NTNU

Norwegian University of
Science and Technology

Tracking for Outdoor Mobile Augmented Reality

Further development of the Zion Augmented Reality Application

Tor Egil Riegels Strand

Master of Science in Computer Science
Submission date: July 2008
Supervisor: Torbjørn Hallgren, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem Description

To successfully combine the real and the virtual worlds in Augmented Reality, one must known the users position. The goal of this Master's Thesis is to try out methods of tracking the user by using equipment suited for outdoor use at Sverresborg. The best method will be used with the Zion Augmented Reality Application, which lets a user see a castle, that was torn down centuries ago, as it once stood.

Assignment given: 17. January 2008
Supervisor: Torbjørn Hallgren, IDI

Abstract

This Master's Thesis deals with providing tracking to an outdoor mobile augmented reality system and the Zion Augmented Reality Application. ZionARA is meant to display a virtual recreation of a 13th century castle on the site it once stood through an augmented reality Head Mounted Display.

Mobile outdoor augmented/mixed reality puts special demands on what kind of equipment is practical. After briefly evaluating the different existing tracking methods, a solution based on GPS and an augmented inertial rotation tracker is further evaluated by trying them out in a real setting. While standard unaugmented GNSS trackers are unable to provide the level of accuracy necessary, a differential GPS receiver is found to be capable of delivering good enough coordinates.

The final result is a new version of ZionARA that actually allows a person to walk around at the site of the castle and see the castle as it most likely once stood. Source code and data files for ZionARA are provided on a supplemental disc.

Acknowledgments

First I would like to thank Torbjørn Hallgren for making it possible to do this interesting project and for his assistance along the way.

I would also like to thank Jo Skjermo for sharing his experience and thoughts about Head Mounted Displays, for trying to help troubleshoot the HMD, and for a brief discussion about visual tracking at the site.

Gunnar Lien continued to lend his personal Bluetooth USB adapter for use in the project.

Terje Skogseth provided some more insight into GPS technology and also some other tracking technology. