

*Master's Thesis*  
*Computer Science*  
*Thesis no: MCS-2011-23*  
*September 2011*



# **Implementing augmented reality for visualisation of virtual buildings using Android**

**Piotr Dąbrowski**

School of Computing  
Blekinge Institute of Technology  
SE – 371 79 Karlskrona  
Sweden

This thesis is submitted to the School of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

**Contact Information:**

Author:

Piotr Dąbrowski

E-mail: [dabrowski.p.w@gmail.com](mailto:dabrowski.p.w@gmail.com)

**University advisor:**

Dr. Charlotte Sennersten

School of Computing

School of Computing  
Blekinge Institute of Technology  
SE – 371 79 Karlskrona  
Sweden

Internet : [www.bth.se/com](http://www.bth.se/com)  
Phone : +46 455 38 50 00  
Fax : +46 455 38 50 57

# ABSTRACT

The mobile phone devices are still developing and they are gaining more functionality and are able to deal with more advanced tasks. One of the technologies timidly trying to approach the mobile phone market is the augmented reality, which does no longer require external equipment to be formed in a programming application. There is a limited number of sources trying to describe the accuracy of augmented reality applications implemented on mobile devices.

Within this study an application of augmented reality visualising virtual models of buildings was implemented on a mobile phone device in order to evaluate the rate of the device explication. Several tests were conducted to evaluate the application total accuracy.

The implemented application was visualising virtual models of the real existing buildings displaying them in the same place the original buildings were. The final position was calculated by the application and the discrepancy of the view between the model and the real building was measured. The results were gathered revealing the application's real accuracy.

For the needs of this study the functional application of augmented reality has been created. The application was implemented on the mobile phone. The results of the application formed the tables with final measurements of accuracy. Also several photographs were taken from the areas of the real existing buildings.

Transferring the functionality of augmented reality from the external devices to mobile phones is possible with some harm to the application accuracy. Visualising building models is one of the possible extensions of the mobile phone market. The improvements in Global Positioning System would significantly improve the application's general accuracy.

**Keywords:** Augmented Reality, Visualising building models, Android, Mobile phones, Smartphones, Global Positioning System, Magnetic sensors, Collada.

# CONTENTS

## Implementing augmented reality for visualisation of virtual buildings using Android:

<b>IMPLEMENTING AUGMENTED REALITY FOR VISUALISATION OF VIRTUAL BUILDINGS USING ANDROID .....</b>	<b>I</b>
<b>ABSTRACT .....</b>	<b>I</b>
<b>CONTENTS .....</b>	<b>II</b>
<b>LIST OF FIGURES.....</b>	<b>IV</b>
<b>LIST OF TABLES.....</b>	<b>V</b>
<b>LIST OF ABBREVIATIONS .....</b>	<b>VI</b>
<b>1 INTRODUCTION .....</b>	<b>1</b>
<b>2 PROBLEM DEFINITION, RISKS AND GOALS.....</b>	<b>4</b>
2.1 PROBLEM AREA.....	4
2.1.1 <i>Android Operation System</i> .....	4
2.1.2 <i>Global Positioning System</i> .....	5
2.1.3 <i>Magnetic sensors</i> .....	7
2.1.4 <i>Interface and camera view</i> .....	10
2.1.5 <i>Reality augmented with building model</i> .....	11
2.2 RESEARCH QUESTIONS .....	12
2.3 RISKS .....	13
2.4 GOALS .....	14
2.4.1 <i>Augmented reality application</i> .....	14
2.4.2 <i>Study on augmented reality</i> .....	14
<b>3 RESEARCH METHODOLOGY .....</b>	<b>19</b>
3.1 FORMULATING THE PROBLEM .....	19
3.2 LITERATURE REVIEW .....	21
3.3 PROJECT DESIGN .....	23
3.4 PROJECT EVALUATION .....	23
3.5 INTERPRETATION AND CONCLUSIONS.....	24
<b>4 THEORETICAL WORK.....</b>	<b>26</b>
4.1 CALCULATING LATITUDINAL DISTANCE.....	28
4.2 CALCULATING LONGITUDINAL DISTANCE .....	29
4.3 BRINGING LATITUDINAL AND LONGITUDINAL DISTANCE TOGETHER .....	30
<b>5 APPLICATION DESCRIPTION.....</b>	<b>31</b>

<b>6</b>	<b>TECHNICAL ISSUES AND SOLUTIONS</b> .....	<b>34</b>
6.1	COLLADA (TRANSITION FORMAT) .....	34
6.2	TEXTURES COLLISION (PROBLEM IN TRANSFERRING BUILDING TEXTURES).....	36
6.3	SENSORS SENSITIVITY AND AZIMUTH SHIFTING.....	39
6.4	DETERMINING VIEWER ALTITUDE .....	39
<b>7</b>	<b>EMPIRICAL TESTS</b> .....	<b>41</b>
<b>8</b>	<b>RESULTS</b> .....	<b>46</b>
<b>9</b>	<b>CONCLUSIONS AND FUTURE WORK</b> .....	<b>56</b>
9.1	CONCLUSIONS.....	56
9.2	FUTURE WORK .....	58
	<b>APPENDIX A</b> .....	<b>61</b>
	<b>APPENDIX B</b> .....	<b>62</b>
	<b>REFERENCES</b> .....	<b>63</b>

# LIST OF FIGURES

Figure 1: GPS receiver block schema (based on Książak, 2002).	6
Figure 2: Axes around the mobile device (Andorid Open Source Project).	8
Figure 3: Azimuth value calculation.	9
Figure 4: Pitch value calculation.	9
Figure 5: Roll value calculation.	10
Figure 6: Sample view of the application.	11
Figure 7: Sample setting the mark-up points discrepancy.	15
Figure 8: Grid of building observations.	16
Figure 9: Division of work process during the study.	210
Figure 10: Relation between the real and calculated distance.	25
Figure 11: The shape of the Earth and consequences (photo by Simmon, 2011).	27
Figure 12: Distance calculation between two points with geographical coordinates.	28
Figure 13: View from camera implemented on AVD.	32
Figure 14: Sample file size changing for avoiding the texture bug.	38
Figure 15: Basic points transition.	38
Figure 16: View of the application building model: south elevation Snapphanevägen, Karlskrona.	41
Figure 17: View of the application building model: north elevation Snapphanevägen, Karlskrona.	42
Figure 18: View of the application building model: ETI Faculty, Gdańsk.	43
Figure 19: View of the application building model: Mechanical Engineering Faculty, Gdańsk.	43
Figure 20: View of the application building model: Golden gate, Gdańsk.	44
Figure 21: View of the application from 40 metres away, Golden gate, Gdańsk.	44
Figure 22: View of the application from 70 metres away, Golden gate, Gdańsk.	45
Figure 23: View of the application from 100 metres away, Golden gate, Gdańsk.	45
Figure 24: View of the ETI faculty as a sample of a building view too near.	48
Figure 25: View of the Mechanical Engineering Faculty as a sample of a building view too far.	49
Figure 26: Displacement in building allocation coming from Collada model, Golden gate, Gdańsk.	49
Figure 27: Graph showing the GPS accuracy progress.	51
Figure 28: Graph showing distribution of answers to Q1.	53
Figure 29: Graph showing distribution of answers to Q3.	53
Figure 30: Result of the application from, Mechanics engineering Faculty, Gdańsk.	54
Figure 31: Result of the application from, ETI Faculty, Gdańsk.	54
Figure 32: Result of the application from, Golden gate, Gdańsk.	55

## LIST OF TABLES

Table 1: Pattern for preparing study about the building: .....	16
Table 2: Keyword data search results: .....	22
Table 3: Percentage share of found sources: .....	22
Table 4: Study area search results: .....	23
Table 5: Number of first sources checked from database for each query:.....	23
Table 6: Result of measuring ETI Faculty model discrepancy:.....	46
Table 7: Result of measuring Mechanical Engineering Faculty model discrepancy:.....	47
Table 8: Result of measuring Golden gate model discrepancy: .....	47
Table 9: The average values of the presented measurements:.....	47
Table 10: The difference between the distance measured manually and by the application: .	48
Table 11: The minimum and the maximum results of the measurements in total: .....	50
Table 12: Results of the GPS receiver accuracy test: .....	51
Table 13: Results of the user's survey: .....	52

## **LIST OF ABBREVIATIONS**

3D – three dimensional  
ADB – Android debug bridge  
AFSPC – Air force space command  
AGPS – Assisted global positioning system  
API – Application programming interface  
AR – Augmented reality  
AVD – Android virtual device  
BTS – Base transceiver station  
CEP – Circular error probable  
DDMS – Dalvik debug monitor service  
FOC – Full operative capability  
GPS – Global positioning system  
GSM – Global System for Mobile  
HMD – Head mounted display  
IDE – Integrated development environment  
IOC – Initial operational capability  
MCS – Master control station  
OHA – Open Handset Alliance  
OS – Operating system  
PC – Personal computer  
PPS – Precise positioning service  
R.Q. – Research question  
SA – Selective availability  
SDK – Software development kit  
TTFF – Time to first fix

# 1 INTRODUCTION

Since beginning the mobile phone has been a revolutionary idea. The mobile phone does not just serve for phoning nor for sending text messages, as it has far more functions, i.e.: internet wi-fi access, large and high-resolution touch displays, memory card input, polyphonic speakers and camera. All these additional functions are now just standard when using a decent mobile phone.

In 1992 IBM designed an IBM Simon, which was the first advanced cellular phone. A new era of mobile devices started, an era of Smartphones (Boyes, Melvin, 2010). Smartphones are a product line of phones that are designed to offer much more than standard feature phones. One of the biggest distinction is that Smartphones are always supplied with a full mobile operating system, i.e.: *Apple iOS*, *Google Android*, *Microsoft Windows Phone 7*, *Symbian*, *BlackBerry OS* or *Maemo* (Falaki et al., 2010). The constant development of the systems mentioned above attracts researchers, because mobile devices are becoming capable of dealing with more and more complicated operations and calculations allowing to interpret human voice or making instant connection with *GoogleMaps* displaying phones current position. Mobile phone development is a very profitable phenomenon, because projects do not necessarily have to require a lot of separate equipment, as plenty of the facilities are accessible to mobile device users. One of the techniques that can benefit on the phone market expansion is Augmented Reality (Wagner, Gruber, Schmalstieg, 2008).

*Augmented Reality (AR)* is a technique supplementing the 3D view of a real environment with the additional virtual items. It is well used in medicine, visualisation, path planning, military operations and still gets plenty new appliances. Since Ronald Azuma presented the definition of “augmented reality” in 1997 in his Survey of augmented reality the number of designed applications is constantly increasing. Still the method of AR is perceived as a future prospect rather than being suitable for everyday-use applications, including practical services used as supplement for industrial services. One of the major factors causing this is that most of the AR projects were done for military services and so most of the achievements on that field are either confidential or slowly adapted for civil activities (Julier, Baillot, Lanzagorta, Brown, Rosenblum, 2000). Within a few years we will observe how AR fits into industrial market and either fills a useful addition to human everyday life or gets rejected as too expensive (Azuma, 1997).

Today, there is a constant research development in the area of civil AR and a clear trend in minimising the display sizes and external equipment required by the applications. The basic idea of AR requires just a display and a calculation system to work passable (Feiner, Höllerer, 2004). Obviously the better display equipment there is, the better feeling of immersion in the new reality the user gets. On the other hand redundant equipment can spoil the impression of simplicity so researchers in the area of augmented reality need to concentrate on two parallel goals. Firstly it is to minimise the necessary number of external devices and secondly to sustain a proper level of calculation quality.

The first AR projects involved extensive equipment like *head-worn-displays* (Webster, Feiner, MacIntyre, Massie, Krueger, 1996) or, in other words, *head-mounted displays (HMD)* (Kato, Billinghurst, 1999) which within the possibilities of smartphones are no longer needed or at least can be successfully replaced by mobile devices. Portability, attractiveness, growing popularity of Smartphones in general and a great breakthrough, i.e.: enabling phones to receive Global Positioning System (GPS) signals, resulted in increasing number of different AR applications arising (Suomela, 2006) (Whitfield, 2010), from navigation projects like *Wikitude Drive*, *AugSatNav*, *3D AR Compass* by applications supplying user with additional terrain information like: *Wikitude World Browser*, *Layar Reality Browser*, *Weather Reality*, *Space InvadAR* to AR social networking projects like: *TagWhat* or *Tweeps Around*. Development of AR projects has different trends. There are a few applications visualising buildings but mostly they are geographically restricted and devoted to one private building. A good example is the *Fraunhofer IGD* application

commemorating the Berlin Wall which shows development of the city of Berlin in the period of 1940-2008 (Zöllner, 2009) or a project known as *AR4BC (Augmented Reality for Building and Construction)* which started with visualising *Skanska's* new office in Helsinki. Those projects, however are not for public use, rather for internal commercial showcasing. Both achieved great successes and required a lot of prerequisites. *AR4BC* was designed with a head-mounted display and including a lot of pre-calculations, thus it was very precise and worked out with great accuracy (Woodward, 2009). The *Fraunhofer IGD* application was a project designed for *iPhones* ordered by German Federal Ministry of Education and Research with some advanced features. It is still only narrowed to Berlin area with no plugins predicted for single users. Basically, there is no realtime AR application visualising buildings that are not geographically restricted, like *GoogleMaps* for visualising maps.

This proves there is a gap existing in the market of augmented reality applications. There are no public applications allowing to prepare building models and view them in augmented reality, and only a few dedicated to given areas. Despite unstopped evolution in the field of graphical libraries and new file formats supported by much more complex three-dimensional viewing programs there is yet a limited number of projects allowing to view those models in real environment. We can store the models in different files as *Wavefront objects (.obj)* or as *3d studio files (.3ds)*. Another option is using *Scalable vector graphics (.svg)* or translatable format known as *Collada (.dae)* which can be operated by many programs like *Blender*, *3dStudioMax* or *GoogleSketchup*. One also has a chance to view a building model created in KML format in *GoogleEarth* using 3d buildings option so then one is able to visualise a gathered set of buildings from one neighbourhood in different angles. The models are stored in *GoogleWarehouse*, a tool created by *Google* in order to gather *Sketchup* projects for *GoogleEarth*, so they are free for downloading and looking into. The whole application of *GoogleEarth* has achieved a great success and viewing 3d buildings is one of its most remarkable options. There were also some single prototypes of application using *GoogleEarth* facilities, like prototype created by the *VTT Technical Research Centre of Finland* successfully presenting the concert hall in Helsinki in augmented reality using markerless tracking. (Honkamaa P., Siltanen S., Jäppinen J., Woodward C., Korkalo O., 2007). An open question is why researchers are still not using the *GoogleWarehouse* capabilities in 3D AR applications at a larger scale.

Augmented reality is still rather a fresh idea. There is just a small number of public articles dedicated to the AR subject despite the fact that it is known that some applications have been successfully implemented for military purposes (Azuma, Baillet et al., 2001). Slowly, inventions and new trends are presented publicly too. It results in a rapid growth of the applications level of advance. Right now it is knowledgeable that there are different methods of calibration and some different approaches in implementation. Low budget applications operate on a GPS device and compasses. Commercial projects rely on bigger displays or HMDs, mentioned before, but mostly they use separate engines operating with image retrieval, based on terrain recognition algorithms. Combination of both is rumoured to give the best results in accuracy, however there are no studies that show the actual achievements. There is also no wide solution presented in comparing discrepancies between the desired position of the building and its appearance in the AR system, as most of such projects focus on tracking sensors (You, Neumann, Azuma, 1999).

Aim of this thesis is to present a clarified evaluation of the augmented reality real-time application implemented on a mobile phone. In order to achieve this aim there are several prerequisites. Most necessary is to create an application fulfilling all the requirements of an AR project being presented here. The basic idea of the application is to pick a building model from *GoogleWarehouse* stock and draw it using tools available for smartphone. Each such building stored in *GoogleWarehouse* has its exact position in the real environment. The plan for the application in this case is also to be able to view the building via the mobile display using graphical library possibilities. The application will be designed for a phone equipped with *Android OS (Operating System)* which has all the necessary facilities for such an aim. Of course applying AR technique means that the application will have to give repetitive results and will be programmed in the way to be able to display the virtual building

in the same place as the already existing one, but instead using digital data, i.e.: GPS coordinates and magnetic sensors position working as an input data for the algorithm. All the information about data received this way should be constantly available for the user by viewing them on the display. By augmented reality within such an application we understand the 3D model of the building added on the top of the view coming from the mobile phone via its camera. Of course the way in which the building is positioned and displayed depends on the position and of the user. A desired user action is to aim with the mobile exactly in the same direction the user wants the augmented building to appear, the distance or the depth has to be considered in relation to size constancy on the display. The further away the user goes, the smaller the model shape will occur on the display, accordingly to human vision cues in relation to the real building. The more the user misses the direction of the building to left or right the more the model is misplaced.

Operating the application, after it will be created, will allow measuring its effectiveness. For determining the quality of the AR application it has to be based via accuracy calculation. The study related to the application implemented will be based on measuring the discrepancy between the real existing building and its model displayed on the mobile phone. A GPS receiver is known for its deviation depending on different conditions like weather, daytime or user's position. To meet the acceptance of application quality and satisfying one's needs the application has to be tested using different points of view. For a proper test of the presented application we will choose view characteristic points of the real existing building. Those points, of course, are also mapped on the building model using graphical textures. An example of such a point is: a building vertex, a window frame, a doorframe, a mark on the wall. Such points will have to be chosen from every elevation. This will allow tracking of every single part of the building. The aim of the test is to spot the discrepancy between the building in reality and its model in augmented reality from 6 different angles for a single building. Measurements will be taken four times from each angle, from different distances: 10, 40, 70 and 100 metres and also written down in metres in a table designed for one single model. Since one might expect different results from different angles, the result depends on several conditions, like GPS results, magnetic sensors distortions, calculation speed of mobile phone, the result of such an inquiry should allow to draw conclusions about the advantages and disadvantages of implementing augmented reality applications on mobile devices, and possible improvements in the future.

## 2 PROBLEM DEFINITION, RISKS AND GOALS

There are two aims of the study i.e.: creating the useful augmented reality application allowing to use a mobile phone as a portable 3-dimensional browser displaying buildings and, because it is still a developing technique with just a few researches being done, expressing the threats of such an application. Augmented reality is still not well described in the technical literature and when it is done it mostly concerns obsolete techniques rather than the mobile equipment. In that case there still is a number of gaps that should be filled. The major one is the accuracy of such an application calculating the position of the implemented item and indicating its position in the “augmented reality”. There are no effective studies measuring the full accuracy of the augmented reality techniques and that is why this study will try to answer questions about this matter by an organised measuring process.

Creating the AR application is not the most important aim of this study, however it is a major prerequisite. The assumption of the application is benefiting from the clarity of the models presented in *GoogleWarehouse*. Optimally the models should become real-time models of buildings, displayed on the mobile phone’s screen chosen by the user and prepared by the programmer place in the world. The creation of such an application should allow giving answers to all the concerns on beforehand. It should also contribute in bringing new questions and inspire for drawing conclusions. Concerning the application and its implementation there are plenty of concerns even before the beginning. One might fear if a mobile device is capable of doing all the necessary calculations in real-time as the implementation requires. The same concerns may arise while referring to magnetic sensors and their dependencies between its deviation and exact position of the building in the environment. GPS is in this case a civil standard type and it is a technique standard well known for its inaccuracies. These are just first few concerns, still there are more and before creating an application, it is good to get to know its potential weaknesses by defining the problem area.

### 2.1 Problem area

A list of problems is usually concerned with used and chosen techniques, systems or devices, e.g.: the magnetic sensors might be too sensitive for the application, which can make the model of the applied building floating instead of being displayed smoothly. Same concerns we come across when the capabilities of the mobile phone are too slow for calculations required by the algorithm. The GPS sensors might bring too bad accuracy which can cause wrong conversion and then the building might be applied with wrong offset in accordance to intended place in the real environment. This means that each aspect requires separate care and brings different challenges. In order to avoid inconveniences, one has to deepen one’s knowledge in the featured issues step by step, so they can bring a full view on what is necessary for the augmented reality application to work optimally.

#### 2.1.1 Android Operation System

Android is most commonly called a software stack designed for mobile phones but in this case we can rather refer to it, as an *Operating System (OS)*. All the necessary tools, programming libraries and the *Application Programming Interfaces (APIs)* for developers are stored in *official Software Development Kit (SDK)*. According to “Android developers” (Android Open Source Group) website Android phone typical specification is:<sup>1</sup>

---

<sup>1</sup> Android Open Source Project, *Dev-Guide section: What is Android?*, in: <http://developer.android.com/guide/basics/what-is-android.html>

- **Application framework** enabling reuse and replacement of components.
- **Dalvik virtual machine** optimised for mobile devices.
- Integrated browser based on the open source WebKit engine.
- **Optimised graphics** powered by a custom 2D graphics library; 3D engine based on the OpenGL ES 1.0 specification (hardware acceleration optional).
- SQLite for structured data storage.
- Media support for common audio, video and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- **GSM Telephony** (hardware dependent).
- Bluetooth, **GPS, compass** and accelerometer (hardware dependent).
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling and a plugin for the Eclipse IDE.

