and Dengzhong Zhao. Integrated application of water informatization: A case study from Zengcheng Guangzhou China. In *2017 6th International Conference on Agro-Geoinformatics, Agro-Geoinformatics 2017*, 2017.
- [HHC13] Jung Rae Hwang, Chang Hee Hong, and Hyun Sang Choi. Implementation of prototype for interoperability between BIM and GIS: Demonstration paper. *Proceedings - International Conference on Research Challenges in Information Science*, pages 3–4, 2013.
- [Hib18] Hibernate. Hibernate. Everything data. - Hibernate, 2018. <http://hibernate.org/>, visited on 2018-02-20.
- [How06] R Howard. Describing the Changes in Architectural Information Free-Form Architectural Expression. *ITcon*, 11(January):395–408, 2006.
- [HPP02] Jonathan How, Nicholas Pohlman, and Chan-Woo Park. GPS Estimation Algorithms for Precise Velocity, Slip and Race-track Position Measurements. *SAE Motorsports Engineering Conference & Exhibition*, 2002.
- [HRB17] Shaun Howell, Yacine Rezgui, and Thomas Beach. Integrating building and urban semantics to empower smart water solutions. *Automation in Construction*, 81:434–448, 2017.
- [IMV⁺13] Will Ikerd, David Merrifield, James Vandezande, Walt Cichonski, Ron Dellaria, Brian Filkins, Murat Karakas, David Francis, Aaron Lawson, and Dan Russell. Level of Development Specification. *Bim Forum*, pages 0–124, 2013.
- [Io18a] Motive Io. Features - Motive.io, 2018. <https://www.motive.io/features/>, visited on 2018-03-16.
- [Io18b] Motive Io. Home - Motive.io, 2018. <https://www.motive.io/>, visited on 2018-03-16.
- [Isi15] U. Isikdag. BIM and IoT: A synopsis from GIS perspective. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 40(2W4):33–38, 2015.
- [JJ17] H. Gao H. Sun J. Jia, Q. Zheng. Research of ancient architectures in jin-fen area based on gis&bim technology. *Journal of Physics: Conference Series*, 842(1), 2017.
- [JW07] Dah Jing Jwo and Sheng Hung Wang. Adaptive fuzzy strong tracking extended Kalman filtering for GPS navigation. *IEEE Sensors Journal*, 7(5):778–789, 2007.
- [KGK16] Piotr Kaniewski, Rafał Gil, and Stanisław Konatowski. Algorithms of Position and Velocity Estimation in GPS Receivers. *Annual of Navigation*, 23(1):53–68, 2016.

REFERENCES

- [KH15] Tae Wook Kang and Chang Hee Hong. A study on software architecture for effective BIM/GIS-based facility management data integration. *Automation in Construction*, 54:25–38, 2015.
- [KI15] Ebrahim P. Karan and Javier Irizarry. Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services. *Automation in Construction*, 53:1–12, 2015.
- [KKSS16] Jung In Kim, Bonsang Koo, Sunduck Suh, and Wonho Suh. Integration of BIM and GIS for formal representation of walkability for safe routes to school programs. *KSCE Journal of Civil Engineering*, 20(5):1669–1675, 2016.
- [KTH16] A. R. Khairuddin, M. S. Talib, and H. Haron. Review on simultaneous localization and mapping (slam). In *Proceedings - 5th IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2015*, pages 85–90, 2016. Cited By :1.
- [LHX⁺16] Clyde Zhengdao Li, Jingke Hong, Fan Xue, Geoffrey Qiping Shen, Xiaoxiao Xu, and Margaret Kayan Mok. Schedule risks in prefabrication housing production in Hong Kong: a social network analysis. *Journal of Cleaner Production*, 134(Part B):482–494, 2016.
- [LLL18] Jung-chun Liu, Chu-hsing Lin, and Chi-yuan Lin. Accuracy Evaluation of Location Service Modes on Mobile Devices. *Sensors and Materials*, 30(3):373–383, 2018.
- [LML16] Peng Lin, Jian Ma, and Siuming Lo. Discrete element crowd model for pedestrian evacuation through an exit. *Chinese Physics B*, 25(3):1–8, 2016.
- [LMS⁺17] Gioele Luchetti, Adriano Mancini, Mirco Sturari, Emanuele Frontoni, and Primo Zingaretti. Whistland: An Augmented Reality Crowd-Mapping System for Civil Protection and Emergency Management. *ISPRS International Journal of Geo-Information*, 6(2):41, 2017.
- [LSZJ14] Hao Liu, Ruoming Shi, Ling Zhu, and Changfeng Jing. Conversion of model file information from IFC to GML. *International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3133–3136, 2014.
- [LWW⁺17] Xin Liu, Xiangyu Wang, Graeme Wright, Jack Cheng, Xiao Li, and Rui Liu. A State-of-the-Art Review on the Integration of Building Information Modeling (BIM) and Geographic Information System (GIS). *ISPRS International Journal of Geo-Information*, 6(2):53, 2017.
- [LYSC16] Zhihan Lv, Tengfei Yin, Houbing Song, and Ge Chen. Virtual Reality Smart City Based on WebVRGIS. *IEEE Internet of Things Journal*, 4662(c):1–1, 2016.
- [Map17] Mapbox. Mapbox Studio, 2017. <https://www.mapbox.com/mapbox-studio/>, visited on 2018-03-16.
- [Map18] Mapbox. Mapbox, 2018. <https://www.mapbox.com>, visited on 2018-03-16.
- [MCE⁺16] L. Marek, M. Campbell, M. Epton, M. Storer, and S. Kingham. Real-time environmental sensors to improve health in the sensing city. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 41(July):729–733, 2016.

REFERENCES

- [Mic17a] Microsoft. Compare Visual Studio Product Offerings | Visual Studio, 2017. <https://www.visualstudio.com/vs/compare/>, visited on 2018-02-19.
- [Mic17b] Microsoft. SQL Server Documentation | Microsoft Docs, 2017. <https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation>, visited on 2018-02-16.
- [Mic17c] Microsoft. Visual Studio IDE, Code Editor, VSTS, & App Center, 2017. <https://www.visualstudio.com/>, visited on 2018-02-19.
- [MN14] Clement Mignard and Christophe Nicolle. Merging BIM and GIS using ontologies application to Urban facility management in ACTIVE3D. *Computers in Industry*, 65(9):1276–1290, 2014.
- [NBS⁺17] Luigi Nardi, Bruno Bodin, Sajad Saeedi, Emanuele Vespa, Andrew J. Davison, and Paul H.J. Kelly. Algorithmic performance-accuracy trade-off in 3D vision applications using HyperMapper. *Proceedings - 2017 IEEE 31st International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2017*, pages 1434–1443, 2017.
- [NH17] Reza Nourjou and Mehdi Hashemipour. Smart Energy Utilities based on Real-Time GIS Web Services and Internet of Things. *Procedia Computer Science*, 110:8–15, 2017.
- [OBD⁺17] Ken Arroyo Otori, Filip Biljecki, Abdoulaye Diakité, Thomas Krijnen, Hugo Ledoux, and Jantien Stoter. TOWARDS AN INTEGRATION of GIS and BIM DATA: WHAT ARE the GEOMETRIC and TOPOLOGICAL ISSUES? In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 4, pages 1–8, 2017.
- [Ora18] Oracle. Java Persistence API, 2018. <http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html>, visited on 2018-02-20.
- [Pau18] Paula. Get SMART - Seamless AR Tracking with AR-Core and ARKit, 2018. <https://www.wikitudo.com/blog-smart-wikitudo-seamless-ar-tracking/>, visited on 2018-03-02.
- [PGSG17] Giovanni Pintore, Fabio Ganovelli, Roberto Scopigno, and Enrico Gobbetti. Mobile metric capture and reconstruction in indoor environments. *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, page 5, 2017.
- [PME⁺17] Edoardo Patti, Angelo Mollame, David Erba, Daniele Dalmaso, Anna Osello, Enrico Macii, and Andrea Acquaviva. Information Modeling for Virtual and Augmented Reality. *IT Professional*, 19(3):52–60, 2017.
- [Pos15] PostQGIS Development Team. Spatial and Geographic objects for PostgreSQL, 2015. <https://postgis.net>, visited on 2018-03-09.
- [Pro18] Apache Maven Project. Maven – Welcome to Apache Maven, 2018. <http://maven.apache.org/>, visited on 2018-02-20.

REFERENCES

- [PSCD17] Shirui Peng, Guofeng Su, Jianguo Chen, and Peng Du. Design of an IoT-BIM-GIS Based Risk Management System for Hospital Basic Operation. *Proceedings - 11th IEEE International Symposium on Service-Oriented System Engineering, SOSE 2017*, pages 69–74, 2017.
- [QGI18] QGIS. Welcome to the QGIS project!, 2018. <https://www.qgis.org/en/site/>, visited on 2018-02-20.
- [Rai17] Jamal Raiyn. Developing Vehicle Locations Strategy on Urban Road. *Transport and Telecommunication Journal*, 18(4):253–262, 2017.
- [RD06] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3951 LNCS:430–443, 2006.
- [RDFS14] Azarakhsh Rafiee, Eduardo Dias, Steven Fruijtier, and Henk Scholten. From BIM to Geo-analysis: View Coverage and Shadow Analysis by BIM/GIS Integration. *Procedia Environmental Sciences*, 22(0):397–402, 2014.
- [Rev17a] Revit. Revit Sample Project Files | Revit Products | Autodesk Knowledge Network, 2017. <https://knowledge.autodesk.com/support/revit-products/getting-started/caas/CloudHelp/cloudhelp/2017/ENU/Revit-GetStarted/files/GUID-61EF2F22-3A1F-4317-B925-1E85F138BE88-htm.html>, visited on 2018-02-20.
- [Rev17b] RevitCity. RevitCity.com | Object | Traffic light (Parametric), 2017. <https://www.revitcity.com/downloads.php?action=view{&}object{&}id=14324>, visited on 2018-02-19.
- [RP17] Eul Hee Roh and Sang Chan Park. A Study on the Quality of Life Improvement in Fixed IoT Environments: Utilizing Active Aging Biomarkers and Big Data. *Quality Innovation Prosperity-Kvalita Inovacia Prosperita*, 21(2):52–70, 2017.
- [Sak17] Sakilux. Fake gps location – Aplicações no Google Play, 2017.
- [SC06] Tina L. Stanton-Chapman and Derek A. Chapman. Using GIS to Investigate the Role of Recreation and Leisure Activities in the Prevention of Emotional and Behavioral Disorders. *International Review of Research in Mental Retardation*, 33:191–211, jan 2006.
- [SH17] Dieter Schmalstieg and Tobias Hollerer. *Augmented reality: Principles and practice*. Publisher Mark L. Taub, 2017.
- [Ska17] Skarredghost. How to install ARCore on all Android phones using ARCore-ForAll - The Ghost Howls, 2017. <https://skarredghost.com/2017/11/02/install-arcore-android-phones-using-arcoreforall/>, visited on 2018-03-03.
- [SL15] Ha Yoon Song and Jun Seok Lee. Detecting positioning errors and estimating correct positions by moving window. *PLoS ONE*, 10(12):1–33, 2015.

