Graz University of Technology

Institute for Computer Graphics and Vision

Master's Thesis

Efficient user interface for accurate registration of GIS data on mobile devices

Saif Aldeen Alsaifi

Graz, Austria, 2013

Thesis supervisors

Univ. Prof. DI Dr. techn. Dieter Schmalstieg

Dr. Dipl.-Med.Sys.Wiss Stefanie Zollmann

To my father

Abstract

New technologies such as Smartphones play a major role in the modern applications development. Combining those technologies with other powerful information technologies (IT) provides us with applications, that can optimize our work processes, making them more efficient and more reliable. Utility workers use paper or digital GIS plans to have foreknowledge about what is located underground. However, AR GIS technology could supply workers with same information the plans provide, in a more efficient and effective way. It would then be possible to visualize the underground infrastructure as an interactive 3D model, get information or descriptions about different objects in the field, and plan the construction site. Such AR GIS applications require accurate data measurements to be used effectively. Modern smartphones and tablets are supported with different kinds of sensors (GPS, accelerometer, etc.) that provide such data measurements, however their level of accuracy does not meet the required level for registering the GIS data with the objects in the real world. The purpose of this work is to provide an interactive user interface for correcting this inaccuracy. This interactive interface gives the user the ability to correct the inaccuracy by manually matching the virtual objects with the real objects.

Keywords. AR; Smartphone; Tablets; GIS; AR GIS; 3D Visualization

Zusammenfassung

Neue Technologien wie Smartphones spielen eine wichtige Rolle in der modernen Anwendungsentwicklung. Diese Technologien in Kombination mit weiteren, aktuellen Informationstechnologien liefern uns Anwendungen, welche unsere Arbeitsprozesse optimieren und sie somit effizienter und zuverlässiger machen können. Die Mitarbeiter von Energieversorgern verwenden zumeist Papier- oder digitale Pläne aus Geoinformationssystemen (GIS), um unterirdische Einbauten lokalisieren zu können. Durch die Verbindung von Augmented Reality (AR) und GIS Technologien kann die selbe Informationen, wie sie Pläne liefern, in einer effizienteren und effektiveren Weise übermittelt werden. Damit lassen sich unterirdische Infrastrukturen als interaktives 3D-Modell visualisieren, Zusatzinformationen zu Objekten abfragen oder Baustellen vor Ort planen. Solche AR GIS-Anwendungen erfordern genaue Daten bzw. Messungen um effektiv genutzt werden zu können. Moderne Smartphones und Tablets verfügen über diverse Arten von Sensoren (GPS, Beschleunigungssensor, etc.), welche zwar solche Messdaten liefern, deren Genauigkeit allerdings nicht das erforderliche Maß für die Registrierung der GIS-Daten mit den Objekten der realen Welt erreichen. Das Ziel dieser Arbeit ist es, die Möglichkeiten einer interaktiver Benutzeroberfläche zur Korrektur dieser Ungenauigkeiten zu untersuchen. Mit einer interaktive Oberfläche wird dem Benutzer die Möglichkeit geboten, die Ungenauigkeiten durch manuelle Eingaben zu korrigieren und die virtuellen Objekte den realen Objekten anzugleichen.

Acknowledgments

It is my pleasure to thank all the people who made this thesis possible. At the beginning i would like to thank my colleagues in GRINTEC GmbH, for their support, collaboration and help. I especially would like to thank Monika Ranzinger, who introduced me into this work, Günther Gleixner, who believed in me and provided everything i needed to achieve this work, and Dominik Gärner, who supported me in every step during the implementation and testing of my work.

I would like to express my gratitude to my supervisors from the institute of Computer Graphics and Vision at Graz University of Technology, Prof. Gerhard Reitmayr, for his guidance, contribution and ideas, Prof. Dieter Schmalstieg, for his support and help, and Stefanie Zollmann, for directing me to the right way and teaching me how to make everything correct.

I also would like to thank my family for helping me along the way and praying to achieve my Goals. To my lovely wife, thank you for believing in me, and for being there all the time for me.

Finally, i especially would like to thank my friend Dr. Karim Morsi for his support and help.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Geographic Information Systems (GIS) have various branches and usages. They are applied in many fields and areas, for example, in business, education, government projects, natural resources surveys and utilities.

Utility workers use GIS mainly to monitor the underground infrastructures and supervise all changes and modifications. Usually, they perform surveys in the field, also digging is required for fixing or maintaining the underground infrastructures. Therefore, it is important to have a foreknowledge about what is located under the ground. GIS is able to provide this kind of knowledge, since it stores a large amount of asset data.

Usually, the utility workers use paper or digital plans to have foreknowledge about what is located underground. These plans provide the workers with sufficient information about underground infrastructure, locations, as well as other important measurements. However, with the AR technology we can provide the workers with the same information that the plans provide but in a more efficient and effective way. Using AR technology one can visualize the underground infrastructure as an interactive 3D model, get information or description about a specific object by clicking that object, be able to delete or modify objects, or even add new objects. Such capabilities do not help us only to have a foreknowledge about the underground infrastructure but also to plan the construction projects and visualize the objects in their places before actually installing them.

The purpose of this work is to provide utility workers with a mobile AR GIS application that substitutes traditional plans and makes their work processes more effective.

1.2 Contribution

Such AR GIS applications require accurate data measurements to be used effectively, on the other hand, modern mobile devices are supported with different kind of sensors (GPS, accelerometer, etc.) that provide such data measurements, but their level of accuracy does not meet the required level. In this work, we are introducing an AR GIS visualization application that register the GIS data on a mobile platform. Due to sensors inaccuracies, the registration of GIS data on mobile devices would help utility workers visualize the infrastructure at the construction site inaccurately, leading to misalignment in the rendered data. For this reason our AR GIS application will be equipped with interactive tools that assist the user to correct misalignments. The correction tools are applied after data rendering on the the device screen. The user could then correct any misalignments using these tools, by rotating and shifting the rendered data around the different axes.

In the second chapter **Related Work**, we will present existing technologies and techniques that are related to our work. We discuss Mobile devices and their power, Geographic Information Systems and their benefits, the concept of Augmented Reality, and finally discuss existing contributions and applications in this field.

The third chapter, **Approach**, describes our approach for visualizing asset information in an AR overlay. In this context, we provide details about sensor data and the different coordinates systems used in this approach.

In the **Implementation** chapter, we introduce a prototype that implements our approach and see how the different technologies are combined together, we also present the manual correction method that the user applies for fixing and correcting the inaccuracies from sensor measurements.

Results and Discussion, in this chapter we discuss the results that we observed during implementing and testing the prototype in the field and suggest several solutions and enhancements.

Finally, **Conclusions and Future work** discusses what we learned during the development of the tool and what we can do in the future to improve it and make the correction method more attractive and useful.

Chapter 2

Related Work

Contents

New technologies such as Smartphones play a major role in the modern applications development. Combining those technology with other powerful information technologies provides us with applications, that can optimize our work processes, making them more efficient and more reliable.

2.1 Introduction

The combination of modern technologies requires a good understanding of each technology separately. However, discussing each one of these technologies in detail is beyond this thesis. Instead we will confine ourselves by introducing these technologies and discuss the major ideas that are related to our work.

Firstly we start with smartphones and tablets, discussing their power, features and usages, followed by the concept of Augmented Reality will be represented with examples that illustrate its idea. Furthermore we will take a look at Geographic Information Systems and discuss their usages and benefits. Finally, we will present existing applications that combine these technologies together.

2.2 Mobile Platforms

Earlier, mobile phones had very basic functionalities, mainly limited to making phone calls and sending text messages. With the existing smartphones and tablets we can do much more than that. Smartphones and tablets are in fact fully functioning computers that fit in our pockets. In many situations such devices are considered to be better than computers, providing better features and usages. For example, as a tourist and while you are using your Smartphone to search for the nearest hotel or restaurant, you can share this with your friend who is living in a different place by sending him some pictures or invite him to a video call via Skype. It will be very hard to perform such things using computers.

In this work we will concentrate on the latest technologies of mobile devices and see their power and capabilities.

2.2.1 Mobile devices, their usage, their power and their spread

Smartphones and tablets provide us with endless capabilities. Their power actually lies in their usage, as the usage of smartphones and tablets varies from normal everyday use and expert professional use, satisfying the needs of almost every group and category. Nowadays there are many industries, companies and even individuals depending entirely on them.

In fields of business, communication, internet and social media, music and videos, education, entertainment, navigation, 3D games and more, we find plenty of apps that make the processes in each one of these fields much faster and more attractive.

Figure 2.1 shows the number of Smartphone owners in various countries in 2012 and how the internet usage with Smartphones is growing faster than desktop computers usage. This is obvious, as the integration of all features that exist in any Smartphone makes it a versatile device, and when we add the mobility feature as well, we could achieve many things which are very hard to be achieved using a desktop or a laptop computer.

On the other hand, the existence of delivery channels and markets such as App Store and Google Play allows the users to browse and download books and magazines, music and movies, television programs and many others apps in seconds.This makes it very easy for developers to distribute and sell their apps all over the world.

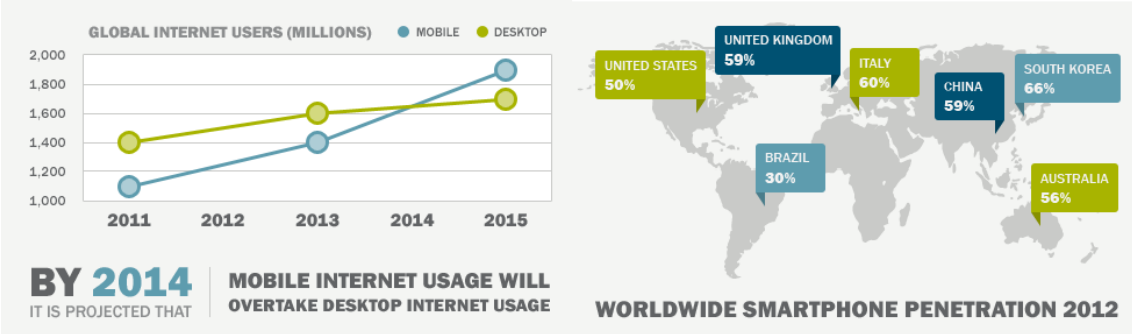

Figure 2.1: Depicts how the internet usage with Smartphones is growing faster than desktop computers usage, and the number of Smartphone owners in various countries in 2012 [5].

2.2.2 Mobile devices features and apps

Technical features of recent smartphones and tablets added new dimensions to the traditional applications. For example, we can use the GPS feature in a normal desktop computer but we will not be using it as it is meant to be used, because the mobility feature makes the use of GPS more applicable and needable.

Figure 2.2 depicts examples of different Apps from different fields. These examples include Apps that teach us how to play music, how to cook or dance, how to arrange our meetings and appointments, or assist us with learning and reading. Other Apps help us to find routes and places or support us by interacting with the real world and visualize it in a better way.

Today's smartphones and tablets come with processors and embedded sensors, including cameras, microphones and speakers, an accelerometer, a GPS, Wi-Fi, a digital compass, a gyroscope, a barometer, a light and temperature sensors. They also have some other important features such as touchscreen, lightweight and ease of use.

With the advances of mobile devices, new types of interfaces arise. For instance, AR allows to integrate digital information spatially registered to the physical environment of a user.

2.3 Augmented Reality

The concept Augmented Reality (AR) refers to any system that has the following characteristics [7]:

1. Combining real and virtual objects

Figure 2.2: Examples of Apps for different mobile platforms. Different Apps that teach us how to play music, how to cook or dance, how to arrange our meetings and appointments, or assist us with learning and reading.

2. Interacting in real time

3. Registering the data in 3D

Real time interaction between the real and the virtual worlds means overlaying texts or graphics on the surrounding world to give descriptions and information about the real objects. For example, Feiner et al. presented KARMA [10], an AR system that assists the users of laser printers by displaying maintenance instructions as shown in Figure 2.3,.

The following subsection covers several applications that explain the idea of AR and show how modern mobile devices can enrich such applications.

Applications

In 1996, Steven Feiner and his lab in Columbia University presented the Touring Machine [9], the first outdoor Mobile Augmented Reality System (MARS). The idea of this system was to overlay the names and websites of the different departments to the users walking around the campus. The prototype consisted of the following units:

Figure 2.3: KARMA. A virtual highlighted representation that shows the user how to remove the paper tray [10].

- Head-worn display with orientation tracker
- A backpack that has a computer, differential GPS, and radio for web access
- A hand-held touchpad interface and stylus

The appearance of the handheld devices solved the problem of carrying all these equipments and formed a more suitable platform for AR applications. Reitmayr and Drummond presented Going out [21], an example that shows an AR application on a handheld device Figure 2.4. It allows the users to interact with information overlaid to the surrounding reality.

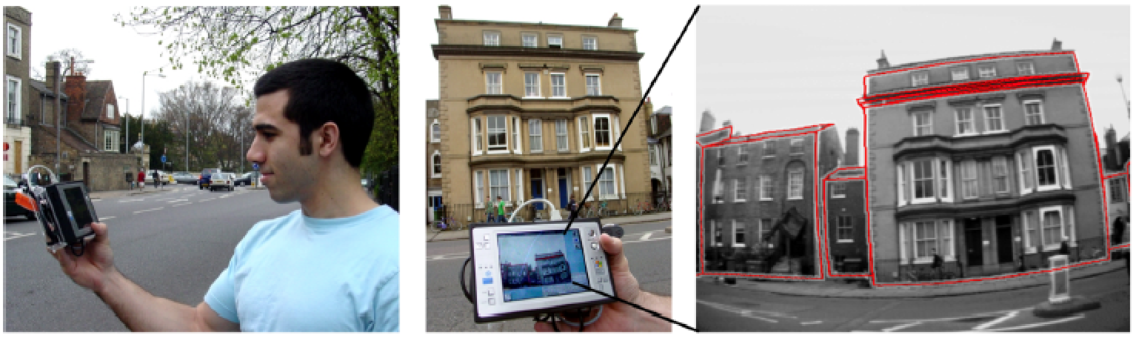

Figure 2.4: (Left) A user operating a hand-held augmented reality unit tracked in an urban environment. (Middle) Live shot showing the unit tracking a building. (Right) Screenshot from a pose close to the left images with overlaid building outline [21].

Today, smartphones and tablets with their powerful features and integrated sensors made the development of AR applications more effective and applicable. Smartphones and tabletsare relatively small in size and could be carried around easily, and are as well equipped with sensors that are employed by most of the AR applications, enabling the new approaches of AR applications to be installed easily and applied widely.

Wikitude[4] is one of the most well-known AR applications. It is a location based AR application that guides the travelers by superimposing information about places on a live stream camera view as shown in Figure 2.5. The problem with such approaches is that the streaming of overlaid information depends on the GPS and magnetometer, and as shown later in this work, the sensors in smartphones and tablets are inaccurate, causing some problems with the registration of information.

Figure 2.5: A user is using Wikitude app on a smart phone to get information about the surroundings [4].

Such approaches could be useful in utility management, through using an AR application that locates the device current position to import pre-stored data of underground pipes and overlay this data on the surrounding world to help them visualizing what is located underground.

2.4 Geographic information system

Many government tasks require geographic information, as almost all information used by governments are usually geographically referenced. Property registration and evaluation, planning and economic development, location of soil types or animals, infrastructure and transportation management, natural resource management and public safety are just few of many examples of such tasks. A geographic information system (GIS) allows us to examine and visualize the relationships between all these applications.

2.4.1 Definition of GIS

Geographical Information System (GIS) is a computer system that gives the users the ability to collect, store, analyze, manipulate, retrieve and display spatial data. [19]

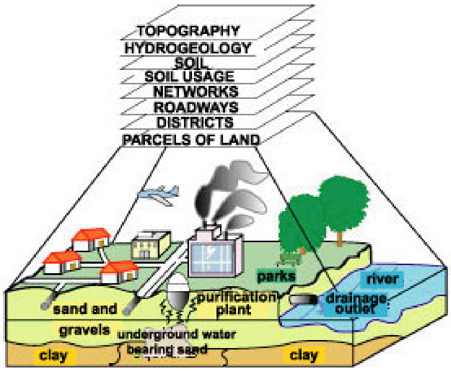

Figure 2.6: GIS integrates different kinds of spatial information layers [17].

Spatial data is information about natural features (e.g, mountains, hills, rivers, etc) or human created features (roads, cities, census data, etc). To make the operations provided by GIS faster and easier, GIS divides and stores all information about these features in separated categories or layers. Figure 2.6 shows different kinds of spatial information layers that are integrated by GIS.

GIS is a powerful system with powerful features, it is a heavy computer system that needs fast processor with big memory and other high quality computer resources, therefore, GIS is typically used on desktop computers. GIS users work generally with a representation of the real world such as maps and satellite images. Mobile GIS, on the other hand, adds the mobility feature to the system. With such important feature GIS users are able to carry the different GIS features with them while working in the field and examining the surrounding area.

2.4.2 Mobile GIS

Mobile GIS is an integrated technological framework that allows GIS users to benefit from mobile device features by having the ability to access geospatial data and location-based services in the field [11]. With the advancement of smartphones and tablets and their powerful features such as GPS, internet and wireless technologies, mobile GIS will be more powerful and applicable.

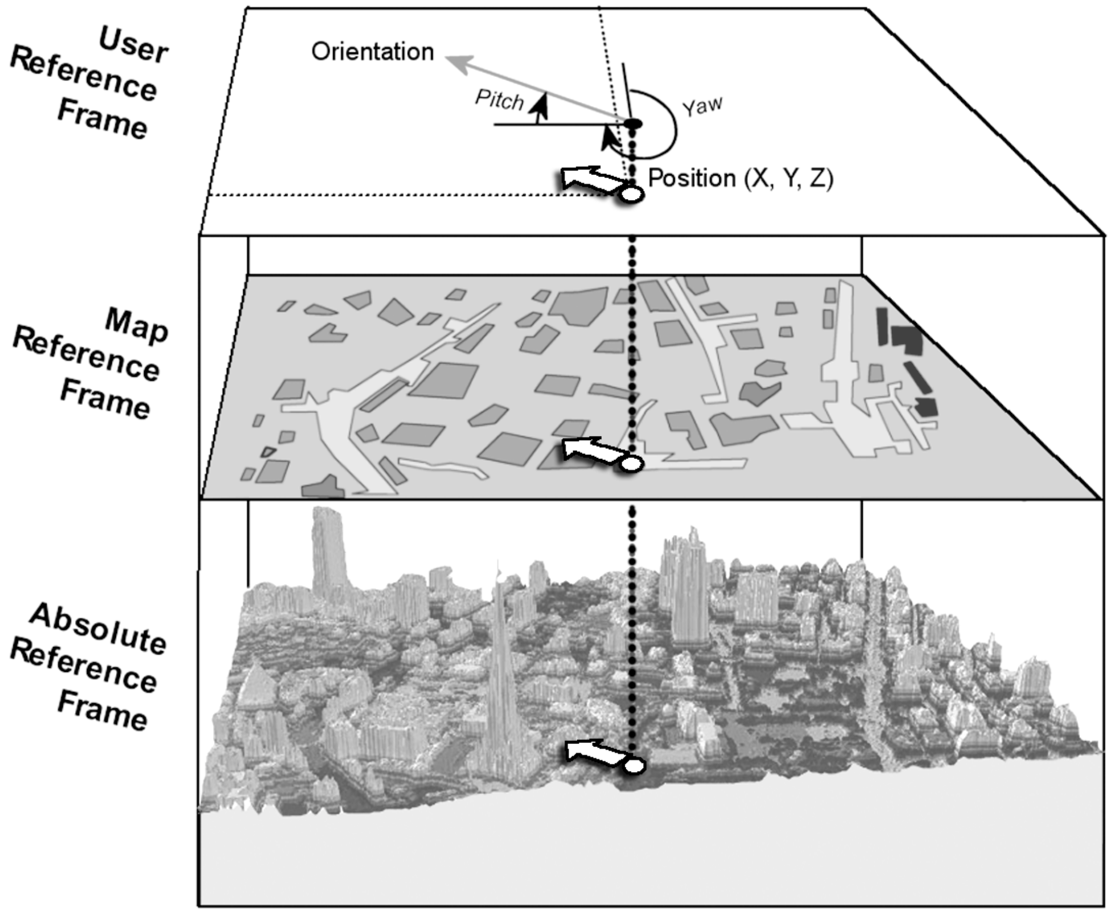

Figure 2.7: A description of the relationship between the real world, the map and the user position and orientation [11].

Mobile GIS is a new growing field of technology that concentrates on providing the user with location information via mobile devices based on specific position, as shown in Figure Figure 2.7. Such a new technology brings, together with its benefit, big challenges. While they have integrated features that make them powerful, they do have some disadvantages, which limit the full implementation of GIS on them, compared to desktop computers. .

Figure 2.8 shows an example of a mobile GIS application. 3D-GIS in the Cloud application gives its users the ability to navigate the city and its underground in 3D. They could gather and report location information about city buildings and infrastructures, and perform 3D GIS analysis.

The application helps organizations to reach the required GIS data in an efficient way and allows them to make accurate business decisions in the field and office as well.

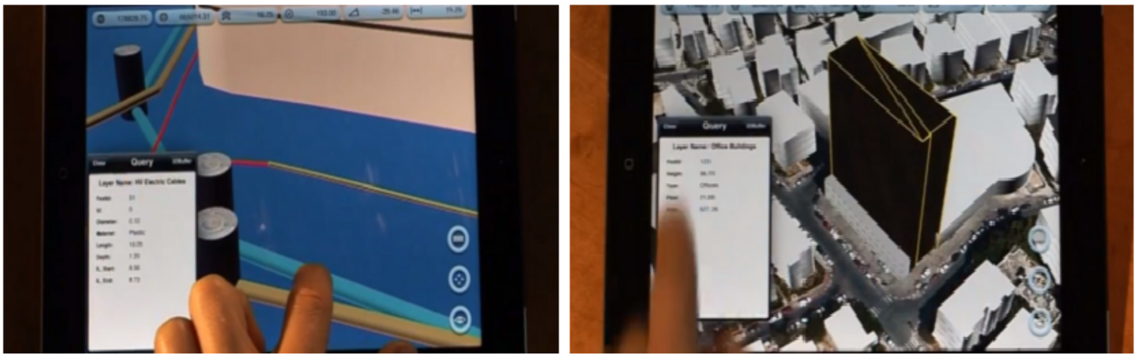

Figure 2.8: A screenshot of the 3D-GIS in the Cloud application, the user interacting with the application to have information about the different objects [23].

For organizations that wish to keep their on-site workers updated with the current GIS data, the 3D-GIS in the Cloud application is a great tool to manage such connections in an efficient way and to reduce potential disinformation mistakes.

The 3D-GIS in the Cloud application, like other mobile GIS applications, lacks the interaction with the real world. Such interaction adds more reality to the application and offers its users a reliable and accurate interface. On the other hand, the concept Augmented Reality GIS enhances mobile GIS applications by combining the real and virtual objects, and improves the user interface with a real time interaction of 3D data registration.

2.5 Augmented Reality GIS

The combination of Augmented Reality and mobile GIS provides GIS users with a new form of visualizing GIS data and a new way of interaction with the surrounding.

In order to get a better understanding of the concept behind AR GIS, one could take a deeper look on the following applications that were developed in this field.

1. **ARVino**

   ARVino is an outdoor augmented reality system. It was developed to help the viticulturists viewing information about the parameters that affect their yields and the quality of grapes[15]. Figure 2.9 shows a typical AR view with overlaid yield data.

   Figure 2.10 depicts the ARVino platform. The platform consists of a tripod that holds a Dell Inspiron 8100 laptop , a firewire video camera, a Garmin

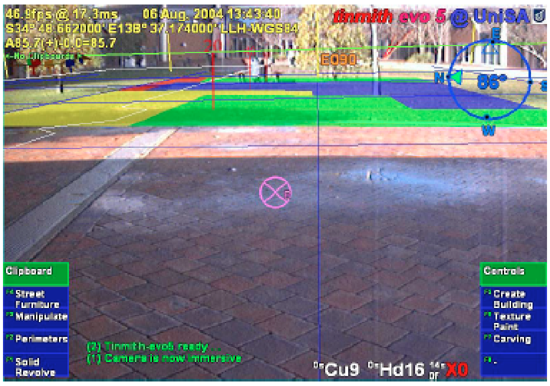

Figure 2.9: Typical AR view with overlaid yield data [15].

Figure 2.10: ARVino hardware with laptop, tripod, and umbrella [15].

12XL 12-channel hand-held GPS unit, and a TCM2 magnetic orientation sensor [15].

2. **Tinmith**

The Tinmith AR system [18] is a mobile outdoor augmented reality modeling system. It gives the user the ability to move freely in an outdoor environment. The user controls the system using specialized pinch gloves and a user interface. The Tinmith users can use this system to capture the 3D geometry of any outdoor structures, such as buildings, cars, trees, or other objects such as roads, grass and fences, as shown in Figure 2.11. Moreover, users are able to create 3D spatial information in real time.

Figure 2.11: Mobile Outdoor Augmented Reality Modeling Demonstration. The picture to the left shows Modeling cursors with pinch gloves, in the middle is a front view of the backpack and the head-worn display, and to the right is a rear view [18].

3. Smart Vidente

Smart Vidente is an outdoor mobile computing device that presents an interactive 3D real-time visualization model of underground GIS data. The Smart Vidente prototype incorporates visualization techniques and features, it does not only use networked RTK for positioning, but also applies methods like vision-based tracking to achieve better registration accuracy [22]. Figure 2.12 shows a registration of underground pipes.

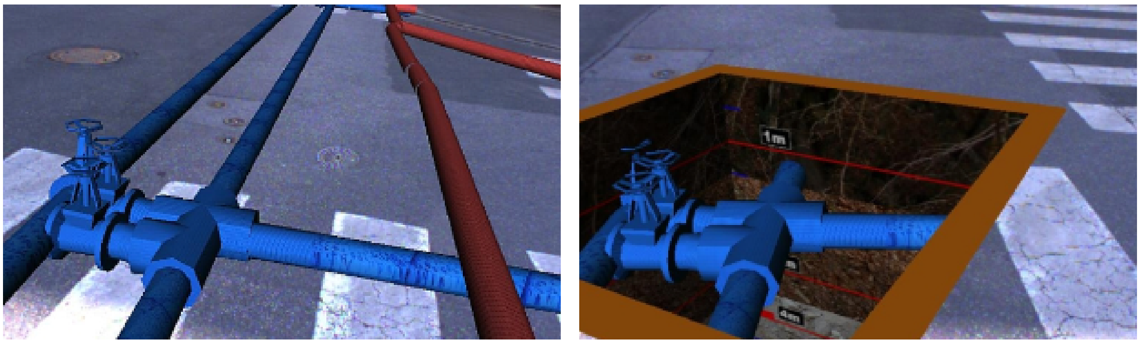

Figure 2.12: Smart Vidente, visualization of underground pipes. The picture to the left shows the underground pipes overlay over the ground. The picture to the right shows the underground pipes with their estimated depth [1].

The Smart Vidente platform is built around a tablet PC with a 1.6GHz Pentium CPU and a viewable touch screen ready to be used in outdoor conditions. It is supported with many external sensors, including a camera that provides a live stream of the real world, a sensor containing gyroscopes, accelerometers and magnetometers, a L1L2 RTK receiver for positional accuracies within the centimeter range.

2.6 Summary

Our goal in this work is to benefit from all these technologies by combining them in an effective way. As mentioned above, the concept of Augmented Reality was added to the mobile GIS applications to enhance them and improve the user interface reality.

However, the Augmented Reality GIS applications that we presented earlier have several disadvantages that we had to take into consideration as we developed our approach. In our approach we propose an AR application for visualizing GIS data on a tablet device which has powerful features that could overcome the other AR GIS applications disadvantages.

For example, the integration between the tablet sensors guarantees compatibility in the data flow between the different resources and, at the same time, eliminates the efforts that are spent on developing an appropriate interface between the different data sources. Furthermore, the lightness feature makes the application attractive and easy to use, as it allows the user to move freely in the field without the need to carry additional equipments. On the other hand, our approach was developed without any need for external sensors, such as an external GPS receiver. This was achieved through developing a method for correcting the alignment offsets with an interactive tool.

This work aims to gathering all the distinctive features from these technologies in one application and building on the existing Augmented Reality GIS applications and enhancing them through using the power of smartphones and tablets.

Chapter 3

Approach

Contents

Based on the previous chapter discussions, our approach is to build on the existing Augmented Reality GIS applications and enhance them through using the power of smartphones and tablets. Additionally, we attempt to solve the Smartphones or Tablets sensors inaccuracy problem by developing a tool that gives the user the ability to correct the GIS data rendering misalignments caused by this problem.

3.1 Introduction

In this approach we are implementing an Augmented Reality GIS application that uses the advantages of smartphones and tablets and benefits from their powerful features. However, the data that smartphones and tablets sensors provide is not accurate enough for our application, therefor our application will be equipped with a tool to correct the data inaccuracy.

The implementation of AR GIS applications requires transforming data between different coordinate systems. Data transformation combines different technologies, which were discussed in the previous chapter, guaranteeing a correct flow of data between the different coordinate systems. Normally, mobile device uses camera to capture real world images, accelerometer and compass to determine orientation and GPS for determining the location. Based on this data and data transformation, virtual objects are registered on the mobile devices.

This chapter discusses our approach of implementing an AR GIS application. Section 3.2 gives a general idea about the Augmented Reality visualizations and how we could use such kind of visualizations in our work. This is followed by Section 3.3, which explains data registration and how it is affected by sensors inaccuracy. We also present the coordinate systems that we use and discuss why we need different coordinate systems and how transformations between them are carried out. Finally, we will give a general overview about the tools used to correct the GIS data rendering misalignments caused by sensors inaccuracy.

3.2 Augmented Reality visualization

Augmented Reality applications enhance the reality by superimposing information about the real objects. This information is mixed with real objects, running in real time and registered with real world structures [7]. This kind of visualization is a powerful approach to interact with the real world by exploring the real objects along with overlaid information which describe and explain them.

AR visualizations might as well get information from the real world to be able to generate a virtual model and overlay it correctly. For instance, the AR application in this work uses the device GPS location and orientation to visualize underground pipes on the mobile device screen. In simple terms, a camera image of an urban scene is overlaid with virtual pipes that are located under the ground. Such visualization will help utility workers to know the exact location of underground pipes without the need to dig.

3.3 Registration

Accuracy is considered a key challenge for such AR applications. As mentioned later, the information that mobile devices receive from their sensors is not accurate and results in renderings where the pipes are overlaid with an offset and are not exactly at the correct position.

To overcome this, the user has to interact with the scene and fix the position of the rendered pipes. Yet remains the question, how the user could know the exact position of the underground pipes. The solution is to render additional contents beside the pipes, contents that the user can match with objects from the real picture. For instance, we can render a digital representation of an existing house in the user's physical environment. The user can use this information for compensating the registration offset by manually realigning the virtual house in such a way that fits the actual physical house.

3.3.1 Sensors

As mentioned before, the mobile device sensors are used to collect the information from the real world and store it to be processed at the real time or at a later time. The mobile device sensors that are used in this work are discussed in details in this section.

3.3.1.1 Camera

With the camera sensor we get information from the real world as still images or live stream videos. Images and videos provide us with important information about the real world and form the gate for the interaction process between real and virtual worlds. For example, in this work there are objects in the real world image that are used as references to correct the position of the virtual world model.

3.3.1.2 GPS Capability

Global Positioning System (GPS) is a satellite navigation system used to determine the location, speed, and direction of mobile objects on earth. The determination of the mobile devices current location accurately is one of the most important issue in this thesis and in many other AR applications.

Mobile devices get information about locations by using one of the following methods [14] :

- Using a built in chip called Assisted GPS (A-GPS). This chip receives the GPS signal directly from GPS satellites.
- Get the information from any Wi-Fi hotspot existing in the area.
- Get the information from mobile networks

Mobile devices may use these methods separately or combine them to speed up the positioning.

3.3.1.3 Magnetometer

Magnetometer is a digital compass that measures the magnetic field strength and direction. To measure the direction, the magnetometer and accelerometer are combined. This is called the heading of the device. As a result, the mobile device always knows the direction of North so it can rotate the digital objects according to the device physical orientation. However there are some problems, such as device exposition to a magnetic field, which might interfere with the process causing false data reading [6].

3.3.1.4 Accelerometer and Gyroscope

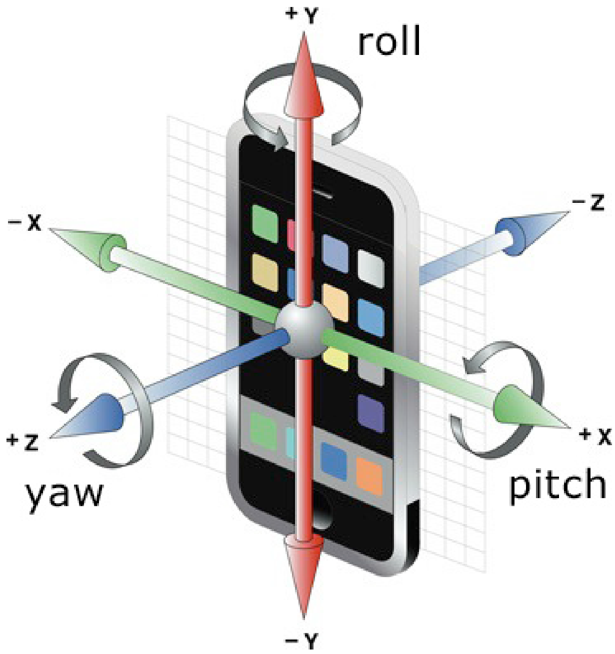

Figure 3.1: Accelerometer and Gyroscope measurement Axes [20].

Accelerometer is used to measure acceleration of movement over the 3-axes. It could be used to sense the motion of the device. On the other hand the gyroscope, or gyro, measures the angular rotational velocity over the 3-axes, Figure 3.1. In practice, accelerometer is able to measure the device directional movement accurately if the acceleration is zero. In order to achieve this during the movement, the gyro must provide the needed information [6] Figure 3.2.

The accurate detection of the device orientation is very important in Augmented Reality applications, without it, it is not possible to determine the device condition in re-

Figure 3.2: Combining the Accelerometer's data with the data from Gyroscope to get the Device Motion.

spect to the real world and therefore can not combine the virtual scene with the real world.

The registration of geospatial data on mobile devices requires the position of the device and its orientation, so that the virtual scene can be combined with the real world as illustrated in Figure 3.3. It is important to transform the virtual data according to the camera coordinate system of the mobile device in reference to the world coordinate system.

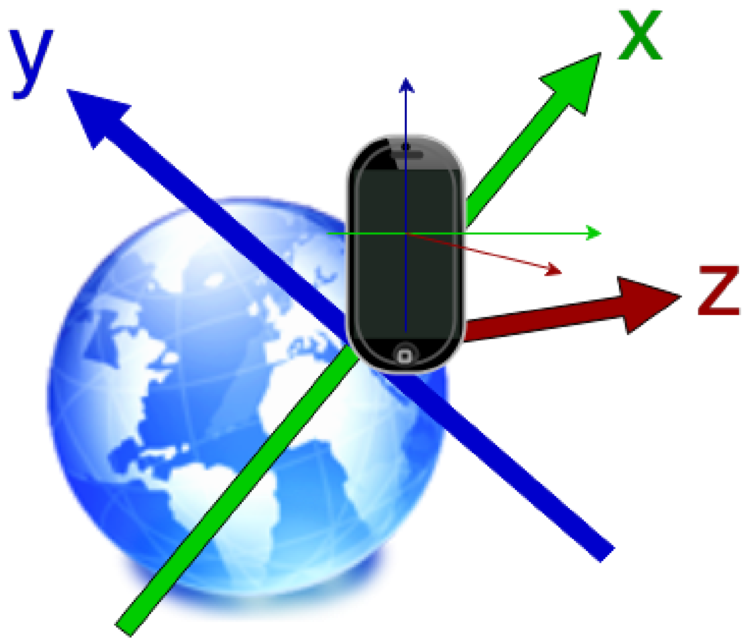

Figure 3.3: The position and orientation of the the device in respect to the world [20].

3.3.2 Coordinate Systems and transformations

In order to discuss the transformations between coordinate systems we first need to understand the coordinate systems that we use. As mentioned later in this section, we are using three different coordinate systems, Latitude-Longitude, Earth-Centered Earth-Fixed (ECEF), and East North Up (ENU) coordinate systems.

To compute the location of objects in respect to earth, we use two concepts, a coordinate system and a datum. To be able to use the GPS in an effective way, It is important to understand both of these concepts.

A coordinate system gives us the ability to measure the location of objects in space. This measurement depends on which coordinate system is used.

There are two major types of global coordinate systems that are used to determine locations on the surface of earth; angular coordinate system and rectangular coordinate system [16]. In this thesis we are working with an angular coordinate system, known as the Latitude-Longitude coordinate systems. This system represents the objects as three-dimensional objects, like a globe. A rectangular coordinate system, like UTM (Universal Transverse Mercator), represents the objects as 2-dimensional objects, like a map [16].

3.3.2.1 The Latitude-Longitude Coordinate System

Latitude-longitude system is represented by three basic formats

- **degrees/decimal/degrees.**
- **degrees/minutes/seconds**
- **degrees/minutes/decimal minutes**

Latitude begins at the Equator, which is the 0 Latitude line. From north to south, It measures the horizontal lines on the globe. The distance between the lines is 15 degrees, which is equal to 1,035 miles. Longitude begins at the Prime Meridian in England. This is the 0 Longitude line. From east to west, it measures the distance between the vertical lines. The distance between the lines is also 15 degrees but it becomes smaller as the lines approach the poles [16]. Figure 3.4 describes this coordinate system and how to represent any location on earth .

The Latitude-Longitude coordinate system is being used in many modern applications. Its importance comes from the fact that it is considered a standard coordinate system in all

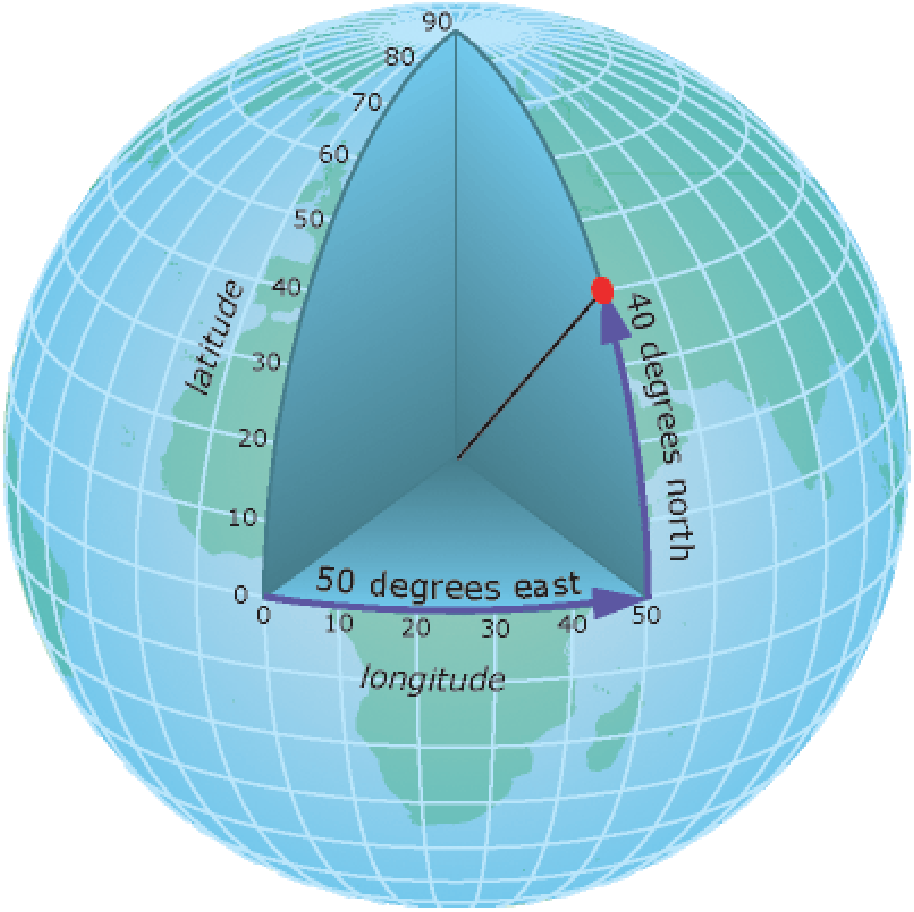

Figure 3.4: Determining any point on earth using the Latitude-Longitude Coordinate System [3].