With above mentioned content and on top of this adding a camera device Android appears to have all the requirements to prepare a proper augmented reality application using devices with his Operating System installed.

## 2.1.2 Global Positioning System

Global Positioning System mostly called by its abbreviation GPS is a system entirely created by United States Department of Defence. GPS was originally based on a network of 24 satellites orbiting twice a day over the Earth and sending radio signals to terrestrial receivers. It was eventually launched in 1973 after some earlier projects starting in the 1960s. The whole system was initially created for military purpose but since 1993 it was available for civil operations also, the difference is in frequency of the radio signal sent. Military frequency is 1227,60 Mhz, civil frequency is allocated on 1575,42 Mhz (Bao-Yen Tsui J., 2000). Due to its industrial benefits GPS system is meant to remain free of any fees for public use. According to its assumption it shall work in any weather condition the signal shall cover the whole globe and thus shall be available to use in every place of the world for twenty four hours a day. The structure of the GPS architecture is separated into three subdivision groups called segments, i.e.: space, control and user segment (Kaplan, Hegarty, 2006). I will shortly present these segments that cover my work directly or indirectly and afterwards I present the concerns.

### Space segment

The space segment is based on GPS satellites. When the system was introduced there were 24 satellites moving around 3 orbits at an altitude of 26 560 kilometres. Then the number of orbits were increased to 6 with 4 satellites on each of them. Old satellites are being constantly replaced so the amount is not rigid and is varying from 24 to 32. At the moment there are 31 active and broadcasting satellites and 2 treated as spare ones (Bao-Yen Tsui J., 2000). Each satellite needs 11 hours 58 minutes to orbit around the globe once (Kaplan, Hegarty, 2006). It is because of the space segment that we can receive signals containing information about our current position and the time of coordinate calculation depends on the number of satellites which the mobile device can reach.

### Control segment

This segment is responsible for organising communication between two other segments and consists of 12 ground stations observing the satellites. All the stations track satellites flight paths and thus gathered information are being sent to Master Control Station (MCS) where they are operated and navigational updates are sent back to satellites using ground antennas. Due to their individual tasks the stations are divided into four groups:

- 1 Master Control Station at the American Air Force Base Shriever in Colorado Springs, United States.
- 1 Alternate Master Control Station
- 4 dedicated ground antennas (in Kwajalein, Ascension Island, Diego Garcia and Cape Canaveral)
- 6 dedicated monitor stations (in United States, Ecuador, Argentina, Great Britain, Bahrain, Australia)

This kind of solution allows satellites to be synchronised up to single nanoseconds (Kaplan, Hegarty, 2006).

### User segment

This is the lowest segment of the GPS system. It consists of the GPS receivers. United States and their allied military forces are using Precise Positioning Service and for civil users the Standard Positioning Service is available. The number of civil receivers is already estimated to tens of millions and is constantly growing.

All the receiving devices have different models, shapes and hardwares and accuracy but they use the same composition schema (Figure 1) (Kaplan, Hegarty, 2006).

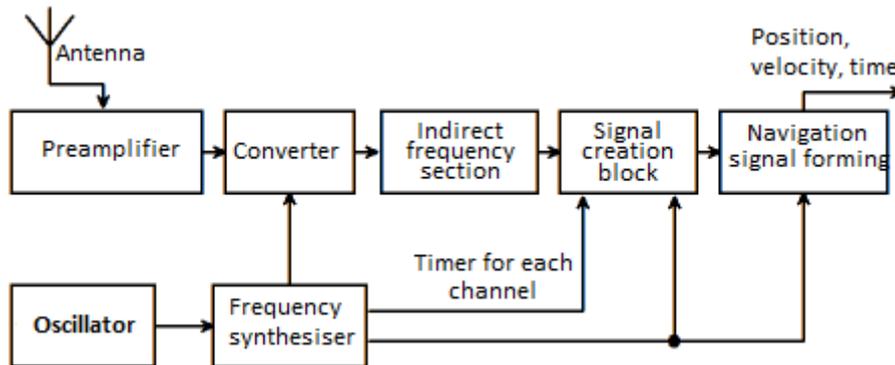


Figure 1: GPS receiver block schema (based on Książak, 2002).

An important circumstance is the number of channels inside a GPS receiver. Each channel allows receiving signal from one satellite at a time. Since we are able to receive signals from 9 satellites and even more, for better accuracy the newest receivers have 12 channels and even up to 20 (Książak, 2002).

Mobile phones with GPS navigation are part of the User segment. Some phones however do not work like standard GPS receivers. Depending on devices with GPS navigation and their mobile operators, the devices might be supported by Assisted GPS (A-GPS). A-GPS is a technique supplementing GPS designed for mobile phones. Instead of navigating by just using radio signals mobile phones with A-GPS also use network capacities when encountering any difficulties. Possible appliance of A-GPS is delay in receiving first GPS signal, technically named as TTFF (time-to-first-fix) or while having poor weather conditions especially in urban areas where A-GPS is particularly successful due to bigger number of network antennas or BTSs (Base transceiver station). A-GPS does not give the same accuracy as GPS but at least it is a passable technique when GPS does not work properly (Djuknic, Richton, 2001).

In 1993 there was a breakthrough in development of positioning systems, in the 8th of December the same year an Initial Operational Capability (IOC) was launched. However it was until April 1995 when the Air Force Space Command (AFSPC) declared Full Operative Capability (FOC) it started with full availability of the system for public users including some military services like Precise Positioning Service (PPS). Even though the system did not guarantee satisfying accuracy of the coordinate system as due to American Congress decision it was limited just to 40 meters precision. Such action was intentional and

caused by interfering the system named Selective Availability (SA). In May 2000 Selective Availability was officially turned off allowing civil users to use global non-distorted signal (Adrados, Girard, Gendner, Janeau, 2002).

Disabling Selective Availability does not necessarily mean that nowadays GPS receivers show exact coordinates within single nanometre. GPS navigation accuracy still remains being one of the most confusing part of the whole system. It is because there is yet no standard measuring GPS inexactness. Circular Error Probable (CEP), which is a way of measuring ballistics in military science, is often used but it only shows percentage fall of a given distance scope. We can be informed that CEP in 1-2 metres is 65% which means that around 65 of 100 measurements are exact within the interval of 1-2 metres. Still we have no information about the other 35% results which basically is useless and show no direct information about the used device (van Diggelen, 1998). Most of the producers claim that GPS devices measurements have single metres inaccuracy. For mobile phones this indication is getting worse and for results pointed by AGPS might even reach 30-50 metres inexactness (Djuknic, Richton, 2001).

Knowledge about GPS specification is very important from the point of view of augmented reality development with mobile devices. According to Richard Lewis from RLA Geosystems and his study about Global Positioning System he says “Technology has reduced the effect of multipath and GPS data gathering capabilities are being strengthened. The user must be informed, however, about GPS advantages and disadvantages” and so he lists a group of both (Lewis, 1998):

#### **Advantages:**

- Spatial and tabular data are collected simultaneously.
- Position accuracy is superior to conventional methods.
- Coordinate system and reference datum can be changed.
- GIS conversion is simple.
- Data collection costs are lower than conventional methods.
- Feature visual inspection is possible while gathering data.

#### **Disadvantages:**

- Requires training and retraining as technology changes.
- Urban canyon buildings can block satellite signals.
- Heavy foliage and thick branched trees can attenuate and/or block satellite signals.
- Multi-path reflective signals can make data inaccurate.
- Requires careful attention to system configuration and data collection standards and procedures.

### **2.1.3 Magnetic sensors**

In order to follow the view on which the mobile phone’s camera is pointed the application needs to listen to magnetic sensors indications. Android Operation System supports several types of sensors like: accelerometer, gyroscope, orientation or proximity.

For purpose of AR application only orientation sensors are required as we need to detect the direction of the phone view. Being able to read information gathered by sensors allows describing the visualised building position in the “augmented” environment. What it means is that by operating orientation sensors one will change the position of the building to

right, left, up or down exactly according to position of the mobile phone. Sensor information is written in three split values:<sup>2</sup>

- **Azimuth** – measured in degrees from  $0^{\circ}$  to  $360^{\circ}$ . It is describing the angle between the X axis of the phone direction and the north direction. This means that azimuth's value 0 indicates that the phone is turned to north direction, 90 means east direction and so on (Figure 3).
- **Pitch** – measured in degrees. This indication shows angle between the Y axis of the mobile phone and horizontal position. Figures may vary from  $-180^{\circ}$  to  $180^{\circ}$  (Figure 4).
- **Roll** – also measured in degrees and it also shows relation between the horizontal position of the ground but this time it is measuring the angle between the X axis and horizon. This angle varies from  $-90^{\circ}$  to  $90^{\circ}$  (Figure 5).

Y axis of the phone is the one along the tall side of the phone and axis X goes along wide side of the phone (Figure 2).

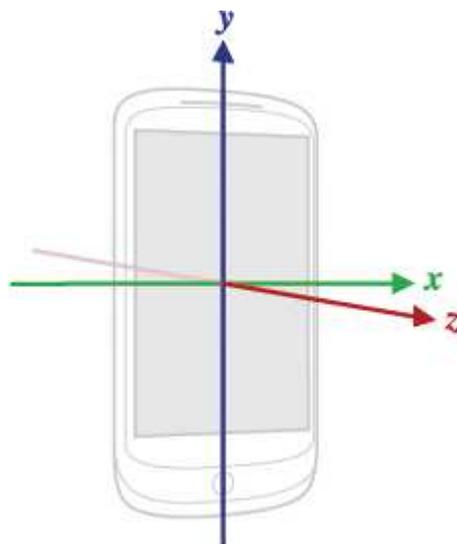


Figure 2: Axes around the mobile device (Andorid Open Source Project).

Orientation sensors is also used in Compass applications available for Android and in fact, especially for determining the azimuth orientation sensors they behave like compasses which means they have disturbing tendency to malfunction when being near to large metal objects. Application might have different future appliances as it might serve to real estate developers to view its future prospects but, especially during studies over the application it will be mainly used in wide urban spaces when it is highly possible to get the opposition of metal objects. This unfortunate feature may influence on the results achieved while trying AR application so one needs to be aware of that.

---

<sup>2</sup> Android Open Source Project, *SensorEvent: Class overview* in:  
<http://developer.android.com/reference/android/hardware/SensorEvent.html>



Figure 3: Azimuth value calculation.



Figure 4: Pitch value calculation.



Figure 5: Roll value calculation.

#### 2.1.4 Interface and camera view

Establishing a proper interface is always vital point for every project. In this project that will attach a background view coming from the camera, a building model displayed using a graphical library and also some additional information displayed for the viewer it is vital to make some major decisions already that will meet the goals set out in the beginning. Information gathered by the mobile device that need to be displayed to the viewer is: longitude, latitude and altitude coming from GPS sensors, azimuth, pitch and roll from magnetic sensors. During the application-work-progress an idea came across of also displaying distance between the user and the place where the nearest building was. The value of distance is calculated in metres.

When the final idea of creating just small text labels the handling of all the inscription and values came on. They were placed in 2 lines at the bottom of the screen to make it concise and straight as the view cannot be covered with too much text (Figure 6).

Preparing the view for the user is also related with one very important problem, which is screen orientation. The Android SDK support using several types of screen orientations: *unspecified*, *user*, *behind*, *landscape*, *portrait*, *reverseLandscape*, *reversePortrait*, *sensorLandscape*, *reversePortrait*, *sensorLandscape*, *sensorPortrait*, *sensor*, *fullSensor*, *nonsensor*. From those listed, two are most important and they are: *landscape* and *portrait* view. The *landscape* orientation is when display is wider than it is tall and contrary, and *portrait* screen is when the display is taller than it is wide. By default Android OS uses *unspecified* setting which works differently depending on the device. For instance it switches between *portrait* and *landscape* view when the keyboard is rolled or unrolled. Designed AR application however uses camera view and is prepared for displaying buildings which means it is better to get a wide view of presented surface which means the chosen screen orientation is *landscape*.



Figure 6: Sample view of the application.

Unfortunately this brings some further complications. As it was mentioned before sensors are calculating phone's position between north direction or horizon line and phone axis. Axes are constant and X axis goes along wide side of the phone, axis Y along tall side of the phone. Due to the fact that in landscape view user holds the tall side along horizon line and X axis along vertical line the axes have been switched comparing to standard approach. This also implicates that magnetic sensors indications might be switched. The values are correct but one has to keep in mind that X and Y axes mentioned in standard are switched or would have to check them on the phone side, remembering that pitch works along the tall side and roll goes along wide side. Azimuth for instance, because of this rotation is changed by 90 degrees comparing to what the phone is being aimed at. One needs to know this while interpreting the results.

### 2.1.5 Reality augmented with building model

Augmented reality requires several conditions, like knowing where you are and what you are looking at. When preparing an environment enriched with building models one also has to organise a building model and its position. A GPS receiver is responsible for indicating current user position and magnetic sensors, of course for describing current users point of view. The buildings stored in GoogleWarehouse are available for downloading in different formats. Most common are: Google Sketchup version 7 and 6 (stored in .skp files), GoogleEarth 4 files (in .kmz format) and COLLADA format. It is saved in .zip package and inside it contains two important files and one important folder, which are:

- *doc.kml* file containing view information and settings when opening in GoogleEarth
- *.dae* file (usually named *warehouse\_model.dae*) all the building details stored in COLLADA format
- *images* folder storing all the textures pictures to which building model refer

When it comes to interpreting those files the primary one is doc.kml. It contains basic information about the model and GoogleEarth camera interpretation. From the point of view of the building model the most important information is latitude, longitude and altitude value inside the <Model> tag. The model contains just one reference point and then it is

linked to .dae file which includes all the building details according to Collada format organisation.

Collada is an open XML standard for interactive 3d building models, also known as “Collaborative design activity”. It allows storing building models for graphical applications by gathering information about every mesh of the building. While describing building’s geometry, the Collada file includes information about:

- position array
- normal array
- uv array

Uv array describes texture transformation from flat image to 3D model and normal array is only used for graphical environment, not really essential for the model itself. From all three position arrays the most vital is for establishing a building model. It contains information about buildings coordinates in 3D environment in all three axes (marked as x, y and z positions). Unit of the values mentioned in position array depends on what is written in <unit> tag. Most common value is meter. All the points in position array are interpreted in reference with the coordinate point from doc.kml file. It means that Collada format store information about distance relation between each point and doc.kml gives an exact position of the building in the environment. Basically, application will just calculate the distance between this reference point and current GPS indication given in geographical degrees and draw the points according to view angle achieved by magnetic sensors results.

## 2.2 Research questions

The augmented reality field shows there are a few unresolved concerns about the whole technique. It is because there are still plenty of fields in which this method has not been implemented and tested properly and in others achieved results were unsatisfactory. Also a lot of remarks concerning appliance of augmented reality in displaying building models. The 3-dimensionality brings data inaccuracy and is more vital than in applications just gathering information about flat surfaces. Viewing the building model means that a lot of effort needs to be put into the way the building is displayed, does it keep the horizontal position and does it supply user with satisfactory quality of the model? Due to all the mentioned concerns, the following research questions (noted as RQ’s) are stated:

- RQ1: Is it possible to reach a good level of accuracy and quality in the AR application only using commonly available mobile tools?
- RQ2: How much, from the user’s point of view, does GPS and magnetic compass sensors inaccuracy affect the AR effect?
- RQ3: Does application inaccuracy differ depending on the angle and distance in relation to real building?
- RQ 4: Can AR application with Google Warehouse’s building models guarantee a satisfying quality of view?

The questions were stated allowing to reveal major aspects of the issue: is it possible to obtain satisfying data calculation accuracy, does it depend on sub-devices and is it visible while changing either the position or the angle. The last aspect concerning the model quality. Answering these questions would point out what is already accessible for a developer and what, on the other hand, is still missing. Relevant actions had to be undertaken in order to gather complete answers about all of the mentioned questions. Also the necessity of answering such questions has its toll on the study which is shaped exactly in a form allowing to point out the answers.

## 2.3 Risks

Knowing much more about the technologies and its weaknesses allows defining application basic risks. In this context the risks are not only about slowing down the application implementation or threatening the whole process of its creation, but are also related to its accuracy which is most critical for this application. Above all it is an augmented reality application which means it cannot work out without proper imposition of the model on exact place. This would mean that basically the application has no sense as its main aim has not been achieved. Few major risks have been accounted:

- GPS accuracy is too low.

As we know GPS, it can bring very limited data exactness. From a programmer point of view there is a possibility to call GPS position listener function once user changes position within a single meter but according to documentation accuracy cannot be properly measured which means that it might vary within few metres, which for application visualising small buildings might be very disturbing.

- Magnetic sensors are too sensitive.

While displaying the building model during application work there are two important issues: where the user currently is and where he is aiming the phone. The exact position of the applied building depends on the phone direction depended on users behaviour. As we know the operating system of the mobile phone registers this position using 3 values, azimuth, pitch and roll and those values are further interpreted by the rotation algorithms. It means by just slight incompatibility between where phone is exactly directed and what the sensor say we can get a totally spoiled effect of building imposition. Unfortunately depending on phone hardware sensors it might work differently not being able to indicate its position precisely. When this happens either the sensors are showing improper values misleading the algorithms or the values are being changed too rapidly (several times per second) which results with the building changing its position in reaction despite the fact that user remained still and GPS indications did not change at all.

- Building model cannot be transferred to Android graphic library.

The building model was prepared using Google SketchUp tools and stored using Collada portable form. However, Collada is just an XML-based script and when using Android phones layout one is only able to make use of the limited number of graphical tools. For the purpose of the application we will use OpenGL ES 2.0 graphical library available since Android's version 2.2. This way does not support importing XML exactly so every single feature needs to be interpreted by programmer, some cases like important textures works differently and the results might be different and efforts are pretty time-consuming.

- Calculations are too complex for the device.

During its process, the AR application on a mobile phone has to cope with GPS and magnetic sensors data, calculate if position and direction is right and then draw a building model. A properly displayed building model assumes calculation of distance between geographical coordinates, organising building textures imposition and organising the whole 3d world. Whereas the mobile phone is a device of limited calculation capabilities. It might happen that the building model will not be displayed stable, because of calculations occupying mobile device memory and by this the building would not be positioned accurately.

- Building position is distorted by changes of distance or angle.

Because of all issues mentioned before, like low GPS or magnetic sensors accuracy resulting in constant updates of the user's position that can bring some irregularities in to the received values and also system issues with the calculated system the position of the building on the screen can change too rapidly or too slowly. Unfortunately the application developer has very little to do as the undesired effect is a result of all the devices and techniques malfunctioning the composition.

## 2.4 Goals

There is one major aim of this study which is to show current possibilities of how to visualise a building with help of augmented reality technique. In order to do this a study about an application will be performed. The target of the mentioned study is to indicate the quality of the actual augmented reality effect. The first step is also a second major aim of the study which is to create an application for augmented reality. The application shall display a building model on a screen on a mobile phone in the place of a real existing building so that one will be able to see the effect of the new reality and also easily compare discrepancies between elevations of both.

For better organisation the whole application and study assumptions, requirements and features are mentioned below:

### 2.4.1 Augmented reality application

As mentioned before, the application has to be implemented on the Java Android environment, thus it will require a phone with the Android Operation System supplied with a camera, magnetic- and GPS sensors. The user of the application will be given information about his/her current position coming from the GPS sensor. Namely the user will be informed about their longitude, latitude and altitude. Also constant information about the direction in which the phone is directed will be displayed on the screen.

Apart from technical information the application will display previously prepared building model coming from *GoogleWarehouse*. All the technical details of the building like its coordinates, textures and more detailed features are written and stored inside of the application. The application is supposed to work in following way if user does not aim successfully at the building s/he will just receive an empty camera view (with just few text labels of information) and if one aims at the exact place the building will be displayed giving best possible accuracy exactly at the desired place.

No separate devices are required, just the equipment of a standard smartphone with the Android operating system is necessary. There are no other actions from the user required like using keyboard or touch display. The only successful behaviour is tracing the terrain when the building was arranged and aiming the exact area.

Creating the application is also a direct step towards forming the study about augmented reality. The measurements were organised to investigate the aspects of the research questions presented before and, it would not be possible to make it without a working application. Thus the application is to consider as a model helping out dealing with and measuring the performance of the technique in general.

### 2.4.2 Study on augmented reality

Creating the application is just a huge step towards achieving the main aim of this study, which is evaluating quality of the application. The study will concern every single model applied in the application. The first result of the application quality evaluation should be from describing inaccuracy of the building's model position. The application will implement the model of an existing building and it should be displayed exactly on the same

position as the real existing one. This will allow viewing the real difference between the intended and achieved results, as shown in Figure 7.