REFERENCES

- [SL17] Bo-Yi Sung and Chang-Hong Lin. A fast 3D scene reconstructing method using continuous video. *EURASIP Journal on Image and Video Processing*, 2017(1):18, 2017.
- [Spr18] Spring. 2. JPA Repositories, 2018. <https://docs.spring.io/spring-data/jpa/docs/1.5.0.RELEASE/reference/html/jpa.repositories.html>, visited on 2018-02-26.
- [SSM⁺15] M. Serrão, S. Shahrabadi, M. Moreno, J. T. José, J. I. Rodrigues, J. M.F. Rodrigues, and J. M.H. du Buf. Computer vision and GIS for the navigation of blind persons in buildings. *Universal Access in the Information Society*, 14(1):67–80, 2015.
- [TB09] John E. Taylor and Phillip G. Bernstein. Paradigm Trajectories of Building Information Modeling Practice in Project Networks. *Journal of Management in Engineering*, 25(2):69–76, 2009.
- [TBDH14] Sergio Trilles, Oscar Belmonte, Laura Diaz, and Joaquin Huerta. Mobile access to sensor networks by using GIS standards and RESTful services. *IEEE Sensors Journal*, 14(12):4143–4153, 2014.
- [TCDM13] C. E. Tolmer, C. Castaing, Y. Diab, and D. Morand. CityGML and IFC: Going further than LOD. *Proceedings of the DigitalHeritage 2013 - Federating the 19th Int'l VSMM, 10th Eurographics GCH, and 2nd UNESCO Memory of the World Conferences, Plus Special Sessions fromCAA, Arqueologica 2.0 et al.*, 1:645–648, 2013.
- [Tho98] Richard B. Thompson. Global Positioning System: The Mathematics of GPS Receivers. *Mathematics Magazine*, 71(4):260, 1998.
- [Thy18] Thymeleaf. Thymeleaf, 2018. <https://www.thymeleaf.org/>, visited on 2018-02-26.
- [TMRR12] Sha Tao, Vasileios Manolopoulos, Saul Rodriguez, and Ana Rusu. Real-Time Urban Traffic State Estimation with A-GPS Mobile Phones as Probes. *Journal of Transportation Technologies*, 02(01):22–31, 2012.
- [Tur12] Turbosquid. free traffic light 3d model, 2012. <https://www.turbosquid.com/FullPreview/Index.cfm/ID/1048013>, visited on 2018-04-20.
- [TWG⁺17] Jochen Teizer, Mario Wolf, Olga Golovina, Manuel Perschewski, and Markus Propach. Internet of Things (IoT) for Integrating Environmental and Localization Data in Building Information Modeling (BIM). *ISARC 2017 - Proceedings of the 34th International Symposium on Automation and Robotics in Construction*, pages 603–609, 2017.
- [TXC⁺17] Chin Feng Tsai, Yi Tin Xiao, Huei Shan Chen, Yi Xuan Ye, Chin Han Wang, and Ting Wen Liang. Integration of BIM & GIS to query management on pipeline of building-A case study of dormitory. In *Proceedings of the 2017 IEEE International Conference on Applied System Innovation: Applied System Innovation for Modern Technology, ICASI 2017*, pages 944–947, 2017.
- [Uni18] Unity. Unity, 2018. <https://unity3d.com/pt>, visited on 2018-02-19.