GIS data resources and web mapping [16]. For our work in this thesis, both geospatial data and GPS position are in fact measured using the Latitude-Longitude coordinate system.

3.3.2.2 Datums

Datums are standard coordinate frames that specify how the coordinate system is related to the earth: the origin's location, the scale, as well as the coordinate axes orientation [16]. There are different types of datums, however, throughout our work we only used the WGS84 standard.

Latitude-Longitude coordinate system is not suitable for rendering data as it represents a global spherical coordinate system and what we need in our approach is a flat horizontal coordinate system. For this reason we have to transform the data to ENU coordinates. As an intermediate step between these two coordinate systems the data has to be transformed to the ECEF coordinate system first. As described in Figure 3.5, the center of the ECEF coordinate system is located at the center of earth and its Z-axis points to the North. On the other hand we transform the data to the ENU coordinate system because it is a plane tangent to the surface of earth and its origin could be any point on the earth's surface, and as shown in Figure 3.5, the Up vector is, unlike the ECEF coordinate system, normal to the surface of earth.

Another important reason for transforming the data to the ENU coordinate system is that the ECEF coordinates are considered too big to be used directly in rendering the scene to the screen using a graphic library like OpenGL ES [8]. That what makes the ENU system more suitable for rendering the data on mobile devices.

3.3.2.3 ECEF Coordinate System

The Earth-Centered, Earth-Fixed (ECEF) coordinate system is a Cartesian coordinate system used to represent any point on earth as a X, Y and Z coordinate. The name "Earth-Centered" means that the origin point (0,0,0) of this coordinate system is located at the center of earth. On the other hand. "Earth-Fixed" means that the X,Y and Z axes have a fixed position and rotation in respect to the earth; they rotate with the earth rotation. The Z-axis points to North, and the XY-axis represent the equatorial plane [24].

However, since the center of the earth (the origin of the ECEF coordinate system) is located far away from any point on the surface, it would be inappropriate to apply such a system for a small area with a mobile device located far from the origin. In this case the ENU coordinate system is considered more intuitive and practical than the ECEF coordinate system.

3.3.2.4 ENU Coordinate System

East North Up (ENU) coordinate system, also known as local coordinate system, is a basic flat Cartesian coordinate system that uses linear X, Y and Z coordinates to locate objects with respect to the origin of the coordinate system (the origin reference point in our case is the GPS position of the mobile device), X axis corresponds to East, Y axis corresponds to North, and Z axis corresponds to the Up vector [24]. This coordinate system is effective for small areas (less than 5 km), where the curvature of earth is not mentionable. Because the local coordinate system consider flat surfaces, it is not right to be used over large surfaces.

ENU is using the device GPS location as a reference point for the geospatial data

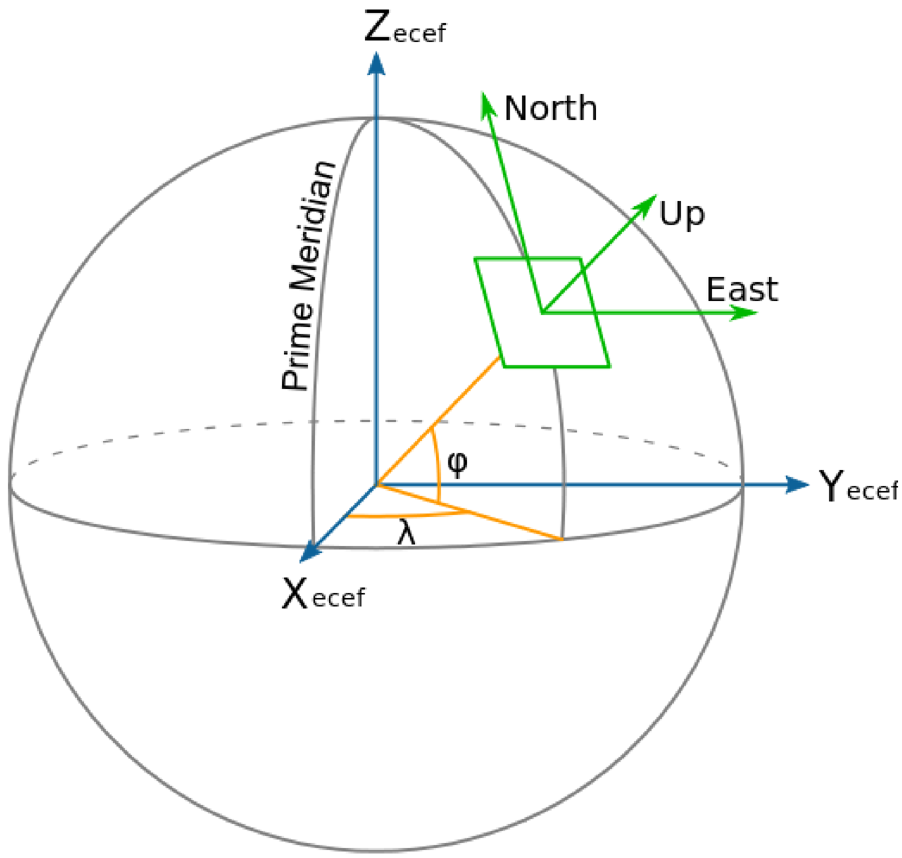

Figure 3.5: The relationship between the three different coordinate systems, The Latitude-Longitude Coordinate System, ECEF Coordinate System, and ENU Coordinate System [24].

surrounding that location. The name local cartesian coordinate system means that user is establishing a unique cartesian coordinate system for each location that has a GPS position as its origin and all the other geospatial data is being referenced by this origin.

However, it is a suitable coordinate system for small areas because it allows the use of Euclidean geometry which plays a very important role in data rendering on mobile devices. Figure 3.5 describes the relationship between the three different coordinate systems.

3.3.2.5 Data Transformation

Transformation guarantees the flow of data between the different coordinate systems in the desired way.

Figure 3.6 shows the necessary steps to achieve this transformation. All geospatial data must be converted from latitude longitude coordinate system into ENU local Cartesian

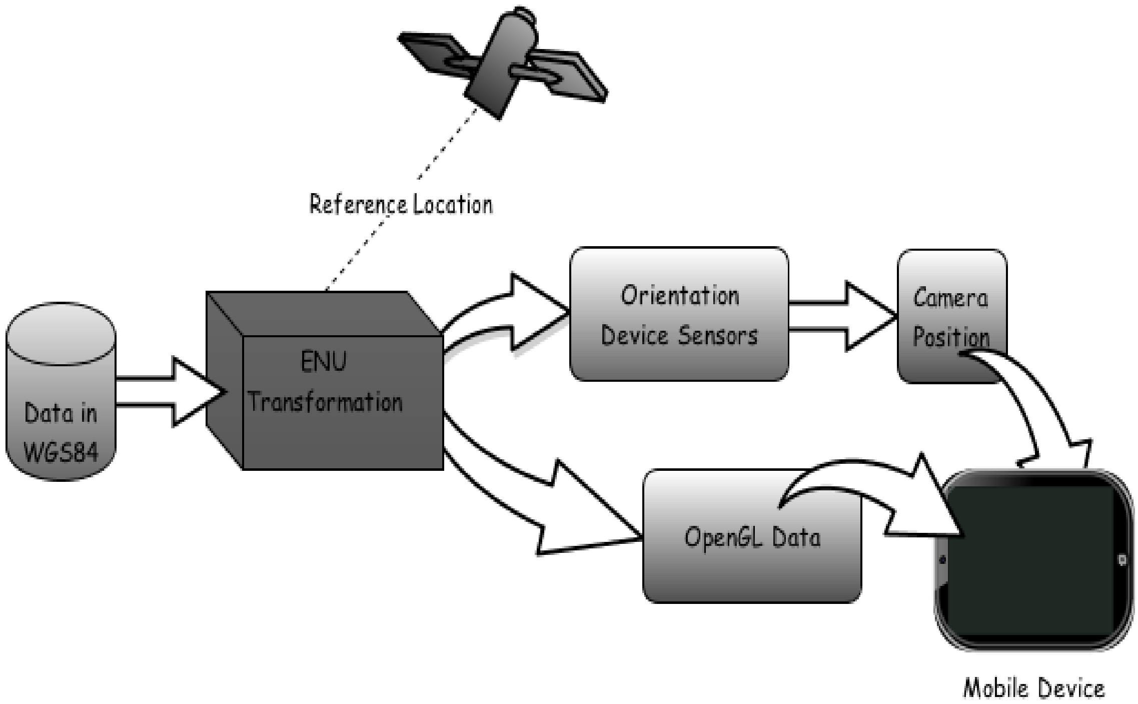

Figure 3.6: Process of registering the GIS data on mobile devices.

coordinate system.

To be able to apply the ENU system, first we have to have all geospatial data as well as GPS position in a 3D form.

The ECEF coordinate system is used as an intermediate step, meaning that we convert the latitude longitude degrees of the geospatial data and the GPS position to ECEF first, then to the local ENU system. Such conversion can be obtained using the following formulas [24]:

$$X = (N + h)\ cos\varphi\ cos\lambda$$
$$Y = (N + h)\ cos\varphi\ sin\lambda$$
$$Z = (N(1 - e^2) + h)\ sin\varphi$$

where φ and λ are latitude and longitude respectively. h is the height above ellipsoid and N is the radius of curvature in meters, it is defined as:

$$N = \frac{a}{\sqrt{1-e^2 sin^2\varphi}}$$

A reference ellipsoid can be represented by a number of parameters that determine its shape, these parameters include:

$a = 6378137$ *is the WGS84 semi-major axis constant.*
$e = 6.69437999014x10^{-3}$ *is the WGS84 eccentricity.*

After converting all data successfully to ECEF the local coordinate system for the given position could be easily created. As mentioned before, the GPS position of the mobile device acts as a reference point for the other geospatial data, meaning that the GPS position considered the (0,0,0) point of the ENU system. The following formulas describe the process [24].

$D_X = X_{Data}$ - X_{GPS}
$D_Y = Y_{Data}$ - Y_{GPS}
$D_Z = Z_{Data}$ - Z_{GPS}

As mentioned earlier, we convert the geospatial data and the device GPS position to ECEF first, then to the local ENU system. Therefore, X_{Data}, Y_{Data}, and Z_{Data} represent the geospatial data in ECEF coordinate system, and the X_{GPS}, Y_{GPS}, and Z_{GPS} represent the device GPS position in ECEF.

$E = -sin\lambda \times D_X + cos\lambda \times D_Y$
$N = -sin\varphi \times cos\lambda \times D_X$ - $sin\lambda \times sin\lambda \times D_Y + cos\varphi \times D_Z$
$U = cos\varphi \times cos\lambda \times D_X + cos\varphi \times sin\lambda \times D_Y + sin\varphi \times D_Z$

3.3.3 Interactive Method

The accurate rendering of GIS data is very important for this work and for other AR GIS applications. The information received by mobile devices through their sensors is not accurate and will result in renderings where the GIS data is overlaid with an offset and are not exactly at the correct position. The aim of this work is to provide an interactive user interface for correcting this inaccuracy.
Figure 3.7 shows the main phases of our approach.

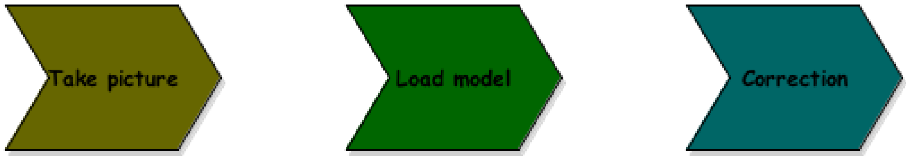

Figure 3.7: Main phases of this approach.

In the **Take picture** phase, we receive the data from sensors. The device takes the real world picture and starts updating the GPS and rotation sensors.

The **Load model** phase takes the GIS data, applies the predefined transformation methods, and uses the data from the rotation sensors to render the virtual model on the real world picture.

On the other hand, the **Correction** phase provides the correction tools that give the user the ability to correct the inaccuracy by shifting or rotating the model around the X,Y and Z - Axes until the virtual objects match the objects in the picture.

Because of the mobile device sensors inaccuracy the misalignment between the real and the virtual objets averages between 5 and 25 meters. Our correction tools were designed to allow the user to interact with the virtual and real objects in the scene, therefore a still picture of the real world is required so that the user can match the objects correctly. In case of real-time video, applying our correction tools would be very hard specially if the misalignment is big because the user will not be able to concentrate during the fixing of misalignments between the real and the virtual objets when the real objects in the video stream are not still.

In the next chapter the implementation of these phases is described in more details.

3.4 Summary

Implementing an Augmented Reality GIS application on Smartphones or Tablets platform requires data accuracy. However, the data that smartphones and tablets sensors provide is not accurate, this is why our application will be equipped with a tool to correct the data inaccuracy.

However, the data has to be transformed between the different coordinate system to be suitable for rendering, the reason is that the Latitude-Longitude and the ECEF coordinate systems are not suitable for rendering data, thus we have to transform it to ENU coordinates.

The correction tools give the user the ability to correct the inaccuracy by shifting or rotating the model around the X,Y and Z - Axes until the virtual objects match the objects in the picture.

Chapter 4

Implementation

Contents

This chapter introduces a prototype that implements the theoretical methods discussed earlier and presents a practical application of rendering geospatial data on mobile devices. Our prototype is an Augmented Reality GIS application that uses the advantages of the iPad and benefits from its powerful features.

4.1 Introduction

In the previous chapter, we discussed the different methods and transformations necessary to achieve the rendering of geospatial data. In this chapter we implement data rendering and take a deeper look on how the geospatial transformations are applied in practice and how the inaccuracy of sensors leads to inaccurate data rendering. However, the purpose of this work is to correct this inaccuracy. Therefore, we will present a set of tools that give the user the ability to correct the inaccuracy of the geospatial data.

We start by introducing an overview of the whole implementation, then the three main phases of the whole approach are discussed. As shown in Figure 4.1, the first phase, Geodata and Transformation, deals with all the requirements needed to have a solid coordinate system on which the other systems depend, in order to guarantee an accurate rendering of the virtual model. It also shows how to prepare the data, how to get the data, where and in which form the data is stored and how the data is sent to be rendered.

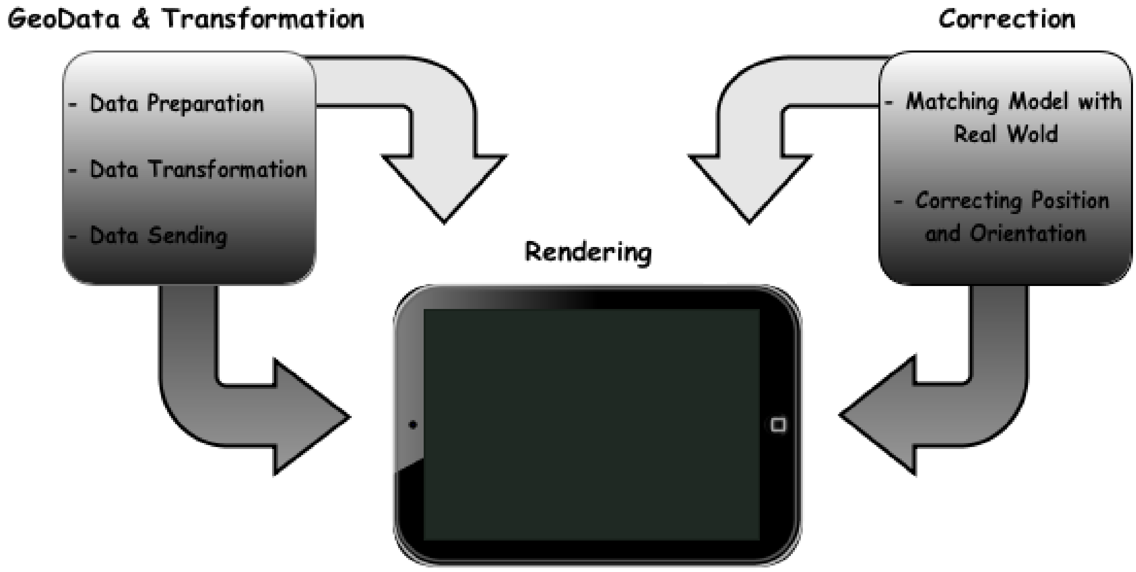

Figure 4.1: The three main phases of implementing the registration of GIS data on a mobile device and the correction tool.

Rendering phase describes the rendering of geospatial data using OpenGL ES graphics library, it explains the transformation matrices including the orientation matrices and translation matrix. The last phase is the Correction phase, in this section we will talk about the correction techniques and tools that fix the rendering problems caused by the sensors inaccuracy.

4.2 Overview

As mentioned above the goal of this work is to help the user visualizing the underground pipes in a specific location. To be able to do so, using this prototype, the user should just follow the three basic steps described in Figure 3.7.

The user starts by taking a picture for the area of interest using his iPad, the picture should contain some fixed visible objects such as houses, sidewalks or any object that is over the ground and its information is stored in GIS. These visible objects will be considered as references for the virtual model to be matched later. After that the user loads the virtual model on top of the taken picture. The model consists of transparent renderings of buildings outlines and renderings of underground pipes outlines. As mentioned before the inaccuracy that is caused from mobile device sensors leads to some inaccuracy in loading the model, in other words, the overlay will not match the objects in the picture. To fix this, the correction step gives the user the ability to correct the inaccuracy by shifting or rotating the model around the X,Y and Z - Axes until the virtual objects match the objects in the picture.

Figure 4.2: A screenshot of the prototype, two buttons on the top bar, **Take Picture** and **Load Model** are for taking the picture of the real world and loading the virtual model respectively. On the other hand the colored buttons are used for the correction method. And the red labels report the current latitude, longitude, and height.

Figure 4.2 shows a snapshot of the prototype. It shows the inaccuracy after the model was loaded, the tools that the user could use for correction and some other information which will be discussed later in this chapter.

4.3 Geodata and Transformation

This section handles the geospatial data that will be rendered on the iPad screen. This data has to go through several steps to be suitable for rendering. The following three subsections will explain these steps in details.

4.3.1 Data preparation

Firstly the geospatial data will be processed and translated to latitude-longitude coordinates using special GIS utility tools called Feature Manipulation Engine (FME) [12]. These tools help transforming and translating the geospatial data between many data formats. During the FME process we assign two values, the height of the building and the depth of the pipes. These two values are not supported in the GIS from which the data sample for this prototype was taken. Afterwards, the geospatial data will be stored in a Wavefront .obj file. One file holds the geospatial for the underground pipes and another file for the other visible objects (e.g, houses or sidewalks).

Figure 4.3 shows an example of a Wavefront .obj file that contains the geospatial data of a building in Latitude-Longitude coordinates. Each line starting with the letter ' v ' represents a Vertex, which has three coordinates, latitude, longitude and hight. The line starting with the letter ' f ' represents a face. A face tells the rendering method which Vertices should be connected together with a line.

4.3.2 Data transformation

The transformation process starts by parsing the file, firstly to count the number of vertices and faces, so that the rendering method knows how many vertices and faces are there and reserve arrays for them, and secondly to store the coordinates of the vertex in variables.

After the file has been parsed and the coordinates are in variables, the data will be transformed to the local ENU coordinate system. As explained in Chapter 3, the ENU local coordinate system will be applied with the device's current GPS location as its point of origin. Recall the transformation formulas described in Chapter 3. Figure 4.4 represents two methods that handle the transformations form latitude-longitude coordinate system to the local coordinate system.

The first method, latLonToEcef, is applied twice. First time it is used to transform the geospatial data from latitude-longitude coordinates to ECEF coordinates. In the method's parameters list, the method takes the first three parameters, lat, lon and alt and returns

```
# Created with FME Version: FME(R) 2012 20120115 - Build 12212 - WIN32
# COORDINATE_SYSTEM:  OGC_DEF GEOGCS["WGS84 datum, Latitude-Longitude; Deg
    GEM 10C",6378137,298.257223563,AUTHORITY["EPSG","7030"]],AUTHORITY["EP
    0.0174532925199433],AUTHORITY["EPSG","4326"]]
# Number of Geometry Coordinates  : 12
# Number of Texture  Coordinates  : 0
# Number of Normal   Coordinates  : 0
v 15.442448 47.062154 10.000000
v 15.442618 47.061804 0.000000
v 15.442618 47.061804 10.000000
v 15.442448 47.062154 0.000000
v 15.442226 47.061698 0.000000
v 15.442226 47.061698 10.000000
v 15.442597 47.062246 10.000000
v 15.442795 47.061870 0.000000
v 15.442795 47.061870 10.000000
v 15.442597 47.062246 0.000000
v 15.443337 47.062029 0.000000
v 15.443337 47.062029 10.000000

# Number of Elements in set       : 8
f 1 2 3
f 1 4 2
f 2 5 6
f 3 2 6
f 7 8 9
f 7 10 8
f 8 11 12
f 9 8 12
# Total Number of Elements in file: 8
# EOF
```

Figure 4.3: A Wavefront .obj file contains the faces and the vertices of the geospatial data in latitude-longitude coordinates with the corresponding height.

the equivalent x, y and z in ECEF. The alt represents the height of the object that is being rendered (e.g, a house). The height of objects must be added manually during the FME process, because the GIS from which we took our geospatial data is not supported with objects height. Recall the data in the .obj file in Figure 4.3, we will see that almost every two consecutive vertices has the same latitude, longitude values but with different height value, either 0 or 10, and this is actually necessary to be able to identify the objects as 3 dimensional objects and not just 2 dimensional plane.

The latLonToEcef method will be used for the second time to transform the GPS data, that represents the current mobile device location, into ECEF. The method will also take the lat, lon and alt values as input and returns the ECEF x, y and z values. This time the alt is not added manually, but rather measured by the mobile device sensor, and represents

```
// Coverts lat, lon to ECEF
void latLonToEcef(double lat, double lon, double alt, double *x, double *y,
    double *z)
{
  double clat = cos(lat * DEGREES_TO_RADIANS);
  double slat = sin(lat * DEGREES_TO_RADIANS);
  double clon = cos(lon * DEGREES_TO_RADIANS);
  double slon = sin(lon * DEGREES_TO_RADIANS);

  double N = WGS84_A / sqrt(1.0 - WGS84_E * WGS84_E * slat * slat);

  *x = (N + alt) * clat * clon;
  *y = (N + alt) * clat * slon;
  *z = (N * (1.0 - WGS84_E * WGS84_E) + alt) * slat;

}


// Coverts ECEF to ENU coordinates centered at given lat, lon
void ecefToEnu(double lat, double lon, double x, double y, double z, double
    xr, double yr, double zr, double *e, double *n, double *u)
{
  double clat = cos(lat * DEGREES_TO_RADIANS);
  double slat = sin(lat * DEGREES_TO_RADIANS);
  double clon = cos(lon * DEGREES_TO_RADIANS);
  double slon = sin(lon * DEGREES_TO_RADIANS);
  double dx = x - xr;
  double dy = y - yr;
  double dz = z - zr;

  *e = -slon*dx  + clon*dy;
  *n = -slat*clon*dx - slat*slon*dy + clat*dz;
  *u = clat*clon*dx + clat*slon*dy + slat*dz;
}
```

Figure 4.4: The two methods, latLonToEcef and ecefToEnu, that transform geospatial data from latitude-longitude coordinate system into ECEF coordinate system and then to ENU coordinate system.

the height of the mobile device over the sea level.

The second method, ecefToEnu,is applied just once. It takes the first 8 parameters, including lat, lon, x, y, z, xr, yr and zr. The x, y and z parameters represent the GPS data in ECEF and the parameters xr, yr and zr represent the ECEF coordinates of the object's geospatial data. Afterwards the ecefToEnu method returns the e, n and u in the local coordinate system.

4.3.3 Sending the data to OpenGL ES

Now the geospatial data is no longer latitude-longitude vertices and it has been converted to ENU coordinates. The ENU is suitable to be passed to OpenGL ES and rendered on the screen.

Any 3D object has a shape, it is called geometry. The geometry of any objects is put together using a set of primitives, the primitives in OpenGL ES are point, line and triangle. Now these primitives are defined using an array of vertices.

The vertices, that are being passed to OpenGL ES, are parsed from two files, whereas the first file contains the vertices of the object (e.g, house) and this object's geometry would probably be rendered as a combination of triangles. The second file contains the vertices of the underground pipes which will be rendered as lines. Because these lines represent water or electricity pipes, it would be more realistic, in future work, to render them as cylindrical objects. During the parsing process the vertices are transformed to the ENU coordinates using the two methods explained in Figure 4.4.

4.4 Rendering of geospatial data

To be able to render objects data on the mobile screen using a graphics library like OpenGL ES, the objects have to be controlled by the Model, View, and Projection transformation matrices.

The Model matrix transforms the vertices from the object space into world space, and the View matrix transforms the vertices from world space to camera space. On the other hand the Projection matrix transforms the vertices from camera space to screen. The following formula describes the relationship between the different transformations:

$$MVP = Projection \text{ x } Model \text{ x } View$$

As mentioned before, the ENU coordinate system uses the current GPS location as its point of origin and at the same time as the camera eye of the scene. However, it is important in the Augmented Reality applications to determine the device position and orientation with respect to the real world to be able to render the geospatial data to their correct positions. So in this case our formula will be described as follows:

$$MVP = Projection \text{ x } R$$

As described in this formula we replaced the *Model x View* matrix with R matrix because the R matrix is the rotation matrix that holds the device motion. Its elements values are updated automatically from the device sensors. If the device changes its orientation the rotation matrix updates its elements accordingly so that the objects are rendered correctly. However, the position of objects was determined by the ENU transformation. Now the ENU transformation could be also applied as a matrix and this matrix will be multiplied by the *Projection x R* matrix, but from experience, applying the ENU transformation before sending the vertices to OpenGL ES provides better rendering.

In Figure 4.5 the rotation matrix, in iOS, is provided in a reference class called CMAttitude, which is a read-only property that returns the device *Attitude* [13].

```
//Sensors Rotation Matrix

    CMRotationMatrix mat = motion.attitude.rotationMatrix;
    GLfloat motionRotMat[] = {
        mat.m11, mat.m21, mat.m31, 0.0,
        mat.m12, mat.m22, mat.m32, 0.0,
        mat.m13, mat.m23, mat.m33, 0.0,
        0.0,     0.0,     0.0,     1.0
    };
```

Figure 4.5: Sensors Rotation Matrix.

Now that we have all the pieces together, the geospatial data could be rendered according to the device location on earth and according to the device condition with respect to the real world.