Before creating the inquiry each building model should be thoroughly checked in order to find mark-up points from its every elevation. Since we suspect there might be different results from different positions, the points chosen as mark-up points need to be gathered from various places. Preferable position of such points is close to ground level, as they are easier to measure this way. A sample mark-up point can be a building's vertex, window's vertex or a doorframe. Since the application should view a same shaped model as the real existing building, one may be able to see the discrepancies between both in real-time. In order to get a good knowledge about the application's total accuracy it is necessary to prepare measurements from different angles but also from different distances. The application is planned for urban environments so it is considered that viewing it from further distances should not work. That is why the measurements will be limited to 100 metres away from the building position in the real environment. The plan is to observe the building from



Figure 7: Sample setting the mark-up points discrepancy.

every side and in order to do it every building will be observed from 6 different angles respectively far from one another. Also 4 observations in steps will be made from every angle of the building depending on the distance from the building. This means after choosing the right angle, the viewer will have to take observations from distance. The plan is to do them from 100 metres away and then down to 10 meters in 2 steps between, which makes 1 measurement from each: 10 metres, 40 metres, 70 metres and 100 metres away from the buildings central point. The central point is typical for every building in accordance to its coordinate point presented in the KML file containing building model features. In total this makes 24 measurements being taken for every single implemented building model (Figure 8). This quite big number should give a descriptive material about quality of the system.

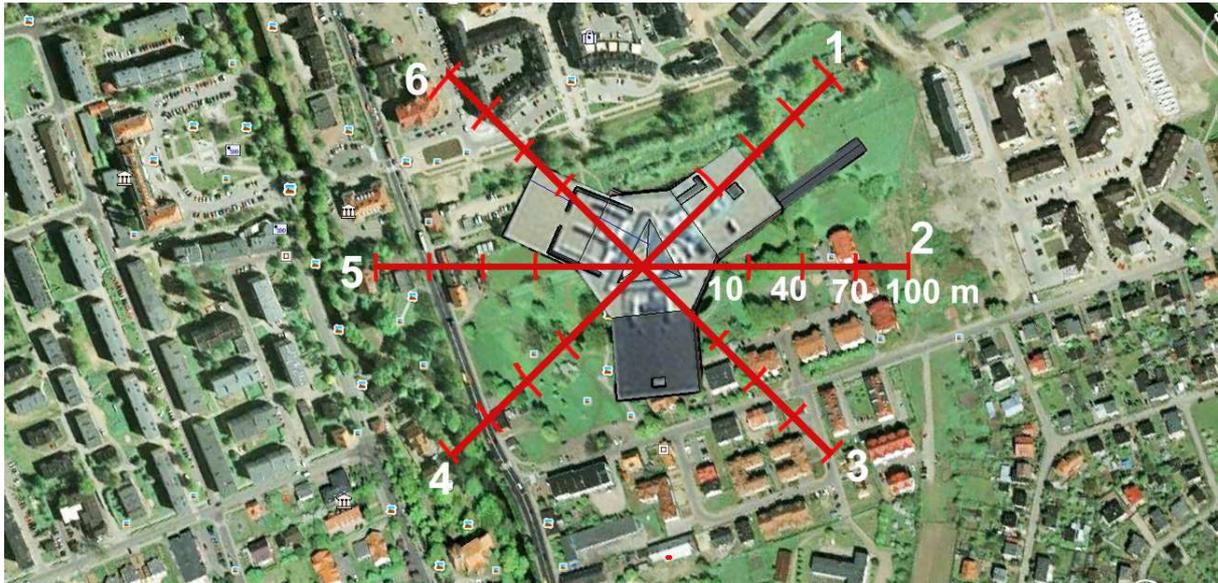


Figure 8: Grid of building observations.

The plan of the observation of building position apart from the number of measurements taken also considers a method which is planned. The general plan pre-assumes decision about the mark-up points. When chosen wisely, for all of the possible views the whole operation can take place. It will require two people, from which the first one is the mobile phone operator – the viewer and the second one is measuring the discrepancies between both model and the real building. In order to this the person responsible for measuring will have to stand exactly in the place where the mark-up point is shown in the camera view of the mobile phone, according to the camera operator indications. The camera operator, of course, needs to aim the mobile camera exactly in the direction of the real building. The next step will require using the measuring tape and marking the discrepancy in metres. This will show how big the shift is between the model and the real building. Still we need to know that from one place the shift might be different than from the other even during the same measurement. That is because the system also calculates height of the building which might vary much from the actual result and even despite the fact that mark-up point has full accuracy. That is why, apart from measuring, the discrepancy and also the distance shown on the display will have a written record. If the measurement will show 50 metres with

Table 1: Pattern for preparing study about the building:

Measurements		1	2	3	4	5	6
10 metres	Difference						
	Distance						
40 metres	Difference						
	Distance						
70 metres	Difference						
	Distance						
100 metres	Difference						
	Distance						

an actual distance 40 metres away, it will be certain that the building has much inaccuracy despite the shift shown from the mark-up point discrepancies. Still, even when the distance is shown being exactly, the same as the real one, the shift will reveal calculation precision. This shows that both of the factors are important during measurement and will have to be treated

with the same attention. The result of all the observations shall be presented in the separate table looking the same as Table 1. After filling every cell, interpretation and analysis will follow.

This will also be supplied by both camera operator and person responsible for measuring observations concerning other conditions, like for example weather during the day on which the study is being taken (weather might affect GPS), or TTFF time when the first indication from GPS comes (it might reveal that the system is working on A-GPS, much less accurate). All together each building will be commented and such a conclusion will allow bringing useful conclusions and ideas for future work.

After organising the measurements the plan is to present the application to 5 unbiased users. This number of users should be just enough present existing tendencies and intensify the study with a user view showing how the application is perceived by the potential users themselves. They will be shortly introduced with the idea of the application (i.e. the idea of augmented reality, the implemented building and the aim of the agenda) and will receive the mobile phone with the application turned on in order to answer 4 short questions. The questions (noted as Q's), about the user shall be asked are concerning the research questions of the study such as:

- Q1: How much, did in your opinion the inaccuracy spoil the effect of the augmented reality?
- Q2: How much did the effect change, viewed from different angles?
- Q3: How much did the effect change, viewed from different distances?
- Q4: How did you like the quality of the presented building model?

The possible answers are:

- 4 – Very much
- 3 - Much
- 2 - Somewhat
- 1 – Not at all

Completing such a survey compared with the previous measurements should allow supplementing the conclusions with the missing users' point of view concerning major parts of the application and developing the study. The survey should be simple and short not to bring too much data and to enable expression of the user's opinion about the augmented reality application.

The measurements bring the unbiased application's performance description. On the other hand the users' s survey allows to represent an opinion from the user's point of view on the application and thus the whole technique in general. Combination of both conducted on the working application makes it possible to answer each of the research questions. For instance RQ1 is a question concerning application general accuracy. In order to answer to this question the results of the distance difference and position discrepancy from all of the operated buildings will be gathered and interpreted. Operating those data will be done by forming the intervals between the obtained minimal and maximal results and also by the mean of all the results. Presenting this allows describing exact outcomes at worst and best case. In order to get a better perspective also the results of the user's survey concerning Q1 will be interpreted while analysing this research question. RQ2 and RQ3 concern the dependency between the work of the sub-devices and the accuracy of the application. RQ2 deals with this issue from the user's point of view and that answer will require data coming from the participants answering questions Q2 and Q3 of the survey. On the same hand RQ3 focuses moreover on unbiased data obtained with different calculations. In order to define the GPS device accuracy more precisely a short study will be organised showing what are

the indications of the mobile device accuracy of the GPS data (it is possible to get such data however its reliability is highly disputable and this require consideration). The study on GPS' accuracy has to be supplied with the analysis of the distance difference between the measured one and the one pointed by the application's calculating system. This study will be done basing on the mean distance values for every building gathered during the measuring phase. Such data can then be recognised by a percentage rate of inaccuracy allowing to determine application's worse and best performances. The last RQ4 can be only evaluated by the user as there is a limited chance of measuring models quality and the actual intent is only the users satisfaction. That is why the answer for this research question will be defined by users answers to Q4 in the user's survey. Q4 is just concerning user's opinion about the quality of the presented model thus the answers should allow drawing conclusions about RQ4.

### 3 RESEARCH METHODOLOGY

Every single scientific study needs to be systematic, so that every single action within can lead towards successful closure of the covered project. To achieve the best order and also the best clarity of the process and quality and the study one has to abide a given research methodology applied in the beginning of the project. According to definitions presented by C.R. Kothari in his “*Research methodology: methods and techniques*” there are different schemas possible for use but the most applicable within this study is choice between quantitative and qualitative research. “*Quantitative research is based on the measurement of quantity or amount. It is applicable to phenomena that can be expressed in terms of quantity. Qualitative research, on the other hand, is concerned with qualitative phenomenon, i.e., phenomena relating to or involving quality or kind.*” (Kothari, 1985). This derivation concern most of the studies depending on how the knowledge has been collected. The AR implementation on mobile phone seems to be a very complex study as researchers have to substitute all the devices responsible for setting the application in previous approaches, like the Head Mounted Displays (HMD) like the external Global Positioning System (GPS) receiver or the calculating mechanism. That is why all the work concerning the application have to be divided into sub-steps allowing to create an organised schedule so that all the work concerning the application were done on time. The whole study idea reaches AR in general but it focuses on visualising building models and measuring the accuracy of such building it has to be included in the plan of the project design apart from the ordinary elements concerning such an application, like GPS and the magnetic sensors, calculating mechanism and a proper user interface. Also the study of the application-general-accuracy needs organising, that is why the general plan of implementing the real building model and testing the correlation between the real building and its model emerged.

In case of the AR study on the mobile phone we have four different stages that are based on different features of the project whereas different methods applied. The workflow progress is presented in the Figure 9. Each stage contains different objectives, as combined study does not provide a full image of neither qualitative nor quantitative research. Since some of the parts of the study cover different perspectives of knowledge and thus they require both qualitative and quantitative approach, a mixed method of both methodologies needs to be applied where data for each part of the study will be collected sequentially (Bryman, 2006). Now every single step has to be investigated in relation to research methods applied.

#### 3.1 Formulating the problem

First step when conducting a work in the area of augmented reality is to state the real problem. After narrowing down the interest to ‘visualising buildings’ one comes close with the current state of knowledge and what might be still missing. This investigation, in this case, allowed to reveal the possibilities of mobile phones with Android Operation System, establishing the programming environment for this purpose or choosing the *GoogleWarehouse* models and find a way of their appliance in an AR application. It was revealed that accuracy issues could be a problem, not only by each device but the whole application in total.

There are various possibilities, mentioned in Figure 9, of data-collecting sources enabling forming the proper character of the research and these are previous knowledge, existing literature and other projects in the same area and current observations, based on suggestions from supervisor as sources mentioned above. Previous knowledge is coming from former attended lectures, mainly about GPS system and programming mobile devices. Observations are described during the writing of the thesis and is mostly concerned about existing problems or lack of good

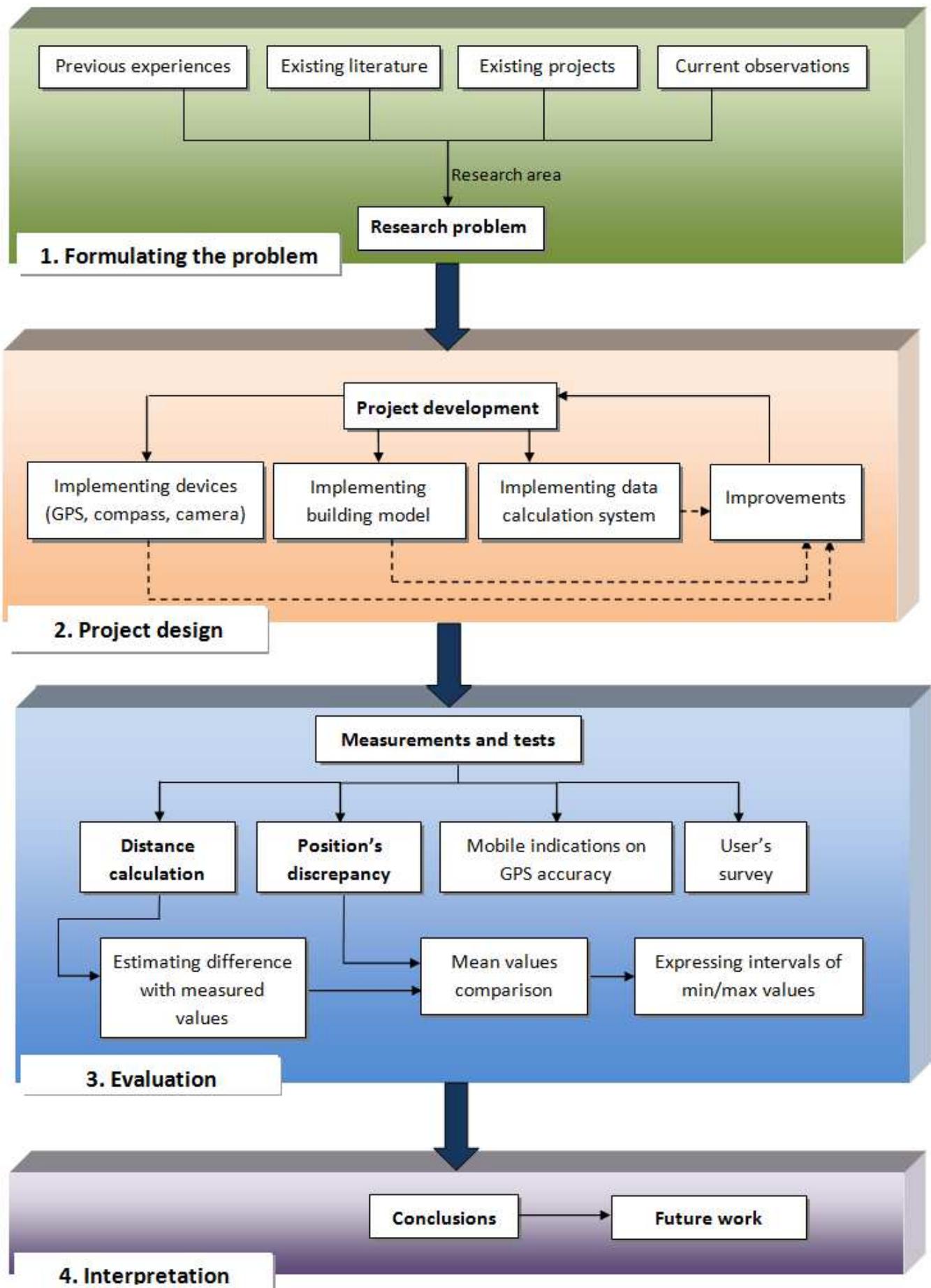


Figure 9: Division of work process during the study.

quality while programming. Gathered knowledge is containing experiences with popular existing projects like *Wikitude* or *Layar*. More complicated actions associated with the research process is happening during revealing the literature and existing projects (which were also revealed by literature references) in a process named literature review. The whole inquiry should bring vital result i.e. research questions, revealing the area of interest in the researched subject and aims for the next steps of the study. Research questions are truly important as even the best answers will not cover the subject if they are stated improperly. Questions chosen within this study were already mentioned in the section 2.2.

## 3.2 Literature review

Technically the literature review is part of formulating the problem issue, for making it clearer it will be considered separately. Forming a proper literature review has undoubted meaning as it directs point of view of the researcher, brings the knowledge and allows to correct potential mistakes. As some areas of the researched spectrum like implementing the code require practical knowledge or can be researched on technical web-portals, they were skipped during the literature search. That is why parts referring to programming or basic knowledge concerning: Java programming language or Eclipse IDE environment were simply not included in the research. For creating the research tools there were some keywords prepared concerning the different areas of the study, which required explanation and clarification. The keywords are:

- Augmented Reality
- Visualising building models
- Android
- Mobile phones
- Smartphones
- Global Positioning System
- Magnetic sensors
- Collada

There were also another choice concerning databases used in search for valuable resources. The list was based on the most popular databases designed for storing 'computer science' materials based on the list from the BTH library resources, i.e.:

- ACM Digital Library
- Google Scholar
- IEEE Xplore Articles
- ISI Web of Science (ISI)
- ScienceDirect Journals
- Springer Link
- Willey Interscience

The databases are presented in Table 2, using following acronyms: ACM – ACM Digital Library, Scholar – Google Scholar, IEEE – IEEE Xplore Articles, ISI – ISI Web of Science, SDJ – ScienceDirect Journals, SL – Springer Link, WI – Willey Interscience. Table 2 shows the number of results found when searching for presented keywords.

Table 2: Keyword data search results:

	ACM	Scholar	IEEE	ISI	Safari	SDJ	Springer	WI
Augmented Reality	14 680	397 000	3 072	369	124	311	11 684	20 857
Visualising building models	716	30 300	332	97	4	1074	1 087	41 763
Android	1 741	54 400	401	669	446	1 828	1 960	1 071
Mobile phones	29 314	425 000	9 032	11 503	1 592	26 164	26 927	19 123
Smartphones	1 940	24 500	373	363	595	816	1 786	285
Global Positioning System	10 243	299 000	17 401	13 732	810	43 746	30 935	204 301
Magnetic sensors	4 773	31 500	13 367	13 151	83	71 707	26 390	25 560
Collada	104	4 380	11	24	0	275	256	248

As a result we can analyse percentage share of the number of all found sources, presented in Table 3. Since only the first four databases have bigger share we will focus on those four: Google Scholar, ScienceDirect Journals, Springer Link and Willey Interscience.

Table 3: Percentage share of found sources:

	ACM	Scholar	IEEE	ISI	Safari	SDJ	Springer	WI
Percentage	3,11%	<b>65,12%</b>	1,85%	1,78%	0,39%	<b>7,82%</b>	<b>5,24%</b>	<b>14,69%</b>

The study in this subject was chosen based on previous knowledge coming from former education, lectures, completed studies and interests. The aim of the review is to extend and fill current knowledge on the presented subject. In order to do this, there was firstly the literature referring to keywords searched and it gave background knowledge while forming the problem or research questions. Once again each of the four databases were searched with the keywords and the 10 first sources from Google Scholar and the 5 first from other four databases were skimmed by its content. Every single source was then quickly evaluated in context of its usefulness in the subject. In total there were 200 sources checked.

The aim was to get deeper knowledge of the subject. In order to achieve it there were three areas formed by combinations of the keywords and one subject left alone (Collada). It allowed to find sources significantly relevant to the subject as now all the literature that was a result of such search was not found by accident. The numbers of sources in given areas are presented in Table 4 and eventually depending on the databases different number of sources was checked in the databases. At first the initial 5 sources were skimmed with its full content when it was not enough to draw a satisfying level of knowledge in the fields presented by keywords another group of five was chosen until receiving all the vital information. The final numbers of the first sources coming from the selected databases, skimmed during the process, are presented in the Table 5. As a result of this step of the process there were 200 sources looked up from which 15 were ultimately included in the thesis during the process of writing. The excluded documents were either redundant with presented information, not relevant to the subject or did not bring any fresh ideas that would fit this study. All the articles included to the reference list coming from the literature research are listed in the Appendix A. The other sources added in the final references are part of the previous education, the attended lectures or academic advices. All the articles included in the reference list not coming from the literature review are listed in the Appendix B. It is important to underline that the literature review was useful not only because of the final list of references but it also gave knowledge about what exists in the literature not only from this particular study's point of view.

Table 4: Study area search results:

	Google Scholar	ScienceDirect	Springer Link	Wiley Interscience
Augmented reality, Visualising building models, Mobile phones	2 730	10	20	404
Smartphones, Global Positioning System	18 600	112	180	99
Android, Magnetic sensors	1 040	23	44	29
Collada	4 380	275	256	468

Table 5: Number of first sources checked from database for each query:

	Google Scholar	ScienceDirect	Springer Link	Wiley Interscience
Number of sources checked	20	10	10	10

### 3.3 Project design

After fixing the research area and performing the literature review one can have a good view of the whole process till the end. In case of this AR study the target is clear and it was to express the AR measuring agenda. In order to conduct the survey an obvious step was to create the application. The application required work in some sense detached from the rest of the study, it was based on former knowledge and experience. This step of the study was much more focusing on the technical side. The implementation was not finished until the code reached certain functionality and quality. Every single code line was investigated hundreds of times before the whole step-sequence was finished and tests were possible. Still, this segment of the study process has an empirical research approach as it was focusing on how well the project was done in terms of worth and how it was probed by accordance to prior assumptions (Kothari, 1985). The only measurable item was the rate of fulfillment and the assumed goals of the project. However afterwards the next step came: assessment, which was meant to control work progress in much more unbiased way.

Also during this part there were decisions concerning technical methods that had to be made. This was the choice of the environment and programming sources. The choice of the mobile platform was done on beforehand and it was concerned with both previous experiences and simplicity of the chosen platform. The application was created on Android mobile phones. The environment was also quite straight forward as it is the most common choice for programmers designing applications on Android phones and it is the Eclipse IDE. The phone used for this application was *HTC Dream* and it worked under Android API 1.6 (version 4) and the SDK version used was 2.2 which was enough for dealing with Android API version.

### 3.4 Project evaluation

The major step of the study involves creation of an application-accuracy survey. The data collected using this method is certainly fitting into the Quantitative methodology as it is strict, unbiased and can be measured. Of course due to different devices applied, here

mainly GPS. The GPS agenda will not be repetitive as GPS indications might bring different results from the same position during two different measurement occasions due to different satellites positions or different weather conditions. This is a problem but the idea of the whole assessment is to try to control and handle the data and see how well the application deals with real-time operations and real-environmental tasks in general. Since there are many variable conditions repeatability is not so vital.

The evaluation plan will be carried out using 24 measurements from different, well described points, just as it was mentioned before. From every single point there will be two values picked, one describes the building-model-shift in relation to the real existing one. It shows much about the existing discrepancy between what is desired and what is actually achieved, however just this one value is not enough to describe how accurate the application is. One needs to create a whole scale and that is why also the second value is needed which is the distance discrepancy. The measurement is planned to be taken from fixed distances, i.e.: 10, 40, 70 and 100 metres away from the central point of the building and on the other hand there is a distance value calculated by the application displayed on the screen. The application has its own calculation system, as it is creating the whole new reality, based on the information stored in the building model and retrieved by the GPS and magnetic sensors. In case of any inaccuracies the distance thus gathered will differ from the real one. There is a simple, angular relation between the distance from the building and its height. For instance when the application estimates that the user is closer to the building, than s/he really is, it will display a bigger building than the real existing one, just as it is shown in Figure 10, where just a slight discrepancy of 10 metres shows that either the model is drawn bigger (if it is presented 10 metres closer) or much smaller (when it is calculated it should be viewed as 10 metres more far away). When also the shift is included one can already grasp two out from three dimensions describing the model position. Because the display is a planar translation of the three dimensional world it is the best possible unbiased piece of information describing accuracy.

Since retrieved data is measured and it serves verification of the process, all the collected measurements are having its unit and it is stored in metres. The last part of the evaluation process is grouping the results in sections of close results. One should consider differences between the values coming from the same angle. Generally the values from the same angle should look similarly, but if some of the displacements will be unnaturally bigger than the other, such a sample, should be treated as GPS fault and if all values from one angle will be inaccurate there must be a reason why wrong calculation or magnetic sensor inaccuracy occurs.

### **3.5 Interpretation and conclusions**

The final stage of the study will cover interpretation of the achieved results and it will pretty much depend on the received data during project evaluation. The received data in assessment should be analysed in order to find some regularities. This should also happen in context with prior steps, for instance slight inaccuracy might have been a reflection of algorithm choices and geographical data calculation solutions. The interpretation should result in comments that should follow with clear conclusions and what should be done in order to improve current results. Also, the whole view needs to be very critical as the AR subject is very extensive thus it is impossible to cover it all. The interpretation should be done on basis of knowing mentioned pitfalls as accuracy dependencies and variations of real-time data and the need for different performances. The final aim is to be able to present comments and conclusions on findings relating to these issues and for researchers and users developing the technique of AR into the future.

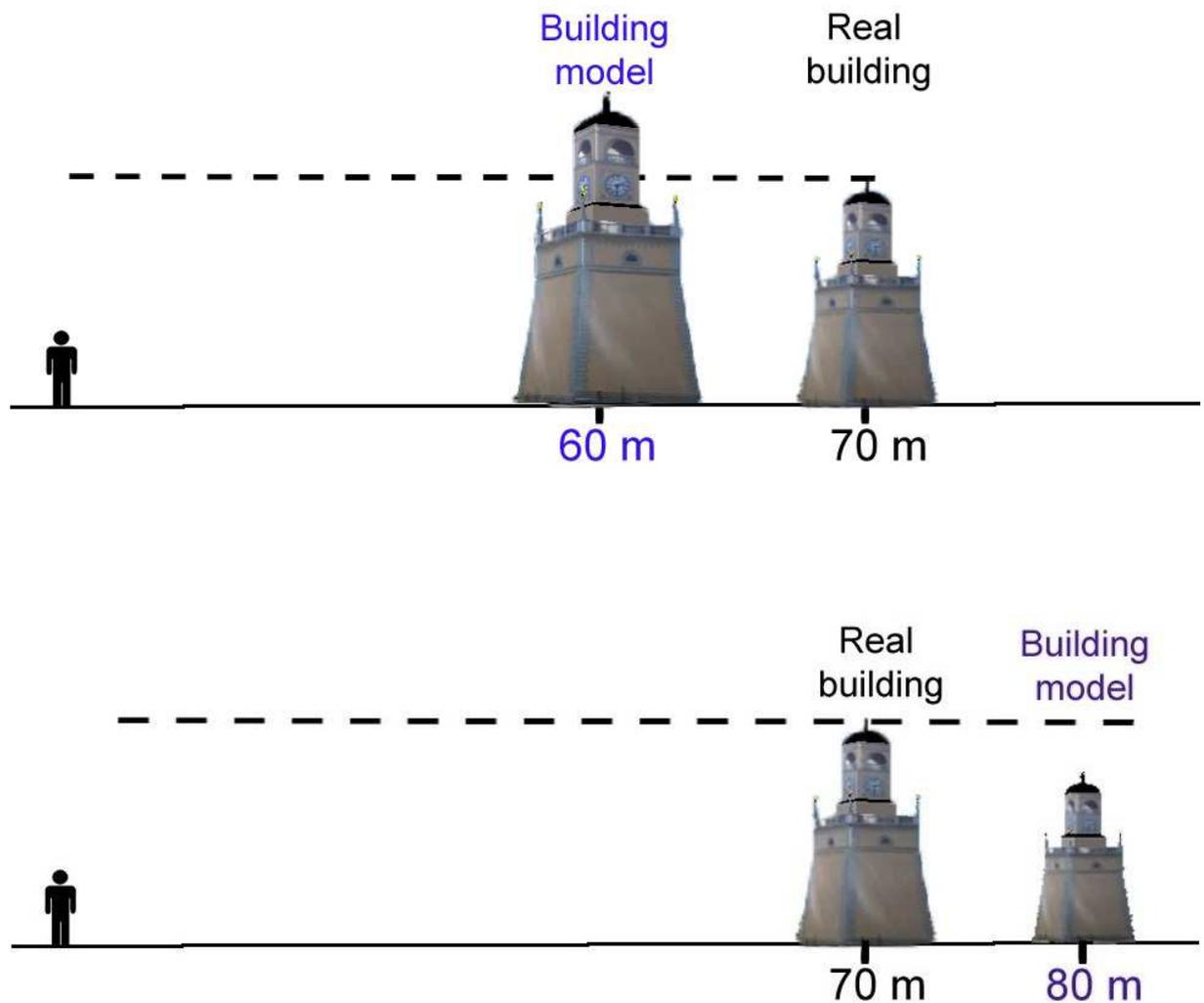


Figure 10: Relation between the real and calculated distance.

From a scientific point of view it also fits into a mixed method of researching as this is a sequence of *analytical research*. According to Kothari “*In analytical research (...) the researcher has to use facts or information already available, and analyze these to make a critical evaluation of the material*” which is exactly what is the plan for this part of the study. During the whole study it will show that there are different scientific techniques applied, mostly: qualitative and quantitative methods. The implementation part and the final propositions for the future work combine the empirical and analytical research. This is a perfect combination, as this should give a full view on the possibilities of visualising buildings applied to Augmented Reality technique only using a mobile phone which also brings the scientifically basis to the subject.

## 4 THEORETICAL WORK

After forming the research spectrum there is still some work left before the implementation. One has to consider all the aspects of the projected application before starting the work over the code. In case of the augmented reality application which is under development during this study there are a lot of cases that require explanation and clarification as the application needs to deal with different devices and operate in diverse conditions. What has to be implemented is a substitute 3D environment working on the same basis as the real environment and immediately projecting building models in the same position as they would have been visualised in the real world. For this purpose there should be a graphical library involved with the building model applied. Also, the whole interface needs to be prepared for storing, retrieving and displaying information for both the user and the calculating mechanism. All the information referring to the newly created “reality” parallel with the real existing one is written with GPS and magnetic sensor indications. What is important is that this is actually enough to get to know about current situation and reference between the real world and mobile phone.

The magnetic sensor gives the user three basic values: azimuth, pitch and roll. These values are used for calculating view rotation which is important but does not have the biggest meaning as compared to Euclidean vector which gives the user the information about sense, but what does sense mean without knowing the direction? That is why vital knowledge is gained by GPS indications.

What we receive from the GPS receiver using the mobile phone facilities are three simple values: longitude ( $\lambda$ ) and latitude ( $\phi$ ) and altitude describing the position of the mobile phone. In order to deal with the two-dimensional world one should firstly focus on the first two values and then consider the world from a cartographic point of view. Normally, both longitude and latitude are expressed in degrees, minutes and seconds. However in case of digital data coordinates the values are stored using normal floating values, which is a different version of the same, where degrees are the integer part and minutes and seconds are calculated as floating part in decimal order. The positive longitude values are describing east direction, the positive latitude is describing northern direction, whereas negative values refer to respectively western and southern directions.

On the other hand, while applying a model from *GoogleWarehouse* the position is written inside the Model tag by the same three values: longitude, latitude and altitude. The same rules apply to the way they are stored as it was with coordinates indicated by the GPS receiver. After getting to know the position of the viewer, and position of the building in the real world, the next step is to calculate how far we shall create the building from the viewer, or in other words: how big should the drawn building model be on the display? Now we need to refer to what we call ‘vertices specification’, and remember that the model-vertices-position is expressed using metres in distance according to exact positions in the real environment. So the questions are here how to combine coordinates expressed in degrees and the position in metres? A simple solution is to calculate the distance between viewer and the building into metres then put all the vertices in accordance to that distance.

The distance between two geographical coordinates can be calculated using spherical trigonometry between those points and evaluating the arc on the top of that created triangle, however during this thesis we will use the simplified version of this calculation, as we intend to create a clarified calculation mechanism and evaluate the AR technique in general. The requirement is just basic geographical knowledge and Pythagorean theorem appliance. For this method we will have to remember that due to Earth’s shape all of the meridians are having same length and parallels on the other hand are sized differently, as they are coming from the equatorial (the longest parallel) to both poles in the Figure 11 it is visualised and showing the view of the northern Earth’s hemisphere. The meridians are marked using red colour and they are looking like a circle radius. Of course it is because this view is planar, but in the spherical view, they are also sized the same as every meridian is

just arched in the same way. On the other hand parallels, marked in yellow, are just a rim of the covered surface. Since Earth is spherical shaped, coming from equatorial through tropics till the north or south pole, they are embracing less and less land. As a result, the length of every parallel, unlike meridians, varies from each other and it depends on the distance from the equatorial. That is why we need to consider longitude and latitude distances separately and bring them together using Pythagorean theorem mentioned above.

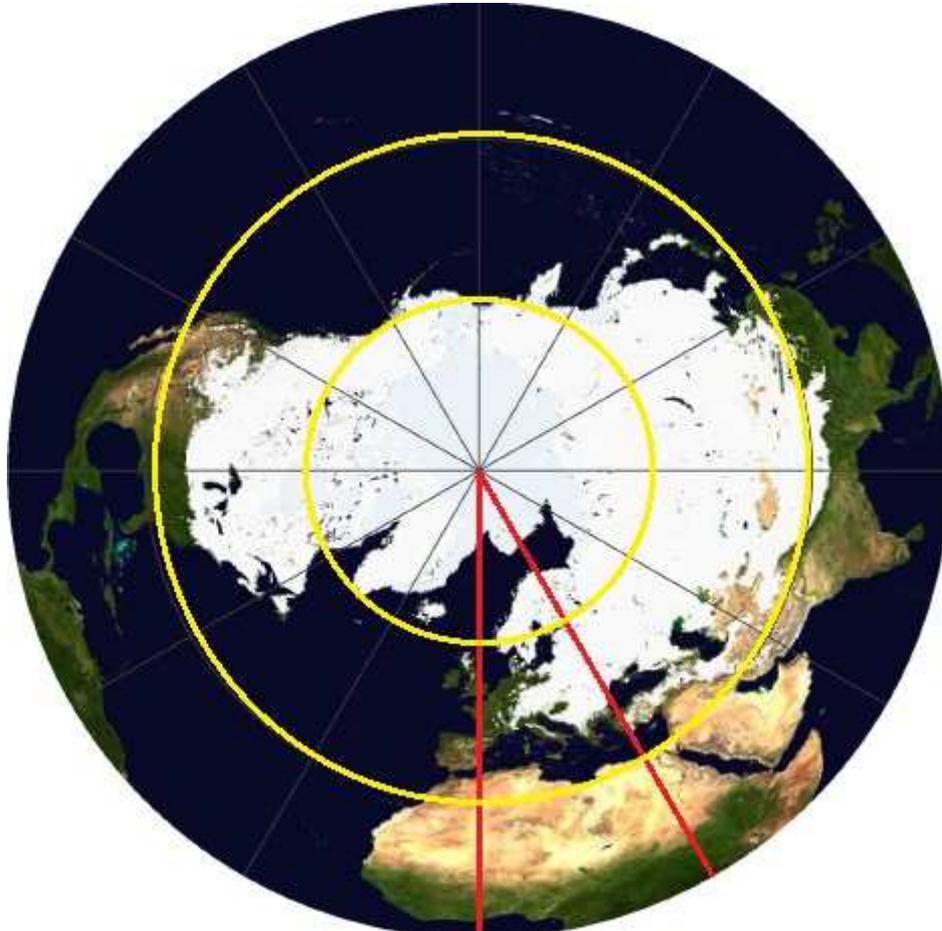


Figure 11: The shape of the Earth and consequences (photo by Simmon, 2011).

In order to do it properly one needs to treat distance separating the model and the real building as a composition of component describing the shift towards latitudinal (parallel to meridians) direction and towards longitudinal shift (along parallels). For both there are separate equations for geographical coordinates allowing to measure and recalculate the distance from being stored in degrees to being stored in metres.

Sample calculation is presented in Figure 12, with two random points A and B as distance between which we wish to measure. Point A is the viewer's position and B has coordinates of the building's position. The *length* is the shortest way between the two points and also it is the hypotenuse of the right triangle allowing to calculate the distance between the points A and B. *Longitudinal distance* is the spectrum between the points, when one only looks on the span on longitude axis and the same applies to *latitudinal distance* but this time it refers to latitude axis spectrum.

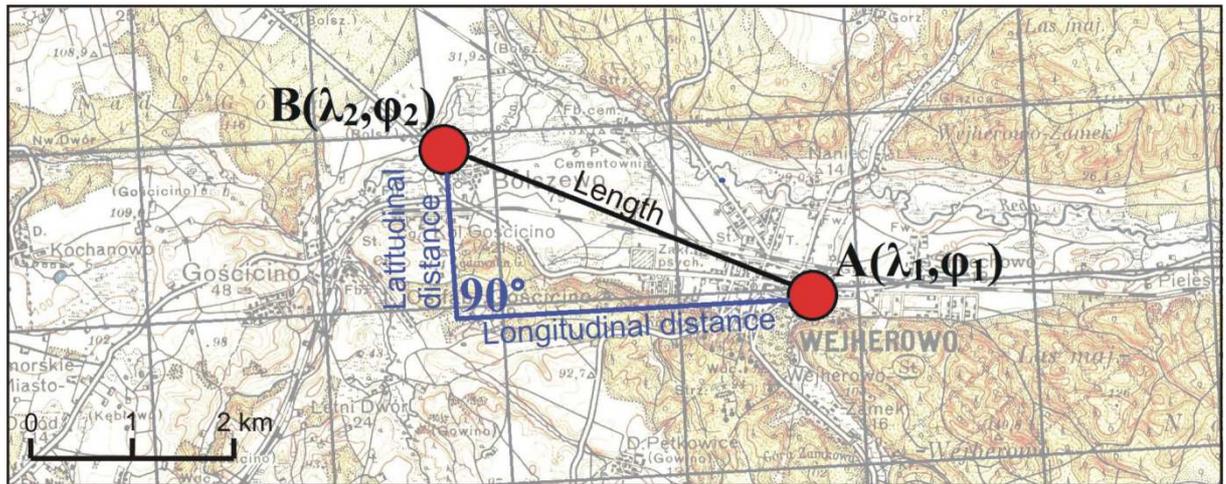


Figure 12: Distance calculation between two points with geographical coordinates.

Coordinates of the point A:

$$\lambda_1 = 18.233828544^\circ$$

$$\varphi_1 = 54.606563198^\circ$$

Coordinates of the point B:

$$\lambda_2 = 18.176429271^\circ$$

$$\varphi_2 = 54.616759527^\circ$$

Therefore *longitudinal* spectrum is:

$$\Delta \lambda = \lambda_2 - \lambda_1 = 18.176429271^\circ - 18.233828544^\circ = -0.057399273^\circ$$

and *latitudinal* spectrum:

$$\Delta \varphi = \varphi_2 - \varphi_1 = 54.616759527^\circ - 54.606563198^\circ = 0.010196329^\circ$$

All we need to do after breaking the coordinate distance into two spectrums is to measure how it transfers into metres. Because, as it is written above, parallels unlike meridians, are not the same length then each one needs to apply different solutions to deal with both lengths.

## 4.1 Calculating latitudinal distance

In order to be able to translate latitudinal spectrum from being measured in degrees into distance stored in metres one needs to have little geodetic knowledge. As it is known, Earth does not have perfect sphere shape as we rather use to call it geoid. Still for basic geodetic calculations we use a model of sphere using which we consider all the laws and rules which should also apply while referring to Earth. This inexactness should have big consequences as the discrepancies between both shapes are not so significant, especially when it comes to distances no longer than 100 metres (which is an estimation of our application range) where those discrepancies are values of a single millimetre. In that case we assume a constant value of the great circle. The Great circle is a rim dividing Earth into two same hemispheres. A possible Great circle is an equatorial or two opposite meridians (like meridian  $0^\circ$  and  $180^\circ$ ), but not every parallel as they have different lengths as it was explained above. The first fact is that the Great circle has size of 40 000 kilometres (De Canck, 2007). Since a Great circle consists of two meridians and every meridian is equal, we can assume that every meridian is 20 000 kilometres long (of course treating the Earth as a

model of a sphere, as actually it is slightly flattened at some points). So now it is important to realise what we want to calculate. We know how long the meridian is. We wish to know how long one degree of the meridian is when speaking in metres. Then we would be able to answer what distance our spectrum does bring. In order to do this we need to get to know how many degrees there are in one meridian. Since there are 90 degrees for the northern hemisphere and also 90 degrees also for the southern hemisphere and in total it is 180 degrees that make up for one full meridian. Now, the calculation is simple. If one meridian is 20 000 kilometres long and one meridian has 180°, then one degree is a result of the following division:

$$l = \frac{20\,000 \text{ km}}{180^\circ} = 111.111 \frac{\text{km}}{1^\circ}$$

This means that a single degree in latitudinal length consists of 111,1111 km. Coming back to our calculations it means that latitudinal distance should be multiplied by value of one degree:

$$\Delta \varphi = 0.010196329^\circ \cdot 111.11111 \frac{\text{km}}{1^\circ} = 1.13292543311519 \text{ km} \approx 1.13293 \text{ km} =$$

$$\Delta \varphi = 1\,132.93 \text{ m}$$

which allows us to estimate the value of latitudinal scope for more than 1 kilometre 132 metres 93 centimetres (Pelczar, Szeliga, Ziółkowski, 1978).

## 4.2 Calculating longitudinal distance

To calculate the longitudinal distance, gladly there are already equations existing. Still, we need to observe that since the lengths of parallels are dropping down while going away from the equatorial, the length of one degree value is differing depending on how far we are from the equatorial. The angular distance from the equatorial is one of the definitions of latitude, that is why basically length of one degree in longitudinal distance is basically a function of latitude. In other words, 1 degree means a huge distance on the equatorial but also a tiny one close to the any of the Earth poles. It is important to keep in mind that one cannot forget about latitude while calculating longitudinal value of one degree. The function that calculates the value of one degree on parallels is already described in geodetic tables so one can easily check it up on relevant tables and it is in the following function:

$$l = \frac{2\pi R \cdot \cos(\varphi)}{360^\circ} \text{ (Pelczar, Szeliga, Ziółkowski, 1978)}$$

where R is Earth radius:

$$R = 6378.245 \text{ km (Pelczar, Szeliga, Ziółkowski, 1978).}$$

So, returning to the example  $\varphi$  is a value of  $\varphi_1$ , as it is the starting point, and as a result we achieve:

$$l = 64.47598228 \frac{\text{km}}{1^\circ}$$

This means that a value of one degree on the latitude 54,606563198 is close to 64,5 kilometres. Of course in order to make a proper calculation we need to give a radian argument for the *cosine* function, which in Java environment means multiplying the latitude value by  $\pi/180^\circ$ . Now, we should remember that the longitudinal range measured in degrees is:

$$\Delta \lambda = -0.057399273^\circ$$

so the same as above we need to multiply it by the value of one degree:

$$\Delta \lambda = -0.057399273^\circ \cdot 64.47598228 \frac{\text{km}}{1^\circ} = -3.700874509 \text{ km} = -3700.87 \text{ m.}$$

A negative value does only give us information about the direction (point A is much more to the East than B point) still the value does normally describe the length and it is 3 kilometres 700 metres 87 centimetres.