REFERENCES

- [VAW⁺17] S. Vilgertshofer, J. Amann, B. Willenborg, A. Borrmann, and T.H. Kolbe. Linking BIM and GIS Models in Infrastructure by Example of IFC and CityGML. *Computing in Civil Engineering 2017*, pages 133–140, 2017.
- [Vuf18a] Vuforia. Introduction to Ground Plane in Unity, 2018. <https://library.vuforia.com/articles/Solution/ground-plane-guide.html>, visited on 2018-03-15.
- [Vuf18b] Vuforia. SDK Download | Vuforia Developer Portal, 2018. <https://developer.vuforia.com/downloads/sampleshttps://developer.vuforia.com/downloads/sdk>, visited on 2018-03-03.
- [Vuf18c] Vuforia. SDK Download | Vuforia Developer Portal, 2018. <https://developer.vuforia.com/downloads/sdk>, visited on 2018-03-03.
- [War07] 3D Warehouse. Traffic Light with Pedestrian Light - 3D Warehouse, 2007. <https://3dwarehouse.sketchup.com/model/ad23dc0fe2158f7197f3f61c4eaa4852/Traffic-Light-with-Pedestrian-Light>, visited on 2018-02-16.
- [WHF⁺17] Peisong Wang, Qinghao Hu, Zhiwei Fang, Chaoyang Zhao, and Jian Cheng. DeepSearch: A Fast Image Search Framework for Mobile Devices. *ACM Trans. Multimedia Comput. Commun. Appl.*, 14(1):6:1—6:22, 2017.
- [Wik18] Wikitude. Download Wikitude SDK and all supported Extensions for Free, 2018. <https://www.wikitude.com/download/>, visited on 2018-03-02.
- [WLK⁺13] Xiangyu Wang, Peter E.D. Love, Mi Jeong Kim, Chan-Sik Park, Chun-Pong Sing, and Lei Hou. A conceptual framework for integrating building information modeling with augmented reality. *Automation in Construction*, 34:37–44, 2013.
- [WLXL14] Qian Kun Wang, Peng Li, Ya Ping Xiao, and Zhi Gang Liu. Integration of GIS and BIM in Metro Construction. *Applied Mechanics and Materials*, 608-609(2014):698–702, 2014.
- [WRM⁺08] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg. Pose tracking from natural features on mobile phones. *Proceedings - 7th IEEE International Symposium on Mixed and Augmented Reality 2008, ISMAR 2008*, pages 125–134, 2008.
- [WRM⁺10] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg. Real-time detection and tracking for augmented reality on mobile phones. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):355–368, 2010.
- [WSL⁺18] Phillip Webb, Dave Syer, Josh Long, Stéphane Nicoll, Rob Winch, Andy Wilkinson, Marcel Overdijk, Christian Dupuis, Sébastien Deleuze, Michael Simons, and Vedran Pavi. Spring Boot Reference Guide, 2018. <https://docs.spring.io/spring-boot/docs/current-SNAPSHOT/reference/htmlsingle/>, visited on 2018-03-09.
- [Wu17] Jianxin Wu. Introduction to Convolutional Neural Networks. *National Key Lab for Novel Software Technology*, pages 1–31, 2017.

REFERENCES

- [YFS17] Shinji Yamamura, Liyang Fan, and Yoshiyasu Suzuki. Assessment of Urban Energy Performance through Integration of BIM and GIS for Smart City Planning. *Procedia Engineering*, 180:1462–1472, 2017.
- [ZPX⁺17] Ray Y. Zhong, Yi Peng, Fan Xue, Ji Fang, Weiwu Zou, Hao Luo, S. Thomas Ng, Weisheng Lu, Geoffrey Q.P. Shen, and George Q. Huang. Prefabricated construction enabled by the Internet-of-Things. *Automation in Construction*, 76:59–70, 2017.
- [ZSH⁺18] Kaiyue Zang, Jie Shen, Haosheng Huang, Mi Wan, and Jiafeng Shi. Assessing and Mapping of Road Surface Roughness based on GPS and Accelerometer Sensors on Bicycle-Mounted Smartphones. *Sensors*, 18(3):914, 2018.

REFERENCES

Appendix A

Database

A.1 BIM database

BIM database complementary images.

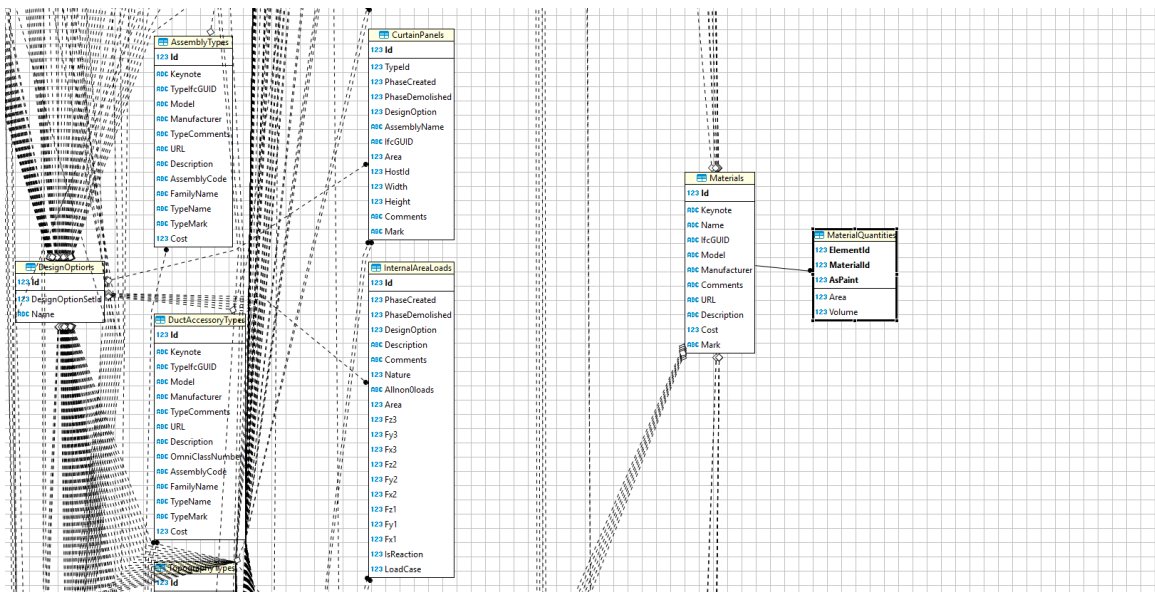


Figure A.1: Revit sample UML diagram cut with an emphasis in the Material table and its relationships. The full UML have 228 tables. It is too large to print with all the table in one single image.