4.5 Correction

As explained before, the data that the mobile device gets from its sensors is not accurate, and this inaccuracy will affect the rendering process, so that the virtual objects will not be matching their corresponding physical objects in the picture.

In order to fix this inaccuracy we provide the user with a set of tools for manual

Figure 4.6: Screenshot of the prototype shows the correction tools, **Rotate X**, **Rotate Y**, **Rotate Z**, and **Scale or Translate**.

correction. As shown in Figure 4.6 there are 4 buttons on the left side of the iPad's screen, including **Rotate X**, **Rotate Y**, **Rotate Z** and **Scale or Translate**.

The moment the user touches the iPad's screen, the picture (real picture + virtual model) will be frozen, the device will stop updating the motion sensors, and the correction process will start.

To record the user touches on the device screen, we use the UIResponder reference class that is defined in the iOS Developer Library. UIResponder class provides three methods to respond to touch events. Figure 4.7 shows these methods. The touchesBegan and touchesEnded methods record the position of the user's finger when the touch begins and ends, and return the coordinates of these touches with respect to the screen coordinates system, which is measured in pixels. On the other hand, the touchesMoved method is used to record the distance of movement, starts when the touch begins and ends when the touch ends.

```
//Respond to the touch events, touchesBegan, touchesMoved, and touchesEnded

- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event;
```

Figure 4.7: Methods from iOS Developer Library that record the touch events.

As described in the following subsection, in order to rotate and translate the virtual model on the screen, the user has to apply touch gestures, like pinching, stretching, or turning. These gestures require two fingers to be touching the screen. Therefore, in order to control these multiple touches and movements, we use the two methods described in Figure 4.8.

The distanceBetweenPoints method returns the distance between two points. Using this method, we can measure the distance between two fingers and record the movements of fingers. If the distance is getting smaller that means the fingers are getting closer to each other and thus the user is applying the pinching gesture. On the other hand, if the distance is getting bigger that indicates that the user is applying the stretching gesture.

```
// Calculate the distance between two points
CGFloat distanceBetweenPoints (CGPoint first, CGPoint second) {

    CGFloat deltaX = second.x - first.x;
    CGFloat deltaY = second.y - first.y;
    return sqrt(deltaX*deltaX + deltaY*deltaY );

};

// Calculate the angle between two points
CGFloat angleBetweenPoints(CGPoint first, CGPoint second) {

    CGFloat rads = atan((second.y - first.y) / (first.x - second.x));
    return radiansToDegrees(rads);

}
```

Figure 4.8: Methods used to calculate the distance and angle between two points.

The second method returns the angle between two points, and is used to record the changing in the angle degrees between two fingers. When the user turns the fingers on the screen the angle could be measured and accordingly know if the angle is increasing or decreasing. If the angle is increasing that means the virtual model will be rotating towards one direction, and if the angle is decreasing the virtual model will be rotating towards the opposite direction.

After recording these gestures and calculating the changes in movements using the predefined methods, the results values will be used as elements in the rotations matrices (RotateX, RotateY, and RotateZ matrices) and the translation matrix (Translate on the x-axis, Translate on the y-axis, and Translate on the z-axis). These matrices, on the other hand, control the virtual model rotation and translation.

However, from experience we noticed that using these values which result from recording gestures changes directly in matrix transformations will cause a fast response of the virtual model according to the fingers gestures, and thus it would be hard to control the virtual model rotations and translations. This is because these values are pixels units, and are considered big values for matrix transformations. For this reason, before using these values in the rotation matrices we divide them by 3, and in the translation matrix the values are divided by 50.

The user has five tools, applied separately to correct the inaccuracy and match the virtual objects to the physical objects in the real picture. The tools are described in the following sections.

4.5.1 Orientation adjustment

With the help of the different touch gestures that are supported in iPad, the user turns two fingers on the screen to rotate the virtual objects around camera origin. Figure 4.9 shows the model with rotations around the different axes. One reason for choosing this gesture to control the rotations of the virtual model is that it is more stable and accurate than other kinds of events, such as turning the whole device for rotation. Moreover this gesture is commonly used for rotating objects in many mobile applications and many users are already familiar with it.

4.5.2 Scale or Translate

For translation, the user applies one finger to translate the model either in the x-axis or in the y-axis, in other words, shifts the model right-left or up-down. But for scaling, two fingers are used to pinch or stretch to translate the model in the Z axis, that means close

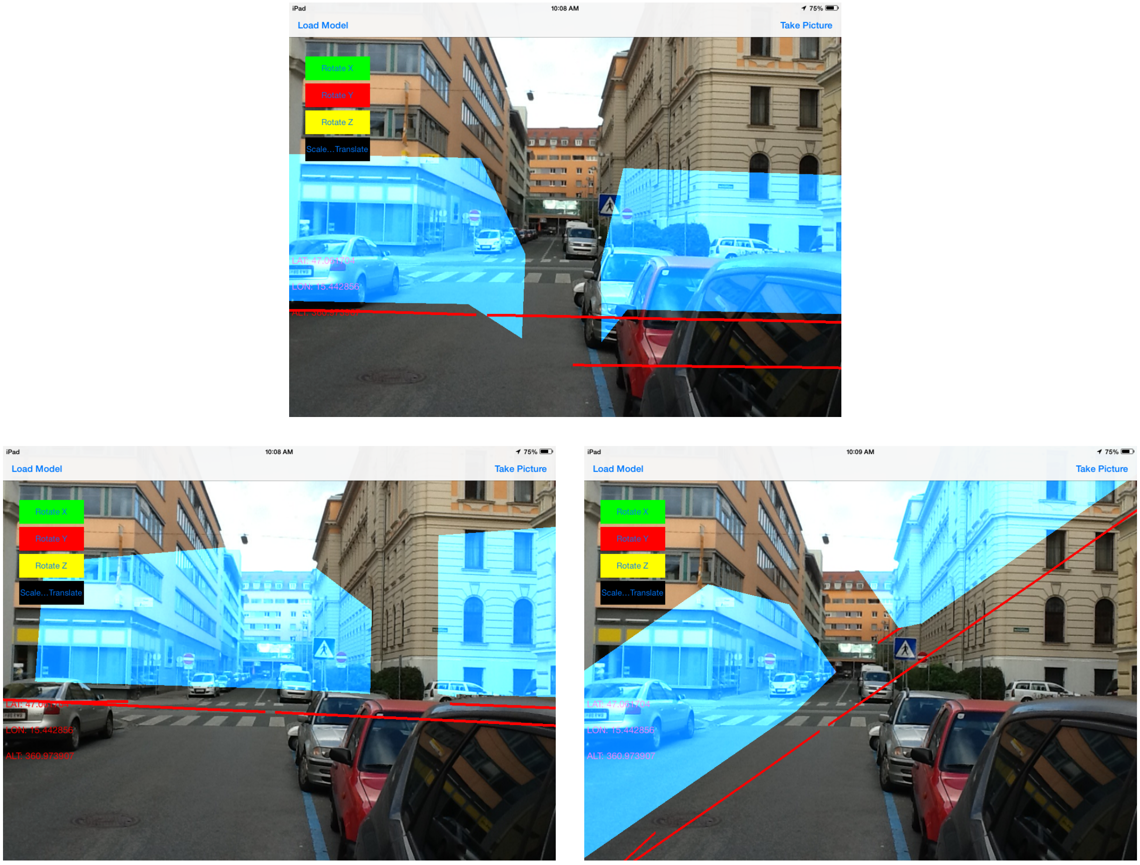

Figure 4.9: Screenshot of the prototype. Using **Rotate X**, **Rotate Y**, **Rotate Z** tools to rotate the model manually around the **x-axis**, **y-axis**, and **z-axis** respectively.

to the camera or away from the it. This is shown in Figure 4.10.

For the same reasons, we have decided to use these gestures for translating the model, they are accurate, stable and commonly used.

Figure 4.11 shows the model and the underground pipes after the user fixes the rendering.

In order to understand how the correction process works, we have to go back to the matrices formula.

$$MVP = Projection \text{ x } R$$

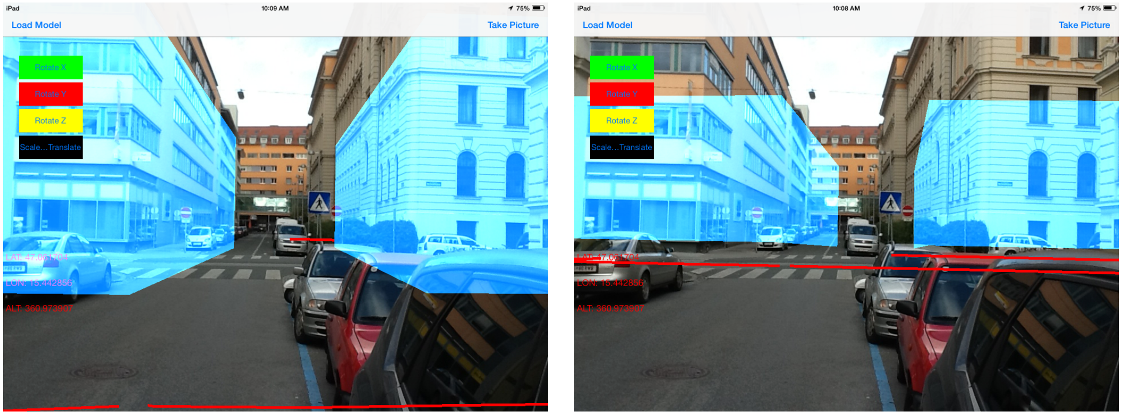

Figure 4.10: Screenshot of the prototype. Using **Scale or Translate** tools to zoom the model manually in (close to the camera) or out (away from the camera).

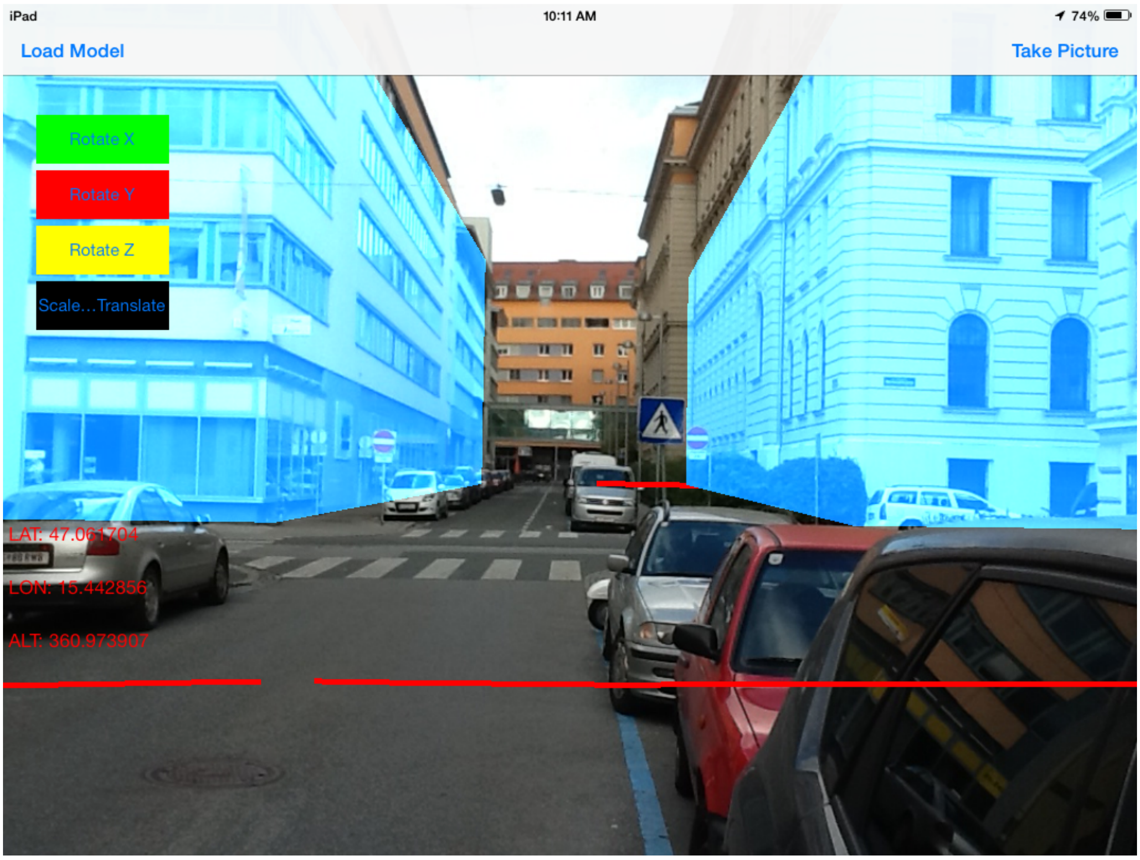

Figure 4.11: Screenshot of the prototype. The virtual model matches the real image, after the user corrected the inaccuracies.

As discussed earlier, this formula represents our rendering process.

When the user makes a number of rotations and translations which are considered as corrections to the object's rotation and the GPS position (the position of the object with respect to the camera), these corrections have to be stored in a matrix and this matrix will be multiplied with $Projection \text{ x } R$ every time the user makes a new correction. Therefore, teh rendering formula will be modified as follows:

$$MVP = Correction \text{ x } Projection \text{ x } R$$

In such way the corrections of the sensors inaccuracies will be saved.Thus remains the question how the *Correction* matrix is formed.