### 4.3 Bringing latitudinal and longitudinal distance together

Since we already have the information about latitudinal and longitudinal distances in metre we have everything it takes to measure the whole distance. Bringing it back to right triangle we have in Figure 12 we have the length of both sides adjacent to the right angle and we can calculate the hypotenuse length using the Pythagorean theorem.

$$\begin{aligned}\Delta \varphi &= 1.13293 \text{ km} \\ \Delta \lambda &= -3.70087 \text{ km}\end{aligned}$$

From the theorem we know that:

$$\text{length} = \sqrt{\Delta \varphi^2 + \Delta \lambda^2}$$

and we can calculate and receive a value of:

$$\text{length} = 3.870399488 \text{ km} \approx 3 \text{ 870.40 m}$$

This is our final result, the distance from the point A to point B is equal 3 kilometres 870 metres and 40 centimetres.

## 5 APPLICATION DESCRIPTION

After clarifying the aim of the study, forming the research area, getting to know all the existing obstacles and possible risks based on the literature review the next step was to start working on the application. The implementation is the breakthrough since that is done we can finally observe how the previous knowledge benefits, this is the moment where the theory finally is implemented and can work out in practice.

The implementation is very complex as it requires organising a lot of prerequisites as well as it is time consuming. The work effort of the application in augmented reality took in total six weeks, as every single step needed to be verified. It started with creating the interface which was enriched and based on current observations and new knowledge applied. The next moment was capturing the view from a standard camera of the mobile device and setting the camera's orientation. After this the aim of filling the interface with data gathered by GPS receivers and magnetic sensors followed. This allowed the user to understand better what was happening with the application during every single moment. The biggest challenge in the implementation of the AR application was dealing with the building model and displaying it on the device screen. The last step was the right positioning which involved organising a whole calculation system dealing with current position data and viewing the model in the right position responding to values given by the GPS receiver and the compass sensor.

The basic idea for the application is that it is a prototype project containing just the crucial functionality of an augmented reality application visualising models of virtual buildings. After the study the agenda about application accuracy will be created and the results of such an agenda will allow to tell what the possibilities are for application extensions and how well the application does act. The intention is to create a complete functional application allowing to browse the environment with a 3-dimensional view of the terrain with possible functionality for real estate agents or building developers allowing them to present their architectural plans.

As mentioned in the study the application was meant to be designed for phones equipped with Android Operation Systems. That is why the work is narrowed to facilities best for these group devices. The most simple choice for programmers developing Android applications is using Eclipse IDE environment, as it is even recommended on "Android Developers website" which is an Open Handset Alliance (OHA is the official AOS owner) product designed for programmers. There are of course other possible ways but mostly they require manual installation of the ADB directory managed from the command line which is not so convenient as while using all the plug-ins organised by the IDE (Integrated development environment). Still, using Eclipse IDE does not mean that everything will be done by itself though it is also concerned with several obligations, firstly installing the Android Software Development Kit (SDK). It is an open source product, available on several sources and installing it is necessary as it contains all the vital tools needed by the programmer developing Android Projects. Then comes the obligation of installing the platform. Android requires at least one platform in SDK to be installed, so if there is yet no platform installed one has to do it right away. To do this, one has to use the *Android SDK and AVD Manager* option and prepare the necessary repository site. The last task before starting the implementation is either preparing the mobile device or using one of the possible Android Virtual Devices (AVD). AVD allows the user to test and debug the application without actually having the mobile device connected to the PC. In some cases it is very useful, however for using the external devices, which is actually a huge part of the AR application intention, it is not so applicable. For instance there is yet no solution presented for implementation using the Android phone's camera. So when the user tries to use the functions based on the Camera view s/he sees just a random image of a chequered area and a square moving (Figure 13). The same concerns also using the GPS receiver or the sensors, which of course is possible but is not functional since there is no real phone and thus the

sensors are not working. Of course there are tools dealing with this like sending the emulator GPX or KML file with trace with GPS coordinates, however in total it makes it much easier to implement and check the application using the real mobile device.

The application will be implemented on the HTC Dream Android phone. It is a standard smartphone with just enough functionality for a sample AR application. The execution of the application on the mobile phone is not done automatically after plugging it into the PC, as it requires additionally installing and turning on the PDA.net application. After doing this one may install the just implemented code on the mobile device and enjoy full mobile phone functionality, which in our case means the possibility of using and testing the camera and the sensors.



Figure 13: View from camera implemented on AVD.

Finally after dealing with all the prerequisites mentioned before, it is possible to create the application. Starting creating the interface, was to settle the layout. There are different possible types of layout available for use while forming the application on the Android phones and there are also two possibilities of implementing them in the application. One may use the Activity class or using the automatically prepared XML file for layouts which later on will be parsed and form the R class storing the information about the application layout. The interface needs to contain elements concerned with information gathered by the application which is the shape of the building model, the view from the camera, the GPS sensor and the magnetic sensor indications, the calculation of the distance between viewer and the building. The final choice is to put all the elements into the *Frame Layout* which is a simplest layout allowing the elements inside being distributed all over the screen. Both of the displaying elements as the building model and the camera view were put inside the separate *Relative Layouts* which was a good choice for the elements that might occupy the full screen. The *TextViews* which is a solution for storing the text information on the interface were put in the *Linear Layout* moved to the bottom of the screen which was for uncovering the view.

Creating the interface is basically a start towards programming the features of the application as the camera view in the background and implementation of the building model from KML. Implementing the camera view requires using the *Camera* package from hardware packages and addressing the view class reading its context, in that case it is done by creating a class extending *SurfaceView* class from android view packages. Of course the plan for the interface also assumes setting proper orientation which has to be done by changing *AndroidManifest* the standard xml file containing all the vital information necessary to run the application. The camera also requires setting the permission for the application to use its

device in the *AndroidManifest* file. Sending information from the sensors to the application is concerned with using the standard *LocationListener()* class and *SensorEventListener()*, the first one for using the GPS receiver, second one for using a magnetic sensor.

Working with sensors is still much easier, than picking the building model and then implementing it into the application. One has to know that since the model is drawn into the application one has to choose the graphical environment. The graphical library supported by Android phones is OpenGL ES 2.0 which is a version of this popular library designed for mobile programming, and it is used in our application. This means that the whole code structure is based on a library standard. The model has to be operated externally and firstly chosen from *GoogleWarehouse* storage. The standard solution for saving files using this service is KML. However for our application we need to read the Collada model file (having .dae format) as it is the file containing all the information needed by the application as the basic rules for storing the information in both Collada model and Open GL library are the same. A Collada file, above all, is listing information about used vertices, normal and texture coordinates which are also applied in OpenGL but using different schemas. In the graphical library there are arrays that need to receive information from previously arranged buffers. There are different types of arrays, from which we need to enable and organise these three:

- Vertex array (GL\_VERTEX\_ARRAY)
- Normal array (GL\_NORMAL\_ARRAY)
- Texture coordinates (GL\_TEXTURE\_COORD\_ARRAY)

All the information which we need to operate is saved in the Collada model file which describes our model well. The biggest difference is the way they are stored and that is why they need to be picked up from one file, redefined according to Java language from the file based on an XML pattern (for instance not containing comas between all the vales). This allows to display the building, however afterwards it needs to be redefined according to user position and direction depending on what the phone is pointed at. Of course the Vertex array describes position of each vertex, the normal array describes normal vector for every triangle drawn and texture coordinates array posses information about the texture piece that needs to be connected with currently drawn vertex.

The last task is done by calculating the system in function *checkVisibility()* containing all the basic information about the model default position, the mobile axes tilt and user's current position.

We need to keep in mind that vertices are described in relation to one reference point in the central part of the building. Vertex information is just stating how far away, in metres, the given vertex is from the reference point described inside the KML file. That is why vertices need to be picked and brought into the application thoroughly. It is done by calculation using a method described in Chapter 4 converting the distance between the user's and the building's position into metres. Since the vertices are picked from the Collada file and they are already giving information in metres, all one has to do is translate the building position on the screen by using OpenGL function named *glTranslate*. It contains 3 arguments showing how to translate the position using three axes. In our *landscape* orientation X is the longitudinal distance, Y is the altitudinal and Z is latitudinal distance. Also the information from the magnetic sensors need to be used. They are addressed using a *glRotatef()* function coming from the OpenGL ES 2.0 standard. Each of the magnetic sensors indications (azimuth, pitch, angle) are stored separately and *glRotatef* also allows rotations on three axes. In that case the pitch means rotation on X axis, azimuth on Y and roll means rotation on Z axis.

## 6 TECHNICAL ISSUES AND SOLUTIONS

The implementation needs to be a systematic process. In fact it is a sequence of clarified operations based on already explained features. However while dealing with more details one receives a number of still unresolved issues. A major one is that model files coming from *GoogleWarehouse* are hard to import to other environments. Some attempts towards this were done, however the *Collada* file format is yet not organised for reworking and there is lack of applicable tools. Also some tools like *ColladaDOM* application for parsing the .dae files is no longer supported and thus it is least trustworthy. The final result of this is that several issues were found and they require deeper explanation.

### 6.1 Collada (transition format)

The name Collada means COLLABorative Design Activity and is a file format storing information of 3-dimensional building models in a textfile. Collada files are stored in a file based on XML format and it is a popular scripting language used in many applications creating 3D files, such as *Google SketchUp* which is an application creating models used by *GoogleWarehouse* (De Paor, Whitmeyer, 2010). From the whole structure of models available on *GoogleWarehouse* stock, the Collada model is the most useful one as it allows for easy transition between various graphical tools. It is also applicable on OpenGL library, however there is yet no tool created that would automatically translate a Collada file into a set of code line appropriate in OpenGL environment. The task for a developer is to manually interpret the code from the Collada file and either create a script or pick the vital information and install them with an application source used for used graphical library (in our case OpenGL ES 2.0). Unfortunately this cannot be done without going into details of the Collada model, as it requires having much knowledge of the .dae file structure and its schema. Collada files by standard provide information written in different libraries, where each is responsible for different features of the building model:

- Asset (asset)
- Animations (library\_animations)
- Lights (library\_lights)
- Images (library\_images)
- Materials (library\_materials)
- Effects (library\_effects)
- Geometries (library\_geometries)
- Controllers (library\_controllers)
- Visual scenes (library\_visual\_scenes)
- Scene (scene)

From all the libraries for transferring the model into OpenGL environment, we just need to read the information written in a few of them. First of all one needs to concentrate on the unit describing distances vertices position. The information about the unit is placed inside the *asset* section. The *image library* content is connecting *material* and textures location, whereas we receive more information about *materials* essential for application in the *geometries library*. It is the place where all the meshes are being described by *position*, *normal* and *geometry uv* arrays are listed. The *position* array is responsible for describing model vertices and *geometry uv* arrays are storing information about texture coordinates. The information finalising the content of the mesh is the *triangles* tag which is describing the

triangle's vertices position. In general the same structure works for OpenGL environment, but the details however are quite different.

Each array is described using a standard way. Firstly there is a *float\_array* with a *count* marker expressing how many values there are inside. There is also an other way in counting number of values inside the *float\_array* and it is multiplying the *count* marker by *stride* value as the content of the accessor tag describes what is the actual format of information stored inside each array, as it is shown below, by sample content of *position array*:

```
<source id="mesh1-geometry-position">
<float_array id="mesh1-geometry-position-array" count="168"> (...)
</float_array>
<technique_common>
<accessor count="56" source="#mesh1-geometry-position-array"
stride="3">
    <param name="X" type="float"/>
    <param name="Y" type="float"/>
    <param name="Z" type="float"/>
</accessor>
</technique_common>
</source>
```

In that case there were 56 vertices containing values for each of the three axes (x,y,z) which in total makes 168 values inside the whole array. In analogy the *normal array* also consists of three values as it needs to describe positions in relation with 3-dimensions. On the other hand the *geometry uv* array consists of only two values in each stride. *The geometry uv* works like the *Texture\_2D* array in the OpenGL environment and this means it only contains two referencing values in each stride and it is either named UV or ST. This is because it refers to planar surface of a texture and imposes this piece of the flat texture on the 3D corresponding vertex. Up to this moment everything seems to be a parallel between both environments: the one we pick model from and the one we want to transfer it to. The vital difference reveals when we realise how the elements are referenced when the vertices are picked.

The drawing part in the Collada model is done in the *triangles* tag inside the *geometries library*, like in this sample code from existing Collada file:

```
<triangles material="material_4_13_0" count="2">
<input semantic="VERTEX" source="#mesh4-geometry-vertex" offset="0"/>
<input semantic="NORMAL" source="#mesh4-geometry-normal" offset="1"/>
<input semantic="TEXCOORD" source="#mesh4-geometry-uv" offset="2" set="0"/>
<p>12 6 21 9 6 22 5 6 23 9 6 22 12 6 21 6 6 27 </p>
</triangles>
```

This requires information about input data because they are included while giving indexes inside the *p* tag. The Collada model while parsing the content of the *p* tag first considers the number of input values. Since there are three input values, the vertices, the normals and the texcoords it groups the whole chain inside into groups of three and then goes inside each of them. In other words the first value refers to *position array*, second one comes to the *normal array*, the third to *geometry uv array*. For instance analysing this sample above the following indexes will be assigned to each array:

Position array: 12, 9, 5, 9, 12, 6.  
Normal array: 6, 6, 6, 6, 6, 6.  
Geometry UV array: 21, 22, 23, 22, 21, 27.

The OpenGL ES, which is a library for mobile devices, has a limitation for drawing triangles only whereas in standard version of this library quads and other meshes are possible. Luckily most of the files coming from *GoogleWarehouse* are designed to build all the shapes with triangles that is why we can consider three values from every array. The meaning of the input coming from the *p* tag distribution is that the first triangle consists of: vertices like: {12, 9, 5}, normals: {6, 6, 6}, and texture coordinates as: {21, 22, 23}. All the three indexes referring to normal array are the same because every drawn mesh needs to have only one normal vector. Referring to our subject the normal index for the first triangle is 6 and since the *normal array* consists of 3 values in order to attach the correct normal vector one needs to pick the sixth group of three inside this array. Same rules are applied while reading vertices and texture coordinates, however there are three indexes for every vertex and two indexes for every texture coordinate (there are also 3 values possible, however the most common amount is 2 and this is how it worked out in the sample file discussed here). What it means is that there can be different normal- and texture coordinates assigned to one vertex, whereas while using OpenGL every normal and texture coordinates has to be in the same position as the vertex is on its list. When one wishes to use one vertex twice s/he needs to write the vertex array twice with responding normal and texture coordinate values. This sample table would have to look the same for every element. For instance when the Geometry UV array would have to look the same like the Position array:

Position array: 12, 9, 5, 9, 12, 6.  
Geometry UV array: 12, 9, 5, 9, 12, 6.

Basically, while translating the Collada model to OpenGL ES structure one has to think of an overwriting solution. It is not possible to overwrite the Geometry UV array as every position there is occupied with important information that cannot be lost. The best solution is completely repositioning for every texture coordinate according to position of a responding vertex and if there is a multiplied vertex that is referencing different texture coordinates it has to be added on the end of the vertices list. This can be done manually (which is very complicated), one can create a script doing this, but also use some of the tools available all over the Internet, like *Collada refinery* introduced by official Collada developer group (The Khronos Group, 2007).

## 6.2 Textures collision (problem in transferring building textures)

As it has been mentioned before, the relation between a given texture image and a specified material is placed in the *image library* and it binds the material with the given image, so later on while tracing the texture one has to search for that material. Of course the images need to be imported separately into the application and the easiest way is to store them in the *drawable resource* storage. Then creating a bitmap type variable which will keep the address of the image (since it is in the *resource* directory it will be part of R.java class). If a proper interpretation of a Collada file is done then the building will be drawn within a proper distance, using correct normals and right texture coordinates, which means that the texture image should apply adequately. Still, on many mobile devices the model has recognisable shape but its surface is blank. This is because of a common bug, because of which on several devices the textures, which length and width are not results of exponentiation of 2, they do not display. Sadly, it concerns most of the textures, since they have irregular rectangle sizes and only few meets this requirement. Unfortunately this bug also works on textures applied on a HTC Dream mobile phone. As a consequence of the fact

that it was the phone on which this study has been performed, it was also another obstacle to overcome.

In order to manage with this issue, one has to take a close look on the structure of the UV mapping of textures. The UV sometimes also called as the ST system is a convention for bringing planar images into the world of 3D models. In our case we prepare a photography of every single part of the building's elevation. Then we start considering every single mesh as a wall, a planar image. That is when the 3D world becomes a digital 2D standard again. The final part is picking every vertex of the mesh and finding a point on texture image exactly corresponding to the one on the model. The moment when the three-dimensional vertex corresponds to the two-dimensional point inside the texture is called the UV mapping. It is important to observe how the point looks inside every single *uv array*. A sample trial is listed below:

```
<float_array id="mesh1-geometry-uv-array" count="110">1.000000  
0.000000 0.614327 0.957395 0.614327 0.000000 .... </float array>
```

Furthermore it looks quite similar, even though there should be 104 more values. As we can see the values inside such an array are in the range between 0 and 1. This is because the texture coordinates are addressed regardless of the file size, at this moment the parser does not have to know what is the exact pixel of the picture as it is described by a percentage value. It means that 0.0 is the pixel 0, value 1.0 means maximum size (u is for width, v means maximum height), 0.5 means half the size (width or length depending if it is u or v). Basically it is enough to understand what can be done in order to apply right texture to the model. First of all we need to check if the image really does not have the length or width that is a result of the exponentiation of 2. If not then we need to resize so it will happen. Since we do not want to change the ratio between length and size and actually from a developer's point of view it is usually better not to change graphical quality of the image, the best solution is to just change the size to the closest power of 2 and leave the thus-obtained area blank. What we can do then is to manipulate the data inside the *uv array* by narrowing the scope starting from 0 to 1, to starting from 0 to value lower than 1, according to image resizing ration. We can take a sample of a texture really existing in one of the models. It has a size of 173x200 which means it has 173 pixels of width and 200 pixels of height. None of these two values is a result of exponentiation of 2. The nearest bigger result of exponentiation of 2 is 256 that is why we will resize the image from 173x200 till 256x256 by adding an empty area around. Whereas pixel 173 was the wide edge of the primary texture now it is a place somewhere in the middle of the newly created texture. That is why if we wish to reach the previous edge of the texture we need to consider that now it is the 173th pixel out of 256th existing. Ultimately, in order to find it by UV mapping one has to look for a value equal to  $173/256$  and i.e.: 0.67578. Since it is very confusing and every texture has to be interpreted individually, resized and then texture coordinates have to be reconfigured it is also possible to automatically stretch the picture. For instance while referring to the texture having size 173x200 one can spoil the proportion by stretching the texture into the new size 256x256 and leaving the old texture coordinates, because UV mapping is only refereeing to proportional position of the pixel in every dimension and this does not change by just stretching the image. Unfortunately this might also mean some image defects as generally at some point of the approximation final result might bring some inaccuracy and thus the texture might get distorted that is why in general the first solution is more secure.

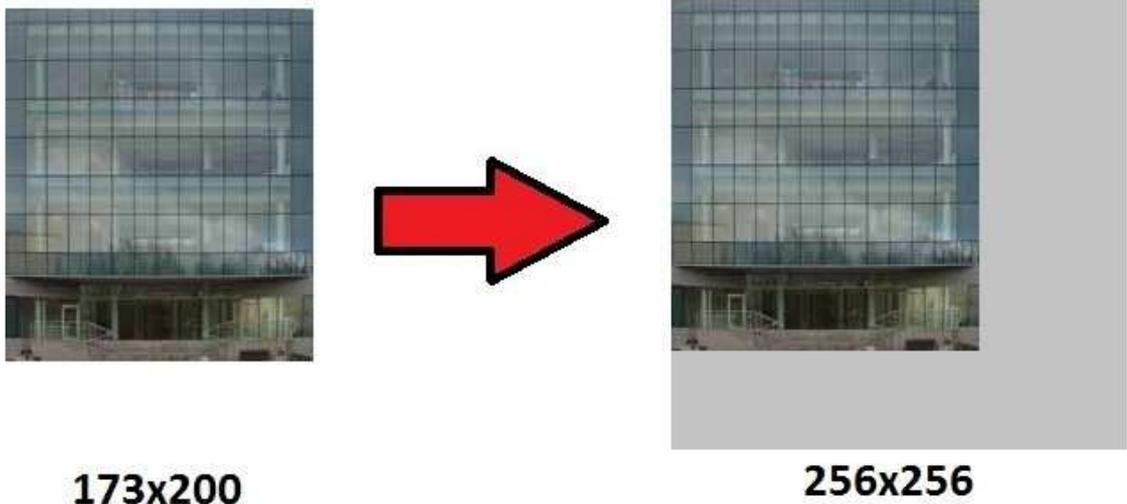


Figure 14: Sample file size changing for avoiding the texture bug.