Database

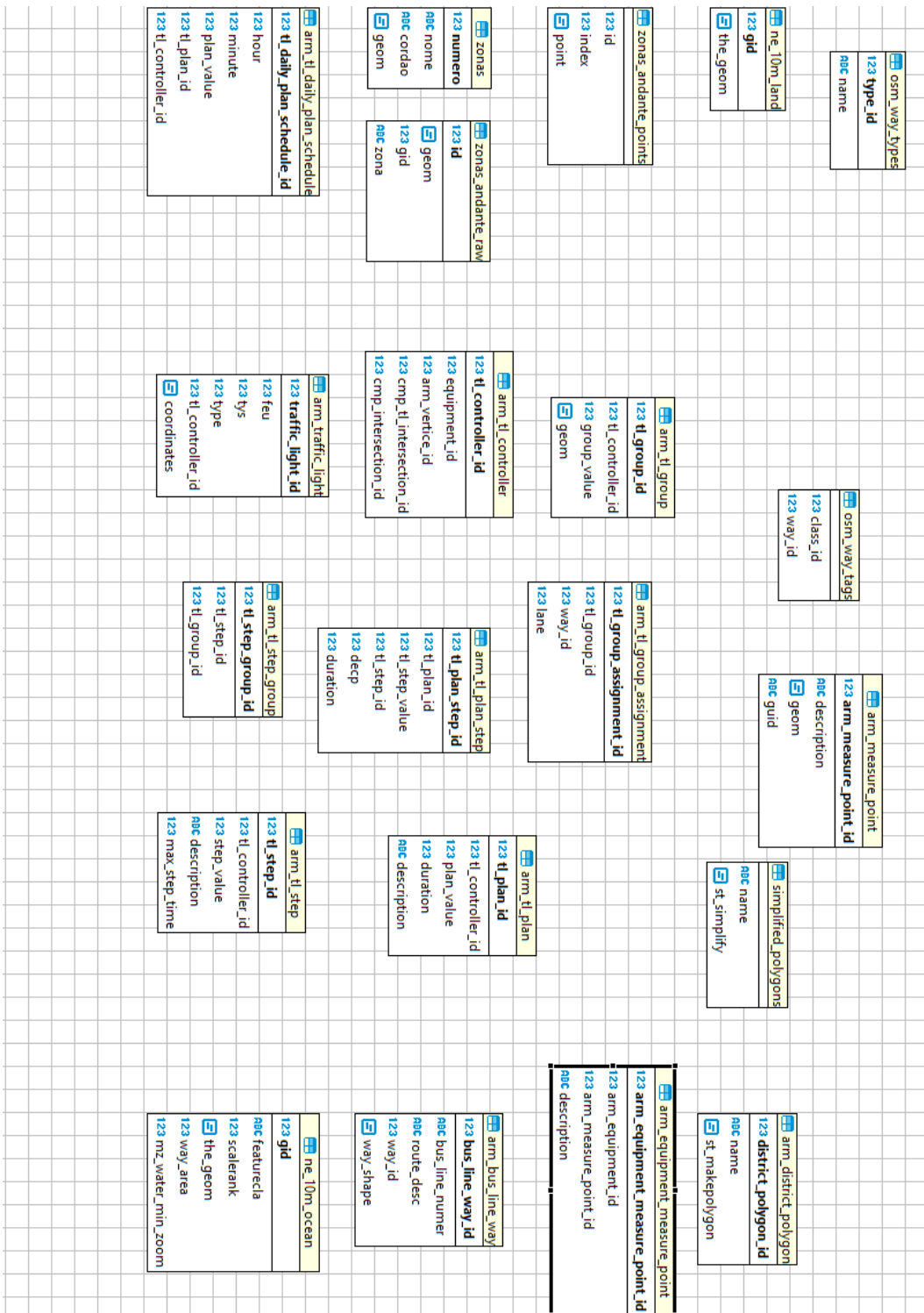


Figure A.2: BIM and GIS used Tables from Armis Database

A.2 Server

A.2.1 Configuration

Listing A.1: pom.xml file.

```
1 <dependencies>
2   <dependency>
3     <groupId>org.springframework.boot</groupId>
4     <artifactId>spring-boot-starter-web</artifactId>
5   </dependency>
6
7   <!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-
8     boot-starter-jdbc -->
9   <dependency>
10    <groupId>org.springframework.boot</groupId>
11    <artifactId>spring-boot-starter-jdbc</artifactId>
12  </dependency>
13
14  <!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-
15    boot-starter-data-jpa -->
16  <dependency>
17    <groupId>org.springframework.boot</groupId>
18    <artifactId>spring-boot-starter-data-jpa</artifactId>
19  </dependency>
20
21  <!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-
22    boot-starter-thymeleaf -->
23  <dependency>
24    <groupId>org.springframework.boot</groupId>
25    <artifactId>spring-boot-starter-thymeleaf</artifactId>
26  </dependency>
27
28  <!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
29  <dependency>
30    <groupId>org.postgresql</groupId>
31    <artifactId>postgresql</artifactId>
32  </dependency>
33
34  <dependency>
35    <groupId>org.postgresql</groupId>
36    <artifactId>postgresql</artifactId>
37    <scope>runtime</scope>
38  </dependency>
39
40  <dependency>
41    <groupId>org.springframework.security</groupId>
42    <artifactId>spring-security-crypto</artifactId>
43  </dependency>
```