The *Correction* matrix is the accumulation of all translations and rotations in one matrix. We know that the rotation and translation operations are done using matrices, however, the elements of these matrices have their values from the touch movements (degrees of shifting) that the user makes with his fingers on the screen. In order to accumulate all these translations and rotations in the *Correction* matrix they have to be multiplied with one another and at the end multiplied with the *Correction*

The following formula describes our *Correction* matrix:

$$Correction = Correction \text{ x } RotateX \text{ x } RotateY \text{ x } RotateZ \text{ x } Translate$$

Accumulating all these matrices is done by multiplying them with each other and saving the result in the *Correction* matrix every time the user touches the screen.

4.6 Summary

This chapter discussed the implementation of a prototype for rendering the geospatial data on the mobile device. The geospatial data has to go through several steps to be suitable for rendering. First we prepare the data using FME tools, then transform it to the ENU coordinate system and finally send it to OpenGL ES to be rendered to the screen.

The virtual objects will be rendered on top of the real objects in the picture, but due to the inaccuracy caused by the sensors, the virtual objects will not be matching their corresponding physical objects in the picture. The user can fix this using the correction tools by shifting or rotating the model around the X,Y and Z - Axes until the virtual objects match the objects in the picture.

Chapter 5

Results and Discussion

Contents

During development and testing the prototype we encountered several issues that we find important for our approach. The purpose of this chapter is to discuss these issues and see what kind of improvements could be applied..

First of all we will start by presenting some preparations regarding the testing area and the geospatial data that will be tested in that area. This is followed by discussing some issues that we encountered during the testing, issues effecting the GIS data, the sensors as well as the overall usability. Finally we discuss solutions and enhancements for these issues.

5.1 Preparation

Before testing the prototype important preparations have to be made. As explained in the previous chapter, using the FME tools [12], the geospatial data will be imported from GIS, processed and translated to latitude-longitude coordinates and prepared for testing.

In order to do this we first select the area where the testing will be performed, then get the geospatial data for that area.

- **Testing area**

  This area is in the city of Graz in Austria, it lies exactly at the intersection between Anzengrubergasse and Brockmanngasse, Figure 5.1.

Figure 5.1: Testing area, this is an aerial detailed image of the testing area.

The purpose of this work is to provide an interactive user interface for correcting the inaccuracy of registered GIS data on mobile devices. Therefore, we concentrate on testing the accuracy, usability in general and the usability of our manual correction method in particular. For this reason, we are not using the whole area for testing, as it requires a lot of time to prepare the data. Instead we are using parts from buildings as a testing sample. The selected testing sample is enough for our purpose.

Figure 5.2 shows the testing GIS data superimposed on the image of the testing area. The red lines are underground water pipes and the two corner shapes in blue represent the corners of the buildings.

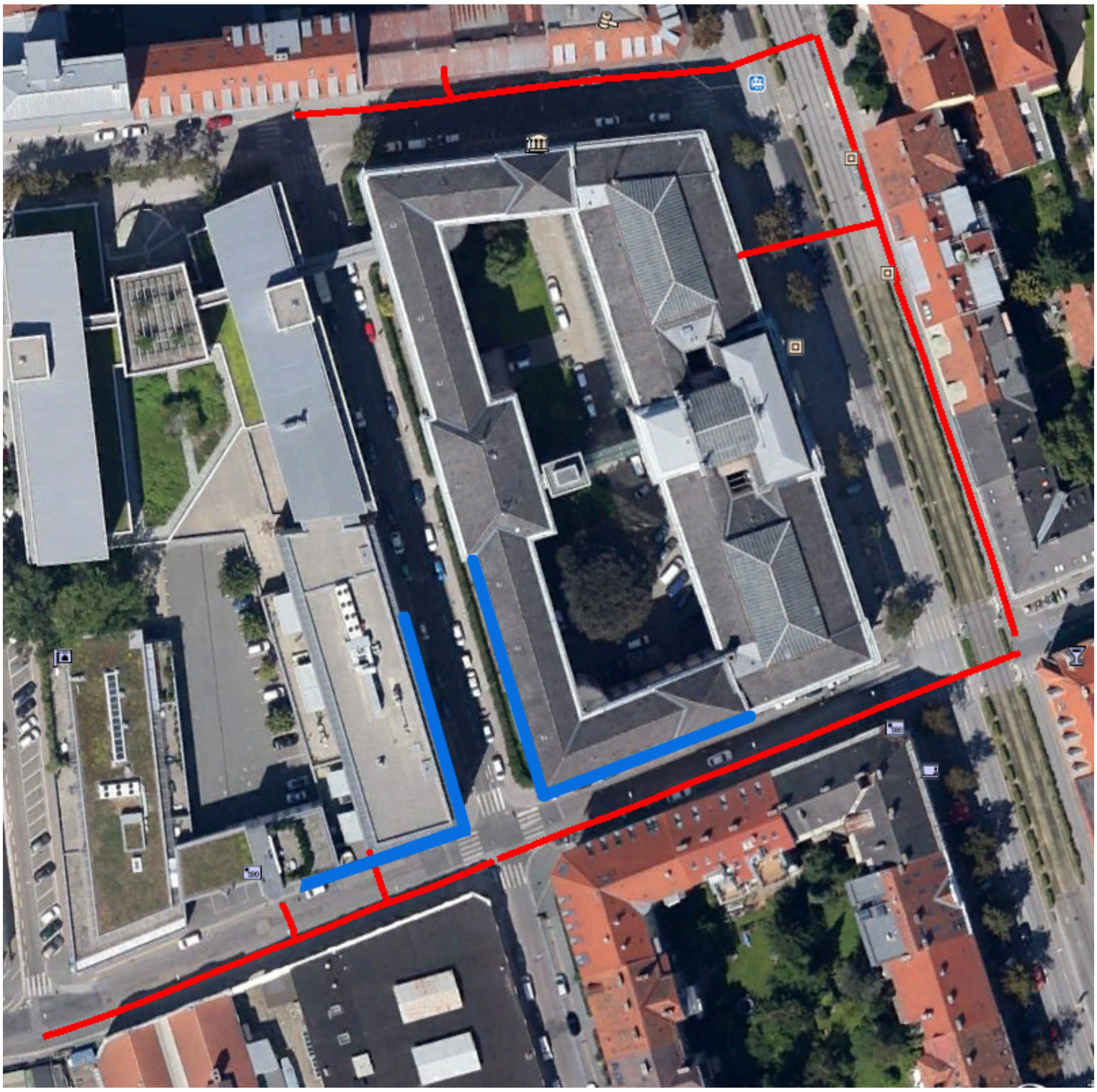

Figure 5.2: Visualizing the testing GIS data on a satellite view image of the testing area.

- **Testing data**

  The testing data was prepared in two .obj files using the FME tools, one file for the buildings vertices and the second for pipes vertices. The files contain the geometries vertices in the latitude-longitude coordinate system.

  As mentioned earlier, the GIS data which we use, comes in a 2 dimensional form, that means there is no information about the height of buildings or the depth of the underground pipes and we have to provide these information manually. Therefore, the height of the buildings was fixed to 10 meters and the underground pipes were fixed to -2 meters.

5.2 Issues

During the testing of the prototype in the predefined testing area we encountered some issues which have to be considered in order to improve not just the accuracy but also the usability, thus improving the overall quality of our approach. These issues are discussed in the following three sections.

5.2.1 GIS data

Having a 3D representation of the model is essential in order to visualize the rendered scene correctly. One of the most significant problems that is related to our GIS data, is the fact that the GIS data that we use does not support 3D modeling in a sufficient way. For instance, we can not have information about the heights of buildings or depth of underground pipes from the GIS. As we tried to handle this problem it turned out to be easy to solve, however not in the best possible way. While it is possible to visualize the 3D model of the buildings, fixing all buildings heights to 10 meters shows all buildings as giant blocks having the same height, which does not look real. Normally, houses and buildings have different sizes and heights.

The real problem lies with the depth of the underground pipes, because the purpose of this prototype is to determine the exact location of pipes and not constructing a 3D model of the city, we only use the buildings as references to correct the rendering of pipes to the right location.

Knowing the exact depth of underground pipes is not an easy thing to do and is beyond the capacity of this work, which is why we will limit ourselves to the idea of fixing the depth of underground pipes to -2 meters.

Another important issue concerning the GIS data is the meta data. Having information about the underground pipes is an important feature. While visualizing the pipes information such as which pipe is a water pipe and which is an electricity pipe is considered a basic information for any person works at the construction site.

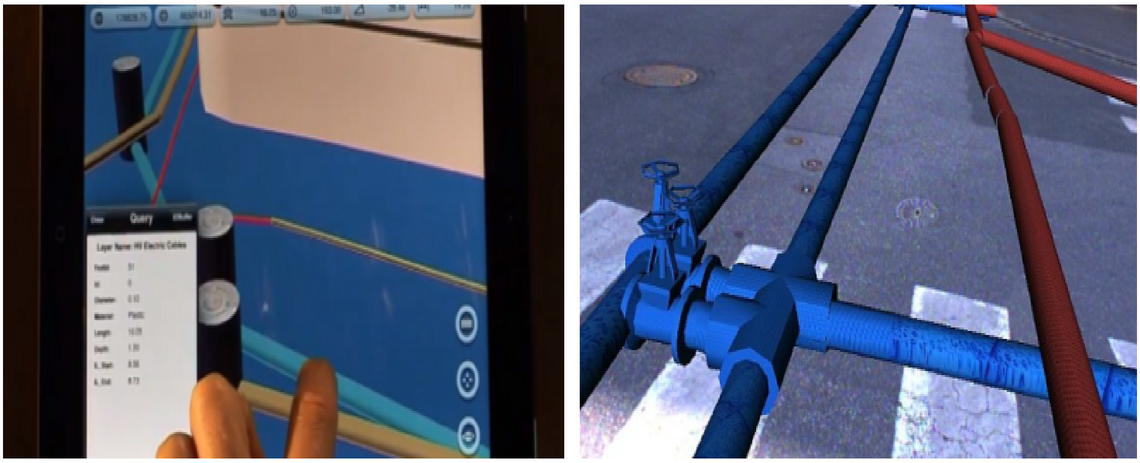

Figure 5.3: Different ways of providing the user with meta data on the screen. The picture to the left shows the user selecting the object to get information about it. The picture to the right shows colored underground pipes, the red color represents electricity pipe and the blue represent water pipe. [23] [1].

Figure 5.3 shows such features, we can provide information either by showing a text message to the user or in more simple way by giving the pipes different colors.

In our case the .obj file contains only geometries and no other information, this is why it is not suitable for providing such information. Instead we could use other file formats that can contain more information about the different GIS data and can be imported directly from any GIS.

5.2.2 Sensors

As mentioned earlier, the data that we get from the device sensors are inaccurate, of course we can rely on it in many situations and cases but unfortunately in our case it has to be perfect. Such inaccuracies effect our rendered model in many ways. The three main inaccuracy effects are GPS position, height and rotation.

5.2.2.1 GPS Position

The GPS accuracy depends on the number of visible GPS satellites that the mobile device receives the location information from. It might take several minutes for the device to find all visible satellites [2].

During the testing of the prototype we were using a simple map application on iPad called Maps. With the Maps application we determine our location in the globe, as shown in Figure 5.4.

The current approximate location is indicated using the blue marker. The blue circle changes it position continuously to approach the correct location, in other words, the accuracy gradually increases over time, but that might take a while and will not reach the needed accuracy. On the other hand, the red marker was determined by the user, it is an approximation of the correct position by looking at the objects (e.g, Houses, streets) in the map and comparing them with objects in the real location, allowing the user to roughly determine his location.

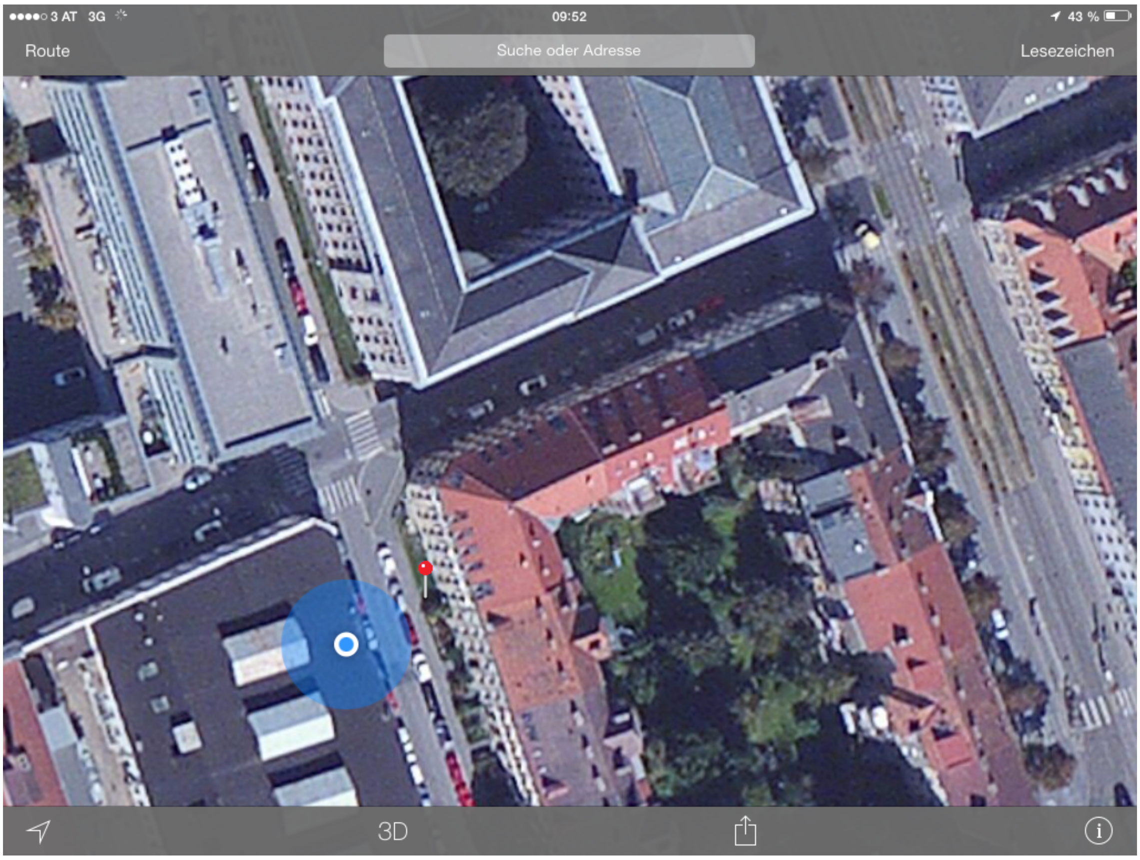

Figure 5.4: A screenshot of the **Maps** application on iPad, taken in the testing area while testing our prototype. The blue circle represents the GPS position that is measured with the iPad's sensors, on the other hand the red marker was put by the user, it represents an approximation of the correct position of the device.