After these changes, position in the texture image which has previously been a (1,0) or (1,1) pixel is no longer being placed on the maximal edge of the texture. As now the coordinate point (1,1) does not mean a (173,200) point in the image it rather says (256,256). That is why all the points need to be changed, just as in the Figure 15.

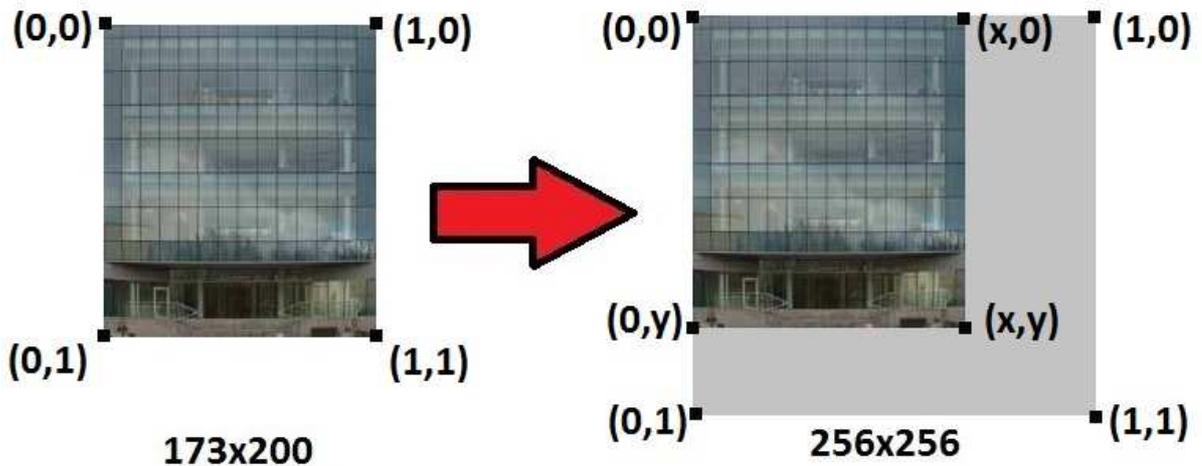


Figure 15: Basic points transition.

The calculation of a new range of coordinate points is simple as all the new values in the *uv* array will have to be divided by a value describing how the image got enlarged. In that case the width increased from 173 to 256 so it increased by  $256/173 = 1.4798$ . The height increased from 200 till 256, which means that the texture spectrum should now shrink by the ratio of  $256/200 = 1.28$ . What it means is that all the first values (width) from every couple in the *uv* array should be divided by 1.4798 and every second value should be divided by 1.28. The final result of this process is that the texture coordinates point having a value (1,1) should now become (0.6758 , 0.78125x) and all the texture points should not reach further than this limitation, from this point the gray area starts which we do not want to see inside our model, see Figure 14 and Figure 15.

### 6.3 Sensors sensitivity and azimuth shifting

The sensors sensitivity was listed as one of the major risks before creating the application. Unfortunately this risk has not been addressed as these issues are also related to the mobile device type and model which cannot be predicted and opposed by the application developer. The sensors sensitivity does frequently affect the model resulting in its floating on the screen, still it was not the major issue while programming the AR application. It was rather connected with the screen orientation. For Android programmers and available screen orientation solutions, the *landscape orientation* has been implemented here. This result in a horizontal position of the device and its display, it becomes wider then it is tall.

Unfortunately this does not work without any further consequences as it affects sensor indications. It is because all the sensors positions are defined for *portrait* orientation, which is not applicable in the application because of the building model feature, as they are mostly wider than taller. Luckily the knowledge of this fact allows turning the functionality back again. This has to be done with some mathematical calculations. For instance the *azimuth* value which has a range from 0° till 360° needs to be increased by 90° because of its destination working in *portrait orientation*, of course this means that values in the segment between 270° and 360° will be lowered into the spectrum between 0° and 90°.

Some other translations were also done to the *pitch*:

$$\text{pitch} = \begin{cases} 180^\circ - \text{pitch}, & \text{if } \text{pitch} > 90^\circ \\ \text{pitch}, & \text{if } \text{pitch} \leq 90^\circ \text{ and } \text{pitch} \geq -90^\circ \\ -180^\circ - \text{pitch} & \text{if } \text{pitch} < -90^\circ \end{cases}$$

and the *roll* values:

$$\text{roll} = \begin{cases} 90^\circ - \text{roll}, & \text{if } \text{roll} > 90^\circ \\ \text{roll} - 90^\circ, & \text{if } \text{roll} \leq 90^\circ \text{ and } \text{roll} \geq -90^\circ \\ 90^\circ - \text{roll} & \text{if } \text{roll} < -90^\circ \end{cases}$$

in order to react to phone sensors rotation. However, it still might be the case that due to sensor sensitivity the result does not fall into the right angle section and building position changes rapidly.

### 6.4 Determining viewer altitude

The GPS receivers support the user with precise values of its current position. When the user is in a good range of satellites it is possible to receive information about longitude, latitude and altitude. Determining longitude and latitude has already been pretty well described in this study. Reading information about the viewer's current altitude would allow us to make other rotations of the building according to user's point of view and for instance revealing the roof of the building when the user moves upstairs with the application. Sadly, during early stages of work it was revealed that altitude values received by the GPS did not show the real altitude position as results varied with up to 60 metres. This is because the GPS receiver works in accordance to WGS-84 geodetic map system. It assumes the ellipsoidal model of Earth, whereas it is just a simplicity that does not take into the consideration terrain denivelations and that Earth is not an ideal ellipsoid. In practice this might differ the measurements significantly and that is why obtained results were varying so much from the desired ones. Unfortunately, dealing with the WGS model would mean a lot of calculations mostly not related with the general subject of this study. That is why the

information coming from the GPS receiver will be displayed on the device screen but it will not be used in further system calculations because of its big discrepancy with the real value. The solution for calculating the user's position will be to treat the altitude by default. Because of the GPS restriction the user altitude will by default be marked at ground level which means that the mobile position will be marked as 1.65 metre height as an estimation of average eye-height of a user whom holds the mobile phone device. Practically this means that the application is restricted for terrain use and the calculations will only depend on user's longitude and latitude, the altitude will not influence the view even if the user will move upstairs.

## 7 EMPIRICAL TESTS

After creating the application the indispensable part is to test how the application acts in the outdoor environment. The more buildings there are tested the better we can describe application's general accuracy. Unfortunately the implementation of every single building model, as it is proved on beforehand, is very time consuming because every single building feature needs to be implemented individually, all the vertices and texture coordinates need to be picked manually also the texture coordinates have to be reconfigured because of the phone's texture size requirement. All this requires a lot of thorough action. This is why only a limited number of models will be evaluated within this study. The work with the application started with one small sample building which was a model being close to BTH campus. From all models available on *GoogleWarehouse* the block of flats on Snapphanevägen 8 in Karlskrona was chosen. This model served for all the initial steps of the application implementation, including drawing the shape of the building, adding the textures and the right positioning. Correct implementation proved that applying *Collada* models into the Android environment (OpenGL mobile graphical library) is possible and can be tested furthermore for realising augmented reality goals. The block of flats on Snapphanevägen, was the first such applied model and it was only created for training and establishing all the necessary functions of the application. The final result of the 'model view' can be seen from different views in Figure 16 and Figure 17. The building displayed in both figures is viewed regardless of GPS indications, just according to azimuth, pitch and roll value. It is a sort of testing mode allowing to display the building during its implementation.



Figure 16: View of the application building model: south elevation Snapphanevägen, Karlskrona.

After implementing the first building model it was to realise the aim of the study, which is to evaluate the quality of the application by performing tests in the outdoor environment. In order to do this, three single models were applied into the application. Each one of them was then visited and the measurements were taken directly from the place in which the building was placed in the real world. The plan was to take measurements from 24 positions for every single building, four times from every angle from different fixed distances from the central point of the building: 10, 40, 70 and 100 metres. The reason for

this is that in urban environments one should not get a better view than 100 metres away from a given building so the measurements were narrowed to 100 metres and then in steps 30 metres down to the centre of the building. The choice of 30 metres as it is an optimal distance giving representative results by not too many measurements. There should 6 different angles from which the measurement is being taken, taking more angles into consideration would become hard to operate and still organising measurements from 6 different angles should in most of the cases at least once give a good view on each of the building elevations. For every single position out of 24 planned for the building there should be two measurements noted, i.e.: discrepancy between one mark-up point in the building model and its corresponding point in the real building and distance between the building shown by the application. All measurements are taken using sticky tape measures. Because of measurement complexity, every measurement requires 2 people involved, one operating camera and the other working with the tape measure and doing the necessary calculations.



Figure 17: View of the application building model: north elevation Snapphanevägen, Karlskrona.

The buildings included in the testing process are coming from Gdańsk and two of them are part of Gdańsk University of Technology one is the new building of the Faculty of Electronics, Telecommunications and Informatics (Figure 18), the other is an old building of Faculty of Mechanical Engineering (Figure 19). Both are placed rather near to each other, but they have different volumes and because of their directions of elevations they are also varying. The third building is a different type, it is part of a crowded city-centre in Gdańsk which is one of the most recognisable landmarks including a gate towards the main street of Gdańsk called the Golden gate (Figure 20). The test carried out for the Golden gate was due to it held with restricted access to some of the views and the span of measurements was also limited.



Figure 18: View of the application building model: ETI Faculty, Gdańsk.

Due to the building's placement it is also a place where plenty of tourists can cover the view and thus spoil the measurements. It is important to get the application to work and to be tested in different environments as augmented reality future appliances also might vary. The testing process was done during two different days with completely altering weather conditions. The first day was very sunny and dry however the second day was on the other hand rainy. This also affected the results, those gathered the second day clearly showed much worse accuracy even if the same building was being measured.



Figure 19: View of the application building model: Mechanical Engineering Faculty, Gdańsk.



Figure 20: View of the application building model: Golden gate, Gdańsk.

As mentioned here before every building was planned to be measured from 24 different positions. Due to different reasons, like hard access to the building from some directions or building composition (for instance when 10 metres was not far enough from the central point, to get outside the building), it was not always possible. For a model in an urban environment as the one in the centre of the city like the Golden gate in the central part of Gdańsk Old Town it means limited access from different angles forcing to drop the observation angles from 6 to 4. Sample views are presented in figures using different distances away from the real building 'Golden gate' building, here placed and seen from 40 metres (Figure 21), 70 metres (Figure 22), 100 metres (Figure 23).

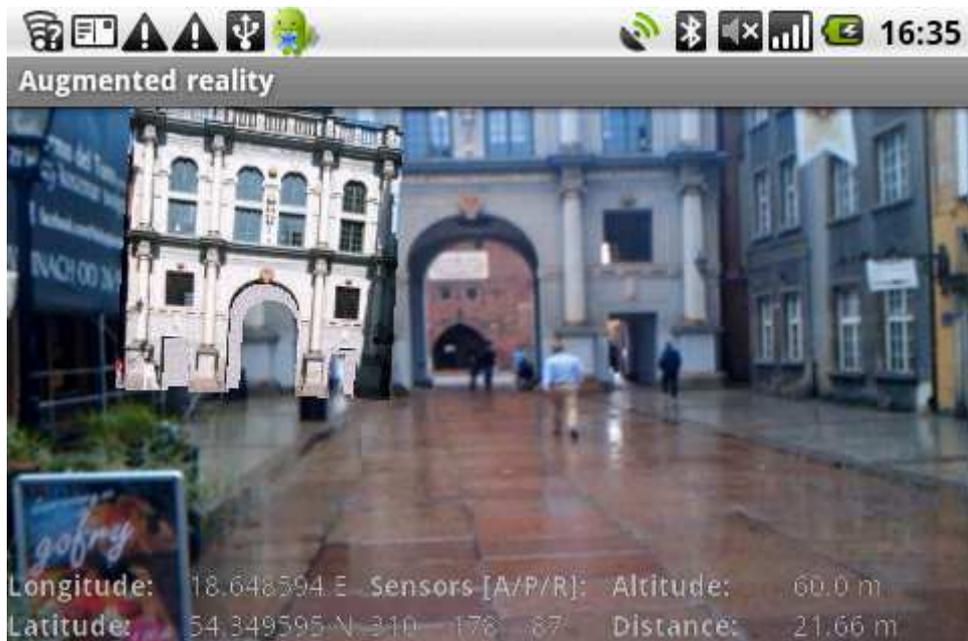


Figure 21: View of the application from 40 metres away, Golden gate, Gdańsk.



Figure 22: View of the application from 70 metres away, Golden gate, Gdańsk.

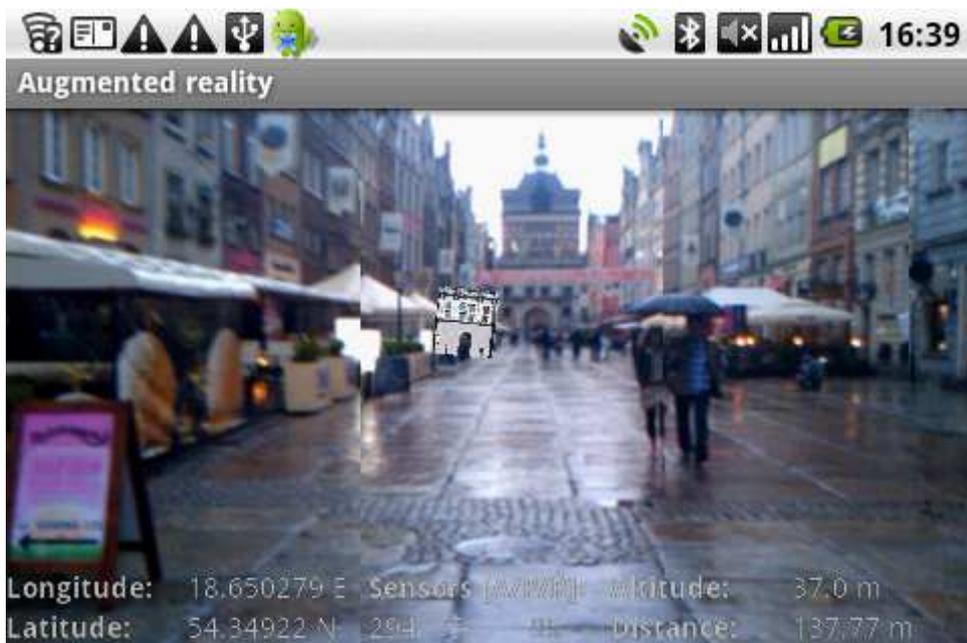


Figure 23: View of the application from 100 metres away, Golden gate, Gdańsk.

After conducting the tests, every measurement was written down in a measurement table. The measurements that could not be made were marked as 'X' meaning there is no result from this position. In order to estimate the importance of the different factors imposing the application complete accuracy some subtests had to follow. Also 5 random users were asked about 4 questions concerning the research part of the study. Answers were also written down and enabled analysis on users' point of view on the application.

A separate last test has been done in order to evaluate scope of the GPS receiver evaluated by the Android mobile device. To do this the application has received data sent by the *Location.getAccuracy()* public method available on the Android API. The tests have been carried out 3 times during different daytimes and every time the 10 first values were noted down. This should allow to evaluate the spectrum of Android common coordinate results.

## 8 RESULTS

The testing phase has revealed several interesting application properties. The augmented reality application has to encompass a lot of components that might have an influence on the final accuracy result, like the GPS and the magnetic sensor indications as well as the graphical library and coordinate calculation mechanism. While preparing the results some dependencies have been discovered.

Firstly it shows that the GPS accuracy has had a significant role during measurements. According to the GPS documentation and expectations shown in chapter 2 sometimes it took plenty of time, sometime even up to 10 minutes, after turning on the application and receiving first accurate GPS fixes with coordinates. The GPS indications are the biggest variables inside the application calculating mechanism. Unfortunately, it was a common fact that the coordinates coming from the GPS receiver were switched, even when the user holding the mobile phone remained still. The shift in such switching coordinates influenced on the value of distance calculated by the application changed even with more than 10 metres, which meant that the size of the building visualised was also varying significantly. Another observation was possible after taking the measurements from the same place in two different days as it happened with the Faculty of Electronics, Telecommunications and Informatics building model when once the measurement were taken during a sunny day and during the second day it was cloudy and almost raining. The results received were differing from each other and one could clearly see that those taken during the sunny day were more accurate. It is also caused by the GPS receiver reaction to satellites location, which cannot repeat twice the same, and signal strength, varying due to weather conditions.

As a result of the testing process all three places: Faculty of Electronics, Telecommunications and Informatics, Faculty of Mechanical Engineering and the Golden Gate building were visited and possible number of measurements were taken. Because both the academic buildings are relatively big in some of the cases distance the 10 metres distance from the central point was still inside the building. Because of that it was impossible to take the measure the discrepancy and the “difference” value was marked with ‘X’ in the corresponding table. The problem with the Golden Gate building was that, because it is a relatively smaller building and placed inside the city-centre some of the views were covered by tenements in its neighbourhood. That is why instead of 6 angles of observation there were only 4 organised which shall also be enough to evaluate the building accuracy. The final result of measuring whole three buildings can be seen below in Table 6 (the Faculty of Electronics, Telecommunications and Informatics building), Table 7 (the Faculty of Mechanical Engineering building) and Table 8 (the Golden Gate building).

Table 6: Result of measuring ETI Faculty model discrepancy:

Measurements		1	2	3	4	5	6
10 metres	Difference	X	X	3.0 m	11.1 m	X	X
	Distance	7.8 m	3.5 m	12.3 m	21.1 m	14.1 m	15.9 m
40 metres	Difference	18.2 m	19.2 m	7.0 m	11.4 m	13.2 m	6.9 m
	Distance	21.1 m	24.5 m	29.6 m	34.1 m	47.1 m	43.2 m
70 metres	Difference	27.2 m	13.7 m	2.5 m	9.3 m	7.4 m	10.9 m
	Distance	87.0 m	75.8 m	68.4 m	75.7 m	83.1 m	67.7 m
100 metres	Difference	22.1 m	24.7 m	29.2 m	13.6 m	15.2 m	19.8 m
	Distance	91.3 m	109.3 m	113.9 m	92.2 m	87.5 m	99.2 m

Table 7: Result of measuring Mechanical Engineering Faculty model discrepancy:

Measurements		1	2	3	4	5	6
10 metres	Difference	11.1 m	12.3 m	X	7.3 m	X	3.9 m
	Distance	21.1 m	14.3 m	16.3 m	14.2 m	11.8 m	21.6 m
40 metres	Difference	13.0 m	19.0 m	7.9 m	10.1 m	10.3 m	9.1 m
	Distance	56.2 m	61.4 m	51.8 m	67.3 m	49.5 m	53.2 m
70 metres	Difference	12.2 m	21.2 m	17.7 m	16.9 m	12.3 m	7.1 m
	Distance	78.9 m	87.7 m	90.6 m	63.4 m	65.3 m	66.6 m
100 metres	Difference	7.0 m	20.0 m	20.9 m	12.9 m	15.3 m	3.0 m
	Distance	112.5 m	125.9 m	115.9 m	118.9 m	91.2 m	109.7 m

Table 8: Result of measuring Golden gate model discrepancy:

Measurements		1	2	3	4
10 metres	Difference	7.1 m	11.1 m	3.4 m	5.9 m
	Distance	3.5 m	9.2 m	7.2 m	17.4 m
40 metres	Difference	12.6 m	17.1 m	20.2 m	6.0 m
	Distance	46.7 m	51.5 m	26.5 m	21.6 m
70 metres	Difference	5.9 m	16.4 m	25.3 m	11.4 m
	Distance	74.4 m	59.3 m	55.4 m	59.8 m
100 metres	Difference	8.6 m	13.4 m	21.4 m	25.4 m
	Distance	92.0 m	111.9 m	122.2 m	137.7 m

The first action after gathering the results of measurements from all three places was calculating the average results of ever group of measurements. It allowed to combine results received from the different angles and to pack them into groups containing results from the same distance. The result can be seen below in Table 9:

Table 9: The average values of the presented measurements:

Measurements		ETI Faculty	Mech. Eng. Faculty	Golden Gate
10 metres	Difference	7.05 m	8.65 m	6.88 m
	Distance	12.45 m	16.55 m	9.33 m
40 metres	Difference	12.65 m	11.57 m	13.98 m
	Distance	33.27 m	56.57 m	36.59 m
70 metres	Difference	11.83 m	14.57 m	14.75 m
	Distance	76.28 m	75.44 m	62.23 m
100 metres	Difference	20.77 m	13.18 m	17.20 m
	Distance	98.90 m	112.35 m	115.96 m

The Table 9 allows clear interpretation when looking at the “difference” value, which is measured in metres and the lower it is the better accuracy the application has. The “distance” value has to be interpreted in relation to the actual distance defining rows in the table (10 metres, 40 metres, 70 metres, 100 metres). It means that one needs to subtract the actual distance measured manually from the distance value shown by the application. For instance when we are referring to the distance measured by the application equal 12.45 metres when the real distance was equal 10 metres we already have 2.45 metres inaccuracy

which then was deepened by the “difference” coming from the table. In order to visualise it more clearly the result of such operation can be seen in Table 10.