Database

```
41
42 <!-- https://mvnrepository.com/artifact/postgresql/postgresql -->
43 <dependency>
44   <groupId>postgresql</groupId>
45   <artifactId>postgresql</artifactId>
46   <version>9.1-901-1.jdbc4</version>
47 </dependency>
48
49 <!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-
50   boot-devtools -->
51 <dependency>
52   <groupId>org.springframework.boot</groupId>
53   <artifactId>spring-boot-devtools</artifactId>
54 </dependency>
55 <dependency>
56   <groupId>com.vividsolutions</groupId>
57   <artifactId>jts</artifactId>
58   <version>1.13</version>
59 </dependency>
60 <dependency>
61   <groupId>org.hibernate</groupId>
62   <artifactId>hibernate-spatial</artifactId>
63   <version>5.2.5.Final</version>
64 </dependency>
65
66 <!-- https://mvnrepository.com/artifact/org.json/json -->
67 <dependency>
68   <groupId>org.json</groupId>
69   <artifactId>json</artifactId>
70   <version>20090211</version>
71 </dependency>
72 </dependencies>
```

A.3 Database RNSFP

A.3.1 BIM data

	crosswalk_id [PK] integer	intersection_id integer	material_id integer
1	1	1	1
2	2	1	1
3	3	1	1
4	4	1	1

Figure A.3: bim_crosswalks table data

	espiras_id [PK] integer	intersection_id integer	material_id integer	tipologia character varying (50)
1	1	1	1313	CD
2	2	1	1313	CD
3	3	1	1313	CD
4	4	1	1313	CD
5	5	1	1313	CD
6	6	1	1313	CD
7	7	1	1313	CD
8	8	1	1313	CD
9	9	1	1313	CD

Figure A.4: bim_espiras table data

Database

id	keynote	name	ifGUID	model	manufacturer	comments	url	description	cost	mark
[PK] integer	character varying	character varying	character varying	character varying	character varying	character varying	character varying	character varying	double precision	character varying
1	1	Crosswalk Material	[null]	[null]	[null]	White Plastic	[null]	Longer Lasting Plastic	50	[null]
2	24	Default	[null]	[null]	[null]	[null]	[null]	[null]	0	[null]
3	25	Default Wall	[null]	[null]	[null]	[null]	[null]	[null]	0	[null]
4	26	Default Roof	[null]	[null]	[null]	[null]	[null]	[null]	0	[null]
5	27	Glass	[null]	[null]	[null]	[null]	[null]	Soda Lime Glass	0	[null]
6	30	Phase - Temporary	[null]	[null]	[null]	[null]	[null]	[null]	0	[null]
7	568	Gypsum Wall Board	[null]	[null]	[null]	[null]	[null]	Gypsum Wall Board	0	[null]
8	569	Ceiling Tile 600 x 1200	[null]	[null]	[null]	[null]	[null]	Acoustic ceiling tile	0	[null]
9	570	Ceiling Tile 600 x 600	[null]	[null]	[null]	[null]	[null]	Acoustic ceiling tile	0	[null]
10	1246	Concrete Masonry U...	[null]	[null]	[null]	[null]	[null]	Concrete masonry un...	0	[null]
11	1247	Rigid insulation	[null]	[null]	[null]	[null]	[null]	Rigid foam insulation ...	0	[null]
12	1248	Damp-proofing	[null]	[null]	[null]	[null]	[null]	Damp-proofing	0	[null]
13	1249	Sand	[null]	[null]	[null]	[null]	[null]	Sand	0	[null]
14	1250	Concrete, Cast In Situ	[null]	[null]	[null]	[null]	[null]	physical material	0	[null]
15	1253	Default Floor	[null]	[null]	[null]	[null]	[null]	[null]	0	[null]
16	1255	Structure, Steel Bar L...	[null]	[null]	[null]	[null]	[null]	Bar joists, thermal air...	0	[null]
17	1256	Concrete, Sand/Cem...	[null]	[null]	[null]	[null]	[null]	Concrete slab topping	0	[null]
18	1257	Site - Hardcore	[null]	[null]	[null]	[null]	[null]	[null]	0	[null]
19	1263	Brick, Common	[null]	[null]	[null]	[null]	[null]	Common brick	0	[null]
20	1264	Metal Stud Layer	[null]	[null]	[null]	[null]	[null]	Light gauge steel fra...	0	[null]
21	1265	Plywood, Sheathing	[null]	[null]	[null]	[null]	[null]	Plywood, sheathing g...	0	[null]
22	1266	Air	[null]	[null]	[null]	[null]	[null]	Air	0	[null]
23	1267	Air Infiltration Barrier	[null]	[null]	[null]	[null]	[null]	physical material	0	[null]
24	1268	Vapour Retarder	[null]	[null]	[null]	[null]	[null]	Polyethylene film me...	0	[null]
25	1269	Cavity Fill	[null]	[null]	[null]	[null]	[null]	Loose fill insulation	0	[null]

Figure A.5: bim_materials table data

id	description
[PK] integer	character varying
1	Cruzamento rua de Nossa Senhora de Fátima