When starting the prototype and load the model with this GPS data we see the big offset between the model and the real world as shown in Figure 5.5.

However, we tested the prototype using a fixed GPS position. We chose a specific

Figure 5.5: A screenshot of the implementation prototype. Rendering the model using the device inaccurate GPS position.

point at the testing area and used Google maps to get the latitude-longitude coordinates for that point, then we ran the prototype from that point using the specified coordinates as our GPS position. The rendered model was almost accurate. Figure 5.6 shows the rendered model with a fixed GPS position.

As shown, the GPS position effects the rendering of the model, which is why it is important to have an accurate position. The idea of waiting for the accuracy to be increased might take a long time, thus a new idea emerged while testing the prototype, which is not hard to implement and at the same time provides us with more accuracy within a shorter period of time.

The idea is still in the developing phase and it has not been implemented yet, but as a future work we think it would provide better results.

The different kinds of maps like Google maps, Open street maps or the Maps application provide us with satellite images, roads images, directions and much more, but at

Figure 5.6: A screenshot of the implementation prototype. Rendering the model using a fix GPS position (constant latitude-longitude value).

the same time they have features that return the GPS coordinates of a fixed point on the map. Such features could be used to enhance our prototype. The user would start the prototype normally, the device will determine the location by using the data from the GPS sensor. Instead of waiting, the user opens a map, like the one shown in Figure 5.4, and determines his approximate location using the red marker, after that the prototype will take the latitude-longitude coordinates under that red marker and use it as a fixed GPS position for the whole rendering process. In this case we will reduce the inaccuracy from 5 or 25 meters to 1 or 1.5 meters respectively, allowing the user to easily apply the manual correction method.

5.2.2.2 Height

The hight of building was previously discussed in Section 5.2.1, together with the GIS data. This section is in fact more concerned with the altitude measurement from the device. As in the inaccuracy issue with the GPS position, the sensors provide us with an inaccurate altitude as well. Figure 5.7 shows such inaccuracy when rendering the model. We can see the inaccuracy of height in the rendered model compared to the real world.

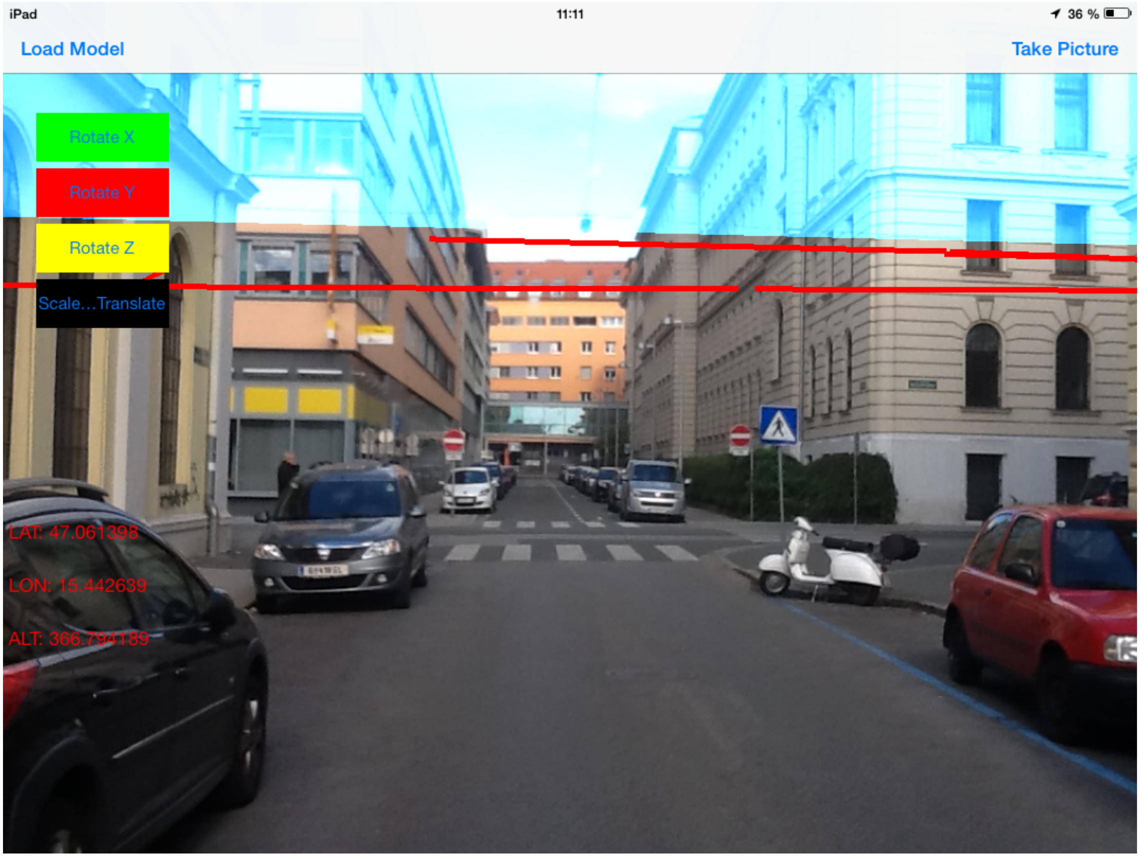

Figure 5.7: A screenshot of the implementation prototype. Rendering the model using the altitude measurement from the device sensors.

As a solution for this issue, we fixed the altitude to 1.5 meters over the ground. This is as well not very accurate, yet it is much easier for the user to apply the manual correction method when the height has 1-3 meters inaccuracy. Figure 5.8 shows the rendered model after fixing the altitude.

5.2.2.3 Rotation

Another issue caused by the sensors inaccuracy is the presence of inaccuracies in rotations when rendering the model. For example, if we have a closer

Figure 5.8: A screenshot of the implementation prototype. Rendering the model after fixing the altitude to 1.5.

look on Figure 5.9, we will see that the rendered model is slanted a little bit toward the right side (around the Z-axis) and also toward the right side (around the Y-axis).

5.2.3 Usability

All these methods discussed earlier, extracting the GPS coordinates from a map or fixing the device altitude are nothing but attempts to make it easer for the user to use the manual correction without putting too much efforts.

The manual correction method is an efficient method when the inaccuracies are not too big. But efficiency is not the only thing that matters, applying the manual correction, the user has to switch between the different axes every time he wants to correct the rotation around an axis, and that will cause a kind of distraction or confusion for some users.

Figure 5.9: A screenshot of the implementation prototype showing the rotation inaccuracies.

So as a future work the usability has to be improved so that the user could easily deal with such corrections, or perhaps he does not have to make any correction at all.

In the next chapter, we will discuss some ideas that enhance the usability in particular and the overall approach in general.

5.3 Summary

Before we start testing the prototype we prepare the testing data and testing area. During the testing of the prototype we encountered several issues which have to be addressed in order to improve not just the accuracy but also the usability and therefore the overall quality of our approach.

Issues related to the GIS data, Sensors and Usability are important to be considered and solved, in order to provide the user with an accurate registration of data and a reliable user interface.

Chapter 6

Conclusions and Future work

The goal of this project was to develop an Augmented Reality interactive approach on the iPad platform, that can be used to register the GIS data accurately on the screen. This chapter briefly debates the results we have obtained and introduces future approaches, it also describes why we think we have succeeded in reaching the stated goal.

6.1 Conclusion

We built on existing Augmented Reality GIS applications and enhanced our approach by using the power of smartphones and tablets. In addition we provided tools that give the user the ability to correct the GIS data rendering misalignments caused by the smartphones or tablets sensors inaccuracy.

The interactive tool for correcting the rendering misalignments is an efficient method when the inaccuracies are not big. The inaccuracies depend on sensors and the mobile devices sensors could be affected by many factors, therefore our prototype requires additional modifications in order to be used by the utility workers. A simple modification that could enhance our prototype's efficiency and usability is to apply the location approximation map idea discussed in the previous chapter. Another modification that has to be developed before the utility workers could benefit from our prototype is to improve the way we process the GIS data. Our prototype should be redesigned to include meta data and provide the user with more information about the GIS data and the surroundings.

6.2 Future work

Moreover, our prototype is prepared for future extension and modification. It does in fact have many features that could be modified from many aspects. For example, additional visualizations to the virtual model could be added easily using OpenGLES, or to make the GPS position more accurate we could attach an external GPS devise to the iPad, that strengthens the iPad's GPS receiver.

The next step of this work will be using a video stream camera instead of a steady picture. Such implementation will not be much different from what we presented here, but the rendering of the model will be live and the user can walk and change the position and orientation of the device while everything is calculated at run time. For such implementation the GIS data will not be saved in a file but instead it will be imported directly from the GIS server.

We could think of many new ideas, improvements and enhancements for our work because it is a combination of several powerful technologies which are being developed very fast separately. The faster these technologies develop the more efficient our work will be.

Bibliography

[1] (2011). Smart Vidente, The information was taken from :. http://www.vidente.at/.

[2] (2013). iOS 5: Understanding Location Services, The information was taken from :. http://support.apple.com/kb/ht4995/.

[3] (2013). Latitude and Longitude, The information was taken from :. http://www.grossmont.edu/judd.curran/GISoutlineSV.htm/.

[4] (2013). Wikitude, The information was taken from :. http://www.wikitude.com/.

[5] Annette, G. (2013). The Mobile Web: An Illustrated Look. http://eatongolden.com/view-may13.php/.

[6] Ayub, S., Bahraminisaab, A., and Honary, B. (2012). A sensor fusion method for smart phone orientation estimation. In *Proceedings of the 13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool*.

[7] Azuma, R. T. et al. (1997). A survey of augmented reality. *Presence*, 6(4):355–385.

[8] Buck, E. (2012). Learning opengl es for ios: A hands-on guide to modern 3d graphics programming.

[9] Feiner, S., MacIntyre, B., Hollerer, T., and Webster, A. (1997). A touring machine: prototyping 3d mobile augmented reality systems for exploring the urban environment. In *Wearable Computers, 1997. Digest of Papers., First International Symposium on*, pages 74–81.

[10] Feiner, S., Macintyre, B., and Seligmann, D. (1993). Knowledge-based augmented reality. *Commun. ACM*, 36(7):53–62.

[11] Frank, C., Caduff, D., and Wuersch, M. (2004). From gis to lbs–an intelligent mobile gis. *IfGI prints*, 22:261–274.

[12] Inc., S. S. (2004). *Feature Manipulation Engine (FME) Functions , Factories and Transformers*.

[13] iOS Developer Library (2013). CMAttitude Class Reference, The information was taken from :. https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CMAttitude_Class/Reference/Reference.html/.

[14] Kansal, A. and Zhao, F. (2007). Location and mobility in a sensor network of mobile phones. In *Proc. ACM SIGMM 17th Int?l Workshop on Network and Operating Systems Support for Digital Audio & Video*.

[15] King, G., Piekarski, W., and Thomas, B. (2005). Arvino - outdoor augmented reality visualisation of viticulture gis data. In *Mixed and Augmented Reality, 2005. Proceedings. Fourth IEEE and ACM International Symposium on*, pages 52–55.

[16] Kresse, W. and Danko, D. M. (2011). *Springer handbook of geographic information*. Springer.

[17] P, B.-F. (2010). SIG - Geographic Information Systems. `http://www.planete-tp.com/en/sig-geographic-information-systems-a1047.html/`.

[18] Piekarski, W. and Thomas, B. (2003). Tinmith - mobile outdoor augmented reality modelling demonstration. In *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*, pages 317–318.

[19] Prakash, A. (2004). Geographical Information Systems – An Overview.

[20] Reitmayr, G. (2011). Sensors, The information was taken from :. `http://www.icg.tugraz.at/Members/gerhard/mvc/MVC_04_Sensors.pdf/view/`.

[21] Reitmayr, G. and Drummond, T. (2006). Going out: robust model-based tracking for outdoor augmented reality. In *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*, pages 109–118.

[22] Schall, G., Zollmann, S., and Reitmayr, G. (2012). Smart Vidente: advances in mobile augmented reality for interactive visualization of underground infrastructure. *Personal and Ubiquitous Computing*, 17(7):1533–1549.

[23] Sivan, D. (2012). 3D-GIS in the Cloud. `http://www.sivandesign.com/products/3dgis/3d-gis-app/`.

[24] Zhang, F., Stahle, H., Chen, G., Chen, C., Simon, C., and Knoll, A. (2012). A sensor fusion approach for localization with cumulative error elimination. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pages 1–6.