Table 10: The difference between the distance measured manually and by the application:

Measurements	ETI Faculty	Mech. Eng. Faculty	Golden Gate
10 metres	2,45 m	6,55 m	-0,68 m
40 metres	-6,73 m	16,57 m	-3,41 m
70 metres	6,28 m	5,44 m	-7,78 m
100 metres	-1,10 m	12,35 m	15,96 m

The average results show different results in every section some tendencies can be seen however. First of all the general impression about the tested buildings has an evidence in the presented numbers. A general impression was concerning how the buildings were visually perceived and the observation was that the ETI faculty model was mostly appearing on the mobile phone’s screen closer than the real building whereas the Mechanical Engineering Faculty model was appearing further and thus it has been drawn smaller comparing to the real building image. It is proved in number presented in Table 10 as we can see that most of the average values referring to ETI faculty were negative whereas the average values describing Mechanical Engineering Faculty were positive. Because the values inside Table 10 are a result of subtraction the distance measured by the application from the real one where the value is positive the application treats the model as placed further away than it really should and that is why the building model is actually smaller than its real version. It is, of course, contrary when the values are negative as then the building appears closer and that is why the model displayed is bigger than the real building frequently covering the view. An example of this effect has been presented in Figure 24, presenting the ETI faculty where the real building has been covered by its model. In contrary in Figure 25 we can see the Mechanical Engineering model being drawn much smaller than the real building due to distance calculation discrepancy.



Figure 24: View of the ETI faculty as a sample of a building view too near.



Figure 25: View of the Mechanical Engineering Faculty as a sample of a building view too far.

Since there is a same method and there are two different results received on the same mobile device one has to wonder what is the underlying cause for that. One has to realise that impression of the distance the building is drawn is achieved by the distance measured by the application and the two factors have an actual influence on how it is measured i.e.: current GPS position and calculating method. Both can be inaccurate, however due to problems mentioned before concerning the GPS variation one might suspect that GPS inaccuracy plays a more significant role in estimating the distance inaccuracy.



Figure 26: Displacement in building allocation coming from Collada model, Golden gate, Gdańsk.

Concerning the Golden Gate it was generally drawn similarly as the real building but there was a different problem observed however. The building in general appeared as being moved towards one certain direction. As this tendency did not seem to be varying depending on different angles one might guess that there is some other fault not concerned with GPS indications. Since the calculating mechanism works in correlation to one constant point taken from the Collada file and chosen by the model author it is possible that this point has been estimated using wrong coordinate system which can add some more inaccuracies to

our system. Unfortunately when we display the Collada model in *GoogleEarth* one can see that there is a shift between where the model is drawn and where the shape of the building in the satellite photography is. According to Figure 26 it could even range few metres. This explains most of the problems concerning the Golden Gate as one can see in the Figure 21, Figure 22 and Figure 23 that the same displacement is preserved in those images.

In order to define the general inaccuracy of the application the minimum and the maximum results were defined after gathering the results from all three building models. We can see the final shape of such comparison in Table 11. The results bring some more conclusions. The closest distance in total was 2.5 whereas the biggest difference was 29.2 so in total the application has not reached inaccuracy bigger than 30 metres even when measured from 100 metres away. On the other hand there were also results close to optimal differing with the real building by only few metres.

Table 11: The minimum and the maximum results of the measurements in total:

Measurements		Minimum	Maximum
10 metres	Difference	3.0 m	12.3 m
	Distance	3.5 m	21.6 m
40 metres	Difference	6.0 m	20.2 m
	Distance	21.1 m	67.3 m
70 metres	Difference	2.5 m	27.2 m
	Distance	55.4 m	90.6 m
100 metres	Difference	3.0 m	29.2 m
	Distance	87.5 m	137.7 m

The results of the application's general accuracy are presented in Table 11. As it has been mentioned above there are several factors forming the application general accuracy i.e.: inaccuracy of the GPS system, inaccuracy of the magnetic sensors, inaccuracy of the coordinates calculation system, inaccuracy of the buildings position coming from the *GoogleWarehouse*. It has been mentioned that the sample inaccuracy of the building position of the Golden Gate building was having around 7 metres 55 centimetres but it is not even the maximal inexactness possible. For a sample distance between the coordinates presented in Chapter 4 being 3 kilometres 870 metres and 40 centimetres long. The more correct distance shown by the GoogleMaps was equal to 3 kilometres 878 metres and 40 centimetres. This means that for a distance of 4 kilometres there is 8 metres discrepancy which means for a distance of hundred metres the discrepancy can be estimated as 20 centimetres. The inaccuracy of the magnetic sensors is technically very hard to predict and measure as there are no correct results that could be compared with the ones achieved on the mobile phone. Unfortunately some tests like laying the mobile phone still on the desk show that data coming from the sensors cannot be treated as confidential. Android however supports a method allowing to calculate the GPS data accuracy. It is possible to observe the accuracy of the GPS coordinates while using the method *Location.getAccuracy()*. A small test has been done in order to observe how the GPS receiver acts on the mobile device. A method showing the accuracy has been implemented and the GPS has been turned on the same place during three separate days, during different daytimes. First 10 different values of the accuracy estimation have been noted. The results are presented in Table 12 (in columns from 1 to 10). As we can see at first the results were containing more than 50 metres inaccuracy. This might indicate that it was working with the A-GPS mode. During the process the accuracy was improving, however the range during all the tests was from 2 to 64 metres which shows how vital for the whole application the GPS receiver is.

Table 12: Results of the GPS receiver accuracy test:

	Day I	Day II	Day III	Mean	Amplitude
1	54 m	64 m	48 m	55.3 m	16 m
2	48 m	48 m	32 m	42.7 m	16 m
3	32 m	32 m	16 m	26.7 m	16 m
4	24 m	24 m	8 m	18.7 m	16 m
5	16 m	16 m	6 m	12.7 m	10 m
6	8 m	8 m	4 m	6.7 m	4 m
7	6 m	6 m	2 m	4.7 m	4 m
8	4 m	4 m	3 m	3.7 m	1 m
9	2 m	8 m	2 m	4 m	6 m
10	3 m	12 m	4 m	6.3 m	8 m

The order of the GPS accuracy indicated by the mobile phone was descending (Figure 27) which meant that during the mobile operation it was receiving more and more précised fixes with coordinates. The best of the received accuracy was 2.0 metres. However time of receiving so good results in all cases exceeded 10 minutes of waiting which is too much for an ordinary user of the application and it is also more than standard time of the measurements taken in this study.

The final steps from all three days were compared together (by calculating a mean of every step during all three days and the amplitude of all three values) and some tendencies could have been observed. If one only interprets Figure 27 it is visible that the accuracy is improving by pointing smaller and smaller values. This means that the final step (the tenth one) should have the better mean of all. The study reveals that this is not true as it was the eighth step to have the better mean value of all the steps together (and also the smaller amplitude on the same hand). When one takes a closer look on the Figure 27 it can be seen from the shape of the graph. It is because the observer was moving constantly changing the

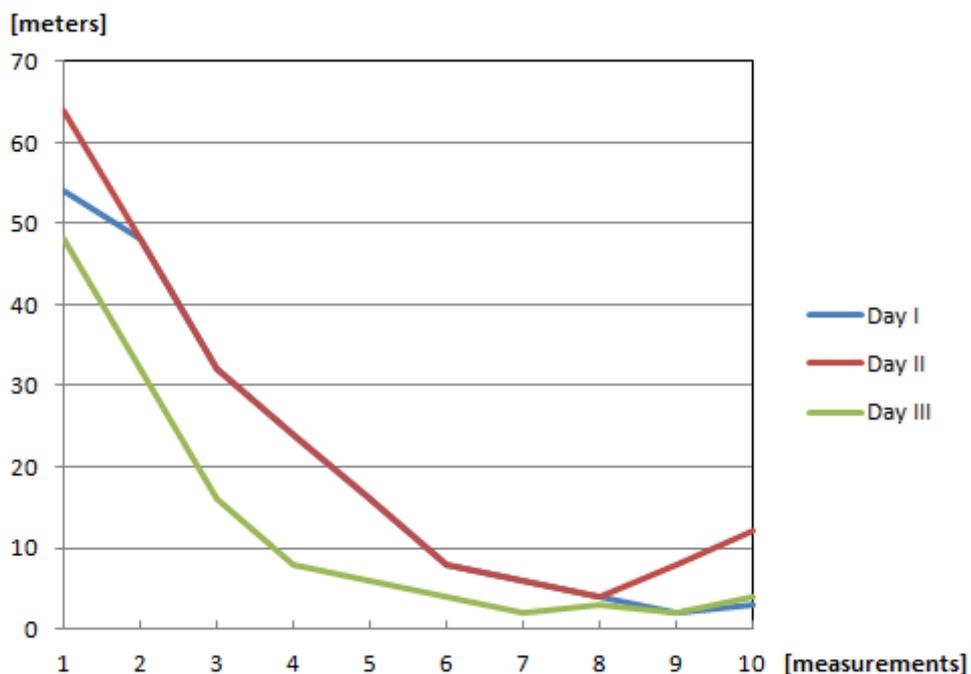


Figure 27: Graph showing the GPS accuracy progress.

position, forcing the GPS to update the coordinates. It is believed that after some time when the probable accuracy drops from 60 metres to just 2 and the curve leans down the upcoming coordinate fixes are spoiled and require more time for improving. This makes the shape of the graph looking like a sine wave, with ups and downs (for instance from 8 metres to 2 metres of probable inaccuracy), showing that there is an accuracy that the GPS cannot exceed and also the device has problem with sustaining it at a certain level.

Apart from all, the basic functionality was achieved, measuring the model position was always possible after pointing the mobile phone in the direction of the real building. If the building was started getting off target of the viewer building model displayed on the screen was also disappearing. In most cases the building model did move towards the same direction as the real building was appearing on the screen. It is moreover investigated in the survey concerning the usable part of the research. 5 random users picked from a group of passers-by curious about the research who were given the mobile phone the application installed and turned on so they could test the functionality for a time they needed. Then they were asked about the basic questions:

Q1: How much did in your opinion the inaccuracy spoil the effect of the augmented reality?

Q2: How much did the effect change viewed from different angles?

Q3: How much did the effect change viewed from different distances?

Q4: How did you like the quality of the presented building model?

With the possible answers: 4 – Very much, 3 – Much, 2 – Somewhat, 1 – Not at all. The answers received from users are presented in the Table 13, where the rows stand for the number of users and Q1, Q2, Q3, Q4 are the number of questions above.

Table 13: Results of the user's survey:

	User 1	User 2	User 3	User 4	User 5
Q1	2	1	3	3	4
Q2	3	2	2	2	2
Q3	2	2	1	4	3
Q4	4	4	3	4	4

Summarising replies above, it appears that the general application condition is enough to enable future work within this area. When combining the results from Table 10 and Table 11 completed with the results presented in Table 13 one can see that the results are respectively satisfactory, not exceeding certain ranges of inaccuracy. This survey was conducted in order to fulfil answers to research questions and as so enabled the conclusions in the presented field. Answers to Q1 show that opinions about the accuracy were rather divided with a slight advantage showing that it might have been a bigger threat (Figure 28).

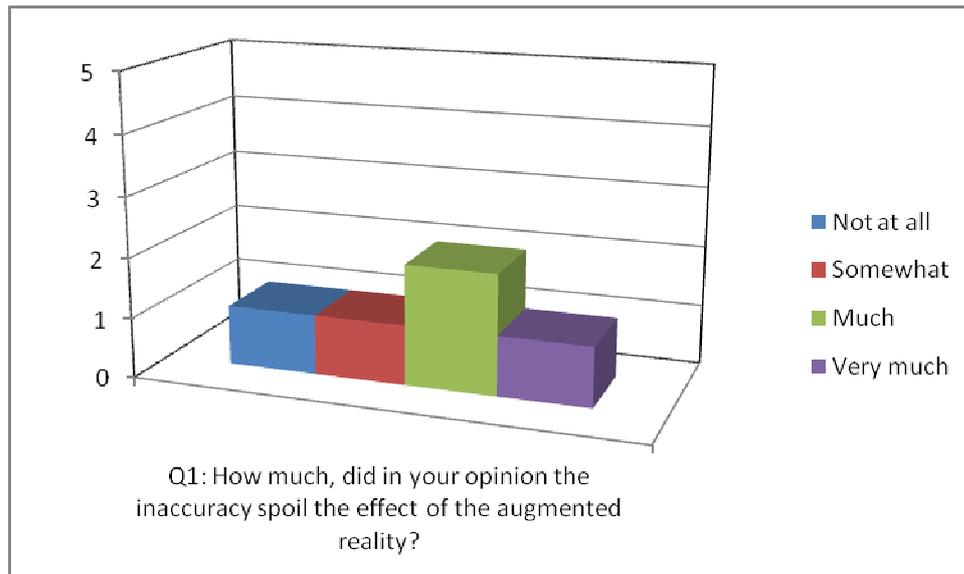


Figure 28: Graph showing distribution of answers to Q1.

On the other hand the respondents were agreeing that the influence on accuracy seen from different angles was existing however not very significant. This might tell that magnetic sensors, varying while the building was seen from different angles, did not have much influence on the final accuracy. Different responds were shown were considering the accuracy seen from various distances were users could not conclusively decide upon the influence on inaccuracy (Figure 29). Despite poor results of the measurements the answers were inconclusive, which shows that the users might have been satisfied or not very demanding towards the application.

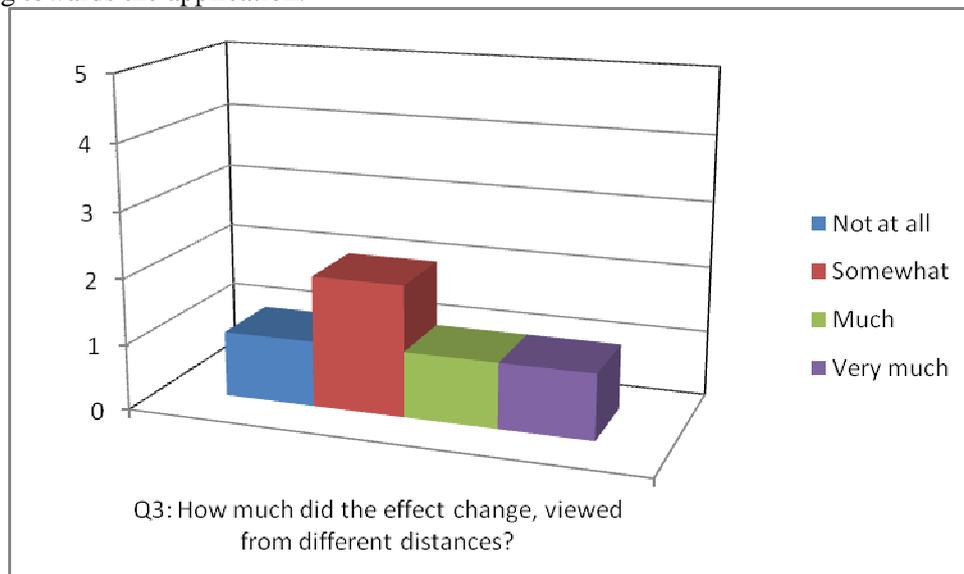


Figure 29: Graph showing distribution of answers to Q3.

A very satisfying results were received while asking Q4, whereas most of the users agreed the quality was fully satisfying their expectations.

Such an application might become a good start-up for future works and after improving some of the faulting factors, like the GPS accuracy the measurements using this method would be lowered. The survey revealed that the quality of the model was satisfying the user needs. Also the general quality of the application is moderately affected by the accuracy of the application, which means that despite the fluctuating accuracy the effect of the new reality was not always spoiled from the user's point of view.

Some of the results of the application were also photographed using DDMS Eclipse tool and the result is visualised in the different following figures, see below (Figure 30, Figure 31, Figure 32).



Figure 30: Result of the application from, Mechanics engineering Faculty, Gdańsk.



Figure 31: Result of the application from, ETI Faculty, Gdańsk.



Figure 32: Result of the application from, Golden gate, Gdańsk.

## 9 CONCLUSIONS AND FUTURE WORK

After ultimately finishing the testing part there is a time for summarising the whole study. The major assumptions i.e.: to create the augmented reality application visualising virtual building models, to evaluate this application in comparison to real-existing buildings and to bring conclusions about gathered results, were fulfilled. The application has been supplied with all the necessary functionality that is a clear interface displaying information received from the GPS sensors and from the magnetic sensors, several buildings attached into the set of displayed models, a calculating mechanism corresponding with the GPS coordinates and magnetic sensors values.

### 9.1 Conclusions

All of the features mentioned above were done in relation to the research questions degree of accomplishment of which the final result of the study shall allow to evaluate. The research questions were concerning the general application accuracy in relation to feature accuracy and users satisfaction which were the major area of interest during this whole study. It is good to start the conclusions with assessment of the research questions execution:

R.Q.1: Is it possible to reach a good level of accuracy and quality in the AR application using only commonly available mobile tools?

Yes, it is possible which is shown by the 'minimum' column in Table 11 and also figures in Chapter 8 (for instance Figure 30). The application has also been presented to unbiased people admitting that general quality of the augmented reality was satisfactory to their expectations and not denying its general performance. This proves that achieving good results and satisfying the needs of the users is possible. The direction of the presented building was mostly correct despite the fact that the distance between the model and the real building was sometimes varying significantly. Of course the measured discrepancies were differing and some were bigger than the other, but in general the application succeeded especially when realizing it was created using civilian purpose GPS receiver and standard geographical calibration which leaves much for future work. Mechanical features, like the GPS accuracy needs much better accuracy to be able to improve the application in the future. The differences from the desired results in general were if not better than expected, at least acceptable and satisfying. Also the user's opinions were presenting a moderate contentment with the way the application works. The only remark was that the quality was fluctuating from exact till far from satisfying that is why the assessments was varying from 1 to 4.

R.Q.2: How much, from the user's point of view, does GPS and magnetic compass sensors inaccuracy affect the AR effect?

The GPS influence is the issue that has the biggest meaning in application ultimate accuracy. It was especially apparent during tests when the building has significantly changed in position on the screen according to new coordinate input data received, with huge delay, from the GPS when the person holding camera stood still. Unfortunately the GPS and the magnetic sensors inaccuracy is an Achilles' heel of the application causing most of the discrepancy of obtained results. The most expected feared discrepancy was the GPS indication inaccuracy which was actually showing different than suspected results from different places around every building. Usually the GPS started to work after a few minutes of waiting for the results which was a problem also during the first seconds after the moment when it started receiving indications from the A-GPS system. Unfortunately the A-GPS system is having much worse accuracy than the standard GPS but the A-GPS has less

requirements as it can support on mobile facilities and thus posses information about user current position, that is why the system automatically starts looking for indications from the A-GPS when it cannot work with the GPS. There were also problems concerning the magnetic sensors results. There are no tests presented of the results achieved from the sensors depending on the mobile phone model used, but some devices are too sensitive, which is especially visible when the phone lies still on the desk and the compass application is turned on. The sensor results are fluctuating and the differences are significant. This has a big effect on the application and it was also shown during the tests as the results coming from different directions where variation in accuracy depends on direction and angles lead to increased discrepancy measured in Table 6, Table 7 and Table 8. The users' answers do not bring any conclusive data as 4 out of 5 users have described the changes occurring from different angles (i.e. does that might be causes by magnetic sensors' inaccuracy) as existing but not very important and only 1 out of 5 stated them as disturbing the effect significantly. This proofs that the imposition effect of the building and the camera view is rather not unambiguously spoiled by the magnetic sensors inaccuracy.

Also the users agreed on existence of the influence between the angle or distance and the final inaccuracy of the application (answers to Q2 and Q3). They claimed it is not a very vital issue (Figure 29 and Table 13). Worse answers were received while asking for the difference on inaccuracy while seeing the image from different distances which also means having different GPS coordinates which due to Figure 27 shows that probably the application was not working long enough as to get good a GPS accuracy and not every change of user's position resulted in a proper reaction of the GPS device which had an influence on the final result.

R.Q.3: Does application inaccuracy differ depending on the angle and distance?