Figure A.6: bim_intersection table data

A.3.2 GIS data

traffic_light_id	feu	tys	type	tl_controller_id	coordinates	signal_orientation
[PK] integer	integer	integer	integer	integer	geometry	integer
1	1	0	10	[null]	1 0101000020E...	244
2	2	1	10	[null]	1 0101000020E...	71
3	3	4	10	[null]	1 0101000020E...	68
4	4	5	10	[null]	1 0101000020E...	297
5	5	6	27	[null]	1 0101000020E...	268
6	6	7	27	[null]	1 0101000020E...	88
7	7	8	27	[null]	1 0101000020E...	178
8	8	9	27	[null]	1 0101000020E...	3
9	9	0	7	0	2 [null]	[null]
10	10	1	7	0	2 [null]	[null]
11	11	2	7	0	2 [null]	[null]
12	12	3	5	23	2 [null]	[null]
13	13	4	5	5	2 [null]	[null]
14	14	5	5	5	2 [null]	[null]

Figure A.7: arm_traffic_light table data

Database

	gis_crosswalk_id [PK] integer	intersection_id integer	crosswalk_coordinates geometry
1	1	1	0101000020E6100000959F...
2	2	1	0101000020E6100000215...
3	3	1	0101000020E6100000B15...
4	4	1	0101000020E6100000A2B...

Figure A.8: gis_crosswalks table data

	gis_espiras_id [PK] integer	gis_espiras_intersection_id integer	espiras_coordinates geometry
1	1	1	0101000020E6100000F...
2	2	1	0101000020E61000009...
3	3	1	0101000020E61000006...
4	4	1	0101000020E61000000...
5	8	1	0101000020E61000000...
6	9	1	0101000020E6100000C...

Figure A.9: gis_espiras table data

A.4 ARIMaRN-PSAL Setup

A.4.1 Wikitude Approach

Wikitude SDK has the Instant Tracking (SLAM) feature that allows us to track the objects in the scene continuously. As the ARIMaRN-PSAL app need the tracking of the ground and the tracking of the traffic signal model, the use of Wikitude can leverage us regarding the time we spent developing. Furthermore, it is one of the most AR SDKs used over the world. Thus, it was tried to use it alongside the Unity engine to develop the app.

A.4.1.1 Unity Setup

First, it was downloaded the Wikitude SDK package for Unity on the Wikitude's website. After, it was created a trial licence to use Wikitude.

Before that, it was used the Instant Tacking scene from the Wikitude packages in the Samples/Scenes folder, with some changes. This scene already has the touch inputs to scale and adjust the position of the objects. However, the scale feature is not necessary for the application.

It was downloaded a free traffic light model to use as the virtual traffic signal in the application [Bru17] and imported the model to the Unity project.

A.4.1.2 Scene

At this moment, the scene's *Game Objects* was changed. In the *Dock* game object, it was removed the *Dock UI* component. Then, it was deleted the *Model Buttons Background*, and all the buttons inside the *Buttons* game object expect the *Clock Button*. The *Dock* game object was enabled to see the buttons in the Unity's Scene view tab.

By now we can add the traffic light model to the scene. After doing it, it was created an empty game object with the *TrafficLight* name, and it was made as the parent of the traffic light model. To appear the traffic light's model correctly in the scene view, we need to put the model and its parent at the coordinates (0,0,0).

To give the physics to the model, the *TrafficLight* game object was selected, and it was added a *Box Collider* component to it. This *collider* is big enough to cover all the traffic light signal. Therefore, we can trigger events when something touches the *collider* as if it was the traffic light model sensing it.

Selecting the *Controller* game object, we have an empty *Element 0* in the *Models* options. The traffic light *prefab* was dragged into the *Element 0*. Now, the *TrafficLight* is associated with the controller.

A.4.1.3 Scripts

To allow the button, that makes appearing the model on the screen, hides and reappears again when the tracking is lost, we need to write a few lines of code. Therefore, it was created a script

to the *TrafficLight prefab*. Furthermore, it is also needed to update the *Instant Tracker Controller* script on the *Controller* game object.

In the *Instant Tracker Controller* script, we need to change the modifier of the *GridRender* from private to public. Thereby, we can now access it from another script. In the *SetSceneActive* function, the the *GridRender* was forced to being enabled before we loop through the models.

Now, in the *TrafficLightController* script, it was created an *InstantTrackerController* and a *GameObject* called *ButtonsParent*, as we can see at Listing A.2. In the *Start* function, it was connected the declared instances variables to the *GameObjects* of the scene. Also, here we state that the *GridRender* is disabled and the *ButtonsParent* is not active, so we cannot see it.

Then, we need more two functions. The *OnEnable* one disables the *InstantTrackerController* object, and the *ButtonsParent* is set as not active. The function is used in case of the tracking is lost, so we cannot put the model on the scene correctly.

The *Start*, *OnEnable*, and *OnDisable* functions are called by Unity in the wanted events, so we do not need to make more code.

Listing A.2: Traffic Light Controller Script

```

1 private InstantTrackingController trackerScript;
2 private GameObject ButtonsParent;
3 void Start () {
4     trackerScript = GameObject.Find("Controller").gameObject.GetComponent<
5         InstantTrackingController>();
6     ButtonsParent = GameObject.Find("Buttons Parent");
7
8     trackerScript._gridRender.enabled = false;
9     ButtonsParent.SetActive(false);
10 }
11 void OnEnable ()
12 {
13     trackerScript._gridRender.enabled = false;
14     ButtonsParent.SetActive(false);
15 }
16 void OnDisable ()
17 {
18     ButtonsParent.SetActive(true);
19 }

```

A.4.1.4 Application's Build

To build the application, we still need to put the Wikitude's licence. Selecting the *WikitudeCamera* game object we can access, in the Unity's Inspector tab, to the Wikitude Licence Key options. My available device to use and test the applications is my One Plus 3 (OP3). Hence, I connected my OP3 to my computer and selected the *Build and Run* option on Unity.

Database

[Back](#)

Instant Tracking

Running without platform assisted tracking support.

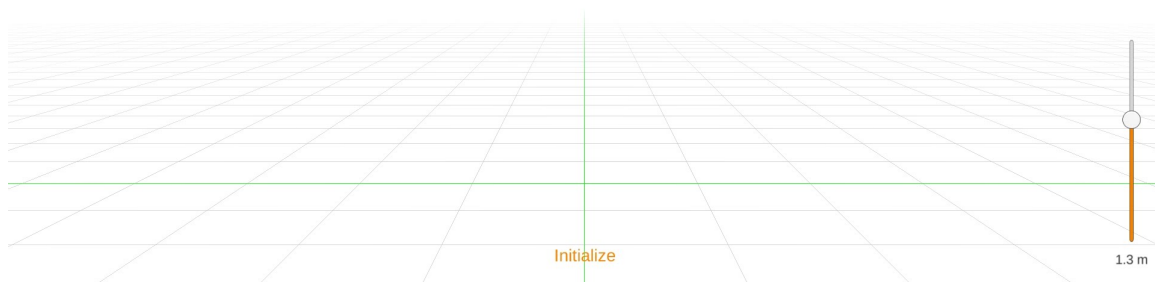


Figure A.10: Wikitude SLAM Application screen.

A.4.1.5 Application's Run

When the app was open, on my OP3, it was allowed the access to the requested permissions by the app. After that, the app starts but something different from the expected appeared on the screen, as we can see in Figure A.10.

The camera does not give an image, and the background is white. We can see the grid for the tracking, but it cannot work because there is no image to analyse. Furthermore, we can see an error message from the Wikitude's Instant Tracking feature, saying it is running on a device that does not support the Instant Tracking feature. The Wikitude requirements listed in their documentation are fulfilled by my OP3. However, the device cannot handle the feature.

A.5 ARIMaRN Setup

A.5.1 Wikitude Approach

As Wikitude JavaScript API supports geolocation services, it was made its setup in the Android Studio IDE. Following the Wikitude's Documentation, it was created a new project in Visual Studio. The necessary project to supports Wikitude is a standard one, no special requirements here. Afterwards, it was downloaded the free version of the Wikitude SDK from its website. The *libs/wikitudesdk.aar* file was copied into the project *libs* folder, and it was declared the necessary dependencies in the *build.gradle* file.

Now we need to add the requested permissions by Wikitude into our *AndroidManifest.xml* file. After, it was inserted one Wikitude example to the application.

Gradle built it without any errors. By now, it is time to test the app. Although Android Studio has an Android Device Emulator Wikitude cannot be executed on it. We need to connect an Android device to build and run the application. It was used the One Plus 3 (OP3), and the Gradle build went fine. However, when the OP3 starts to open the app, it shows an error message in the display before I even give the permissions to the GPS and camera access. The message says that the application is crashing. It was tried with other examples from the Wikitude SDK package, and none of them run correctly. Always getting the same error message. The Wikitude requirements are fulfilled by my OP3. However, although it has all the necessary resources to run Wikitude, we always get the same error.

According to the error and the fact that the code is correct, we can conclude the OP3 cannot support Wikitude on the now Wikitude version. It was already tried to use it with Unity as we saw at A.4.1, and although at that time the app did not crash the instant tracking (SLAM) did not run on the OP3.

Thus, Wikitude has geolocation services, but its developers must give us support to more devices. Otherwise, we cannot use their SDK in our applications, as was my case.

A.5.2 Mapbox Approach

Nowadays, Vuforia SDK is practically the most used AR SDK in the world. Furthermore, in a course unit on the Master in Informatics and Computing Engineering (MIEIC), called Augmented Reality, I already used it, and I got excellent results. Thus, it was decided to use Vuforia since the AR SDKs with geolocation services do not work on the OP3 and also they are not very accurate. Vuforia gives us the AR features. However, we need to develop the location services by ourselves. Therefore, it was tried to use a location-based SDK, Mapbox, alongside Vuforia and Unity to help in that job.

A.5.2.0.1 Unity Setup

To use Mapbox, it was tested if it works correctly with Unity. To accomplish that, it was followed the Mapbox Documentation.

A.5.2.0.2 Scene

In order to test the SDK, it was followed the documentation to create a new map and visualise it as it is supposed to be. Thereby, it was added a *BasicMapVisualizer*, *TerrainFactory*, and *ImageFactory* to the scene as the documentation said. Furthermore, the settings in each game object given by the documentation were used. When it was built, it ran successfully showing the 2D map. Continuing the Mapbox documentation steps, we can add to the scene the *Vector Tile Factory*, *Vector Layer Visualizer*, and *Merged Modifier Stack* game objects. Here, it was used the same settings as the documentation. However, this time the result was not the 3D buildings on the map. The map appears again in a 2D view. After a while trying to overcome these adversities the result was always the same. Thus, I gave up using the Mapbox SDK to help me in the location service part.