Yes, as it can be read from Tables 6, 7 and 8 that there are different values obtained. Also in Table 9 we can see that the results are varying depending on different distances and angles. It is a reaction to two factors. Firstly, it is a result of the GPS inaccuracy as it was clear that the GPS was reacting to viewer changes of position but mostly not in the same way as it should, for instance when the viewer change position within 30 meters the GPS values were changed by 25 meters. Secondly is that it has been affected by the magnetic sensors because of its over-sensitivity was sending values that were rapidly changing and sometimes not showing the correct values when they were measured. Basically on the results coming from Table 9 we can draw a conclusion that the application works much better from further distances. It is because the perspective is narrowed and the possible errors concerned with the starting coordinates (shown by GPS) and calculating mechanism (used by the application) are less meaningful than when from closer distances. From the results presented in the Table 13 none of the respondents answered that there were no dependency between the accuracy and the angle. Most of the user's described this relation as existing but not very important. On the other hand user's opinion on the dependency between the distance and application accuracy were varying significantly which allows to say that this relation was existing and in one out of five cases it was disturbing.

R.Q.4: Can AR application with Google Warehouse's building models guarantee a satisfying quality of view?

Yes, it can! It has been mentioned before that the GPS and the compass' position are the biggest concern, however the model's quality is its biggest asset, which can be read from the Table 13 were almost every user were satisfied with the model quality. Of course work related with applying every model without a proper parser requires a lot of patience, however the final effect is definitely worth it. Due to the texturing mechanism and the fact that model are usually viewed from far away the textures are displayed on a small scale and low graphical processing abilities are not so important when transiting the building quality into the mobile services. Since textured buildings are consisting of planar images one cannot

influence the way the buildings are displayed and that is why the models are basically the same as their versions presented in the *GoogleEarth*, which is definitely a satisfying quality for such an application. Sometimes there were too many calculations for the graphical engine because of the number of textures and meshes in the building.

## 9.2 Future work

It is believed that the idea of AR is a promising technique, still having plenty of possibilities to be developed in the future. With some effort the field will advance forward and also because of existing tendencies, increasing simplicity and diverse appliances. One can believe that all the shortcomings might soon be solved or other technologies will be presented replacing current deficiencies. There are certain concerns listed that might or should be solved soon, such as:

- Creating an everyday use application.

This application from the very beginning is planned to be a prototype showing future possibilities for the AR technique. The conducted study has had a major assumption which was to evaluate the quality and accuracy of the displayed buildings. This project can evolve in the future and become a full feature application supplying user with all the necessary options from which one can choose the most suitable options. This can result in a wide variety of buildings available maybe even a full *GoogleEarth* set of models and also supplying the user with some basic information about the building, not only the user's position. In order to make this happen there shall be some helpful options created like a prerequisite-building-choice designed for each user allowing to pick up more than one building and enjoy functionality of such an application. Basing on the tendencies existing in the world of computer science we might expect that in the nearest future there will be several applications visualising building models.

- Allowing user-extensions.

The next step is to allow the user to create his own extensions, from which the first and the most important one is allowing to pick and apply buildings chosen by the user available from *GoogleWarehouse* or created using the Collada standard. It is important for this application that every single model is read manually and every single information line picked from the existing Collada model. For a full functional application there should be an independent parser created. Of course such a solution would require a good team of programmers as it is time consuming, but in order to extend the functionality of the application this would be the most important step. Such an extension should also allow a system to pick a whole set of buildings and display as many of them at once as it is possible. Because this could be done externally and then just installed on the mobile device such a parser shall not have any calculation power requirements.

- Creating cloud computing system.

To apply a set of buildings would slow down the application rapidly as it is already very complex and charging a graphical calculation system. The solution for this would be to create a cloud system dealing with all the complex calculations by itself and leaving the application the functionality not related with calculations, like displaying and showing the devices indications. The cloud system is a very popular functionality for the latest and most new systems. When applying this to AR technique it will require the Internet connection which would not be too hard to get in urban areas while using a mobile device with the WI-FI connection. It could store information about created private profiles containing information from places specified by the user and models applied by one.

- Using spherical triangles.

The application has much more to offer when it comes to accuracy of the presented data (visualised model of buildings) and there are several possibilities how this could be improved. First of all it is to use different data calibration methods. Within this application the applied method used was by calculating the longitudinal and latitudinal distance and unifying them together. This method however also brings some inaccuracy to the building's model position. For instance by taking the example from Chapter 4 when calculating the distance between point A (54.606563198N, 18.23382854 E) and B (54.616759527 N, 18.176429271 E). Using the method with longitudinal and longitudinal distance the distance was evaluated for 3 kilometres 870 metres 40 centimetres. The distance between those two points shown by the GoogleMaps' tools using exact calculating method has shown that the distance equals 3 kilometres 878 metres 40 centimetres. This in total gives equally 8 metres discrepancy. Of course this is a distance of nearly 4 kilometres which in comparison to 100 metres on which the application should work with, is a distance 40 times smaller which means the inaccuracy should reach 20 centimetres but it shows that there are still chances in gaining better accuracy of the application.

- Better accuracy achieved by using image retrieving algorithms.

The general assumption of this project is making it universal. It was even considered above to create an universal parser dealing with all the models stored in Collada format allowing users to extend the application functionality with their own models. However one of the solutions for getting better accuracy is treating every model separately. It is because every model is prepared for one particular place it should be presented within. As to make the application more successful one can search for outstanding features of the terrain where the model is placed and search for similarities while using the mobile phone's camera. Such a feature can be a horizon line, background or surface colour and some other buildings in the neighbourhood or some typical features of the place. The algorithms applying such methods are called 'image retrieval' which is a natural name because of their feature in analyzing what the user can see. Such a solution might help and improve the algorithm accuracy. For instance instead of considering the pitch value, the algorithm might check if there is a horizontal line somewhere in the image coming from the camera, for instance whereas the surface changes colour from green (on the grass) or gray (on the pavement or concrete surface) to blue (when sky is visible). Then the slope of thus discovered horizon line(s) should be analyzed and it might clutch the building model so that it will be displayed according to that line. This solution has also one disadvantage and it is dropping the universality of the technique, whereas the applications nowadays should be common and same for every user. Creating an application using image retrieval algorithms means that it has to attach the building to certain terrain and a research towards that should be done by a team of programmers, which is not applicable when we create application that is intended to be developed by the users themselves.

Also other possible extensions are possible. An interesting approach is combining advanced calibration methods and image retrieval methods. Such a solution shall be very complex as using it, it would be possible to normally display all building models with standard accuracy and apply image retrieval algorithms as an optional extension referring to the buildings researched in this way.

- Improving featured devices

The differences between "real distance" shown and the "measured distance" shown in the tests is mainly an effect of the composition of two major inaccuracies one coming from the algorithm used in distance calculation and the other related to the GPS inaccuracy. In order to make the whole augmented reality technique more accurate one has to use a

better positioning system. Right now the Global Positioning System provides the user with position accuracy ranging from a few metres which is not plausible when it comes to very small distances, like viewing the building from 10 metres or less. In the future the GPS will keep developing and civil systems will be revealed having much better accuracy but also there is a chance that some other positioning systems will be created, right now Glonass, which is a Russian system which exists but it has worse accuracy and is not so common. However there are also plans of launching the Galileo Positioning System created by the European Union countries and a Compass navigation system by China. The future will show if those systems will start competing in accuracy and if they will be able to indicate exact user's position which would allow such AR applications being much more accurate.

The difference or discrepancy between the real building position and the graphical position was due to magnetic sensors inaccuracy. Unfortunately this is the application's worst element as it is spoiling the building to show correct position. Whereas the GPS inaccuracy only cause a few metres of the so called "building shift" or changes but still there are other graphical factors to be aware of like the size of the building (size constancy), the compass inaccuracy often changes the slope of the building towards horizon line which created an impression so the building looks different than it should or distorting consistency of the building position as the model was floating on the screen. A lot of work has to be done by the mobile designers so that the magnetic sensors will work more fluently and directly affect such application less.

- Applying WGS relation system or other method in determining altitude

The full AR effect would be achieved if the application would not only determine the users latitude, longitude but also be able to indicate the user's altitude. This would allow the application to supply the user with full 3D functionality. It would be a great solution when the user is having a good viewpoint on the building attached in the application. One could for instance have the upside perspective view on the attached building. To do this application one needs to display user's current position in relation to ground level. Unfortunately altitude position indicated by the GPS receiving system is in accordance to WGS-84 system which describes the ellipsoidal model of the Earth not taking the terrain inclines into the consideration. For improvements a calculating system relating distance shown by GPS and position in WGS-84 should be applied so the application is able to show user's current altitude. There are also other possibilities in determining user's position, for instance with calculating current pressure. In the era when programmers are able to determine which keys one has pressed according only to the recent sensor logs (Cai, Chen, 2011) determining users current position does not seem to be so hard.

Summarising the study it shows that applying the augmented reality techniques in programming applications, created for smartphones, is useful and the technique is ready for everyday use. The conducted study is just one step towards creating a complex application visualising buildings. No bigger obstacles were presented and concerning programmers they have all the necessary tools to be able to create useful supplements towards establishing the augmented reality applications designed for certain human activities.

It has been shown that mobile phones already have all the necessary equipment towards visualising building models in such a reality and this feature may further be investigated. Helpful appliance can be useful for real estate developers, architects and constructors who can from now on be able to present their own building projects in reality and persuade potential customers to become interested in developing such. There are several other domains in where usefulness of applied visualised buildings can be used, such as navigation systems, merchandise for tourists who can visualise development of given place in different periods or read information concerning particular buildings. Due to all of the facts mentioned before it is believed that augmented reality can come to play a major role for the market of mobile- and virtual applications, in the nearest future and visualising buildings might become one of its most valuable appliance.

## APPENDIX A

### Articles included in the list of references after the literature review:

1. Adrados C., Girard I., Gendner J.P., Janeau G., 2002, *Global Positioning System (GPS) location accuracy improvement due to selective availability removal*, *Comptes Rendus Biologies* 325 (2002), pages 165–170.
2. Azuma R., Baillot Y., Behringer R., Feiner S., Julier S., MacIntyre B., 2001, *Recent Advances in Augmented Reality*, *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pages 34-47.
3. Azuma R.T., 1997, *A survey of Augmented Reality*, [in *Presence Teleoperators and Virtual Environments*, pages 355-385].
4. Bao-Yen Tsui J., 2000, *Fundamentals of Global Positioning System receivers*, John Wiley & Sons, inc.
5. Falaki H., Mahajan R., Kandula S., Lymberopoulos D., Govindan R., Estrin D., 2010, *Diversity in smartphone usage*, Proceedings of the 8th international conference on Mobile systems, applications, and services, June 15-18, 2010, San Francisco, California, USA.
6. Feiner S.K., Höllerer T.H., 2004, *Mobile Augmented Reality*, [in Karimi H., Hammad A., *Telegeoinformatics, Location-Based Computing and Services*, 2004].
7. Honkamaa P., Siltanen S., Jäppinen J., Woodward C., Korkalo O., 2007, Interactive outdoor mobile augmentation using markerless tracking and GPS, [In Proc. Virtual Reality International Conference (VRIC), Laval, France, April 2007, pages 285-288].
8. Julier S., Baillot Y., Lanzagorta M., Brown D., Rosenblum L., 2000, *BARS battlefield augmented reality syste*, NATO Symposium on Information Processing Techniques for Military Systems. Istanbul, Turkey, October 2000.
9. Kaplan E.D., Hegarty C.J., 2006, *Understanding GPS - Principles and Applications*, Artech House, London.
10. Kato H., Billingham M., 1999, *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System*, [in Proc. IWAR '99, San Francisco, CA, October 20-21, 1999, pages 85-94].
11. Lewis. R., 1998, *Global positioning systems*, [Chapter in Hohl P., 1998, *GIS data conversion: strategies, techniques, and management*, OnWord Press, pages 251-258].
12. Wagner D., Gruber L., Schmalstieg D., 2008, *Augmented Reality on Mobile Phones*, Charles River Media.
13. Webster A., Feiner S., MacIntyre B., Massie W., Krueger T., 1996, *Augmented reality in architectural construction, inspection and renovation.*, [in *Proceedings of the ASCE third congress on computing in civil engineering*, Anaheim, CA, 17-19 June 1996, 913-919].
14. Whitfield T., 2010, *Augmented reality [AR]: The technology is now market ready*, Posted 25<sup>th</sup> of January 2010 on Econsultancy, Digital Marketers United webportal (<http://econsultancy.com/uk/blog/5293-augmented-reality-ar-the-technology-is-now-market-ready>).
15. You S., Neumann U., Azuma R., 1999, *Hybrid inertial and vision tracking for Augmented Reality registration*, [In Proceedings of IEEE Virtual Reality, IEEE CS Press, Los Alamitos, Calif., 1999, pages 260-267].

## APPENDIX B

### Other articles included in the list of references:

1. Boyes W., Melvin M., 2010, *Economics*, eighth edition, Cengage learning, page 75.
2. Bryman A., 2006, *Integrating quantitative and qualitative research: How is it done?*, [In *Qualitative Research vol. 6*, Sage Publications, pages 97-113].
3. Cai L., Chen H., 2011, *TouchLogger: Inferring Keystrokes On Touch Screen From Smartphone Motion*, Research paper.
4. De Canck M.H., 2007, *Ionosphere Properties and Behaviors - Part 9*, *antenneX* Issue No. 118 – February 2007.
5. De Paor D.G., Whitmeyer S.J., 2010, *Geological and geophysical modeling on virtual globes using KML, COLLADA and Javascript*, [In *Computer & Geosciences* 37, 2011, pages 100-110].
6. Djuknic G.M., Richton R.E., 2001, *Geo-location and assisted GPS*, *IEEE Computer*, vol. 34, no. 2, 2001, pages 123-125.
7. Kothari C.R., 1985, *Research methodology: Methods and techniques*, New Delhi, Wiley Eastern Limited.
8. Książak M., 2002, *GPS – Construction and appliance of satellite navigation system /GPS – Budowa i zastosowanie systemu nawigacji satelitarnej/*, Thesis seminary 2001/02 presented for Rzeszów University of Technology in Department of avionics and control, Faculty of Mechanical Engineering and Aeronautics [Written in Polish].
9. Pelczar M., Szeliga J., Ziółkowski J., 1978, *Outline of Cartography and Topography /Zarys Kartografii i Topografii/*, University of Gdańsk, 1978 [Written in Polish].
10. Simmon R., 2011, *Map projections matter*, [Article published in: *NASA Earth Observatory*, February 24<sup>th</sup> 2011, <http://earthobservatory.nasa.gov/blogs/elegantfigures/2011/02/24/map-projections-matter/>].
11. Suomela, R., 2006, *Constructing and Examining Location-Based Applications and Their User Interfaces by Applying Rapid Software Development and Structural Analysis*, Ph.D. Thesis, Tampere University of Technology, Finland.
12. The Khronos Group, 2007, *Collada refinery definition on media-wiki web-portal*, [[http://www.collada.org/mediawiki/index.php/COLLADA\\_Refinery](http://www.collada.org/mediawiki/index.php/COLLADA_Refinery)].
13. van Diggelen F., 1998, *GPS Accuracy: Lies, Damn Lies, and Statistics*, [In *GPS World* 9, pages 41-45, 1998].
14. Woodward C., 2009, *Project summary Augmented Reality for Building and Construction (AR4BC)*, VTT technical research centre of Finland, 9th February, 2009
15. Zöllner M., 2009, *Augmented Reality for historical cities*, [In *Fraunhofer IGD, Symposium Virtual Historical Cities*, 6th November, 2009].

## REFERENCES

1. Adrados C., Girard I., Gendner J.P., Janeau G., 2002, *Global Positioning System (GPS) location accuracy improvement due to selective availability removal*, Comptes Rendus Biologies 325 (2002), pages 165–170.
2. Azuma R., Baillot Y., Behringer R., Feiner S., Julier S., MacIntyre B., 2001, *Recent Advances in Augmented Reality*, *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pages 34-47.
3. Azuma R.T., 1997, *A survey of Augmented Reality*, [in *Presence Teleoperators and Virtual Environments*, pages 355-385].
4. Bao-Yen Tsui J., 2000, *Fundamentals of Global Positioning System receivers*, John Wiley & Sons, inc.
5. Boyes W., Melvin M., 2010, *Economics*, eighth edition, Cengage learning, page 75.
6. Bryman A., 2006, *Integrating quantitative and qualitative research: How is it done?*, [In *Qualitative Research vol. 6*, Sage Publications, pages 97-113].
7. Cai L., Chen H., 2011, *TouchLogger: Inferring Keystrokes On Touch Screen From Smartphone Motion*, Research paper.
8. De Canck M.H., 2007, *Ionosphere Properties and Behaviors - Part 9*, *antenneX* Issue No. 118 – February 2007.
9. De Paor D.G., Whitmeyer S.J., 2010, *Geological and geophysical modeling on virtual globes using KML, COLLADA and Javascript*, [In *Computer & Geosciences 37*, 2011, pages 100-110].
10. Djuknic G.M., Richton R.E., 2001, *Geo-location and assisted GPS*, *IEEE Computer*, vol. 34, no. 2, 2001, pages 123-125.
11. Falaki H., Mahajan R., Kandula S., Lymberopoulos D., Govindan R., Estrin D., 2010, *Diversity in smartphone usage*, Proceedings of the 8th international conference on Mobile systems, applications, and services, June 15-18, 2010, San Francisco, California, USA.
12. Feiner S.K., Höllerer T.H., 2004, *Mobile Augmented Reality*, [in Karimi H., Hammad A., *Telegeoinformatics, Location-Based Computing and Services*, 2004].
13. Honkamaa P., Siltanen S., Jäppinen J., Woodward C., Korkalo O., 2007, *Interactive outdoor mobile augmentation using markerless tracking and GPS*, [In Proc. Virtual Reality International Conference (VRIC), Laval, France, April 2007, pages 285-288].
14. Julier S., Baillot Y., Lanzagorta M., Brown D., Rosenblum L., 2000, *BARS battlefield augmented reality syste*, NATO Symposium on Information Processing Techniques for Military Systems. Istanbul, Turkey, October 2000.
15. Kaplan E.D., Hegarty C.J., 2006, *Understanding GPS - Principles and Applications*, Artech House, London.
16. Kato H., Billinghurst M., 1999, *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System*, [in Proc. IWAR '99, San Francisco, CA, October 20-21, 1999, pages 85-94].
17. Kothari C.R., 1985, *Research methodology: Methods and techniques*, New Delhi, Wiley Eastern Limited.
18. Księżak M., 2002, *GPS – Construction and appliance of satellite navigation system /GPS – Budowa i zastosowanie systemu nawigacji satelitarnej/*, Thesis seminary 2001/02 presented for Rzeszów University of Technology in Department of avionics and control, Faculty of Mechanical Engineering and Aeronautics [Written in Polish].
19. Lewis. R., 1998, *Global positioning systems*, [Chapter in Hohl P., 1998, *GIS data conversion: strategies, techniques, and management*, OnWord Press, pages 251-258].
20. Pelczar M., Szeliga J., Ziółkowski J., 1978, *Outline of Cartography and Topography /Zarys Kartografii i Topografii/*, University of Gdańsk, 1978 [Written in Polish].

21. Simmon R., 2011, *Map projections matter*, [Article published in: NASA Earth Observatory, February 24<sup>th</sup> 2011, <http://earthobservatory.nasa.gov/blogs/elegantfigures/2011/02/24/map-projections-matter/>].
22. Suomela, R., 2006, *Constructing and Examining Location-Based Applications and Their User Interfaces by Applying Rapid Software Development and Structural Analysis*, Ph.D. Thesis, Tampere University of Technology, Finland.
23. The Khronos Group, 2007, *Collada refinery definition on media-wiki web-portal*, [[http://www.collada.org/mediawiki/index.php/COLLADA\\_Refinery](http://www.collada.org/mediawiki/index.php/COLLADA_Refinery)].
24. van Diggelen F., 1998, *GPS Accuracy: Lies, Damn Lies, and Statistics*, [In *GPS World 9*, pages 41-45, 1998].
25. Wagner D., Gruber L., Schmalstieg D., 2008, *Augmented Reality on Mobile Phones*, Charles River Media.
26. Webster A., Feiner S., MacIntyre B., Massie W., Krueger T., 1996, *Augmented reality in architectural construction, inspection and renovation.*, [in *Proceedings of the ASCE third congress on computing in civil engineering*, Anaheim, CA, 17-19 June 1996, 913-919].
27. Whitfield T., 2010, *Augmented reality [AR]: The technology is now market ready*, Posted 25<sup>th</sup> of January 2010 on Econsultancy, Digital Marketers United webportal (<http://econsultancy.com/uk/blog/5293-augmented-reality-ar-the-technology-is-now-market-ready>).
28. Woodward C., 2009, *Project summary Augmented Reality for Building and Construction (AR4BC)*, VTT technical research centre of Finland, 9th February, 2009.
29. You S., Neumann U., Azuma R., 1999, *Hybrid inertial and vision tracking for Augmented Reality registration*, [In *Proceedings of IEEE Virtual Reality*, IEEE CS Press, Los Alamitos, Calif., 1999, pages 260-267].
30. Zöllner M., 2009, *Augmented Reality for historical cities*, [In *Fraunhofer IGD, Symposium Virtual Historical Cities*, 6th November, 2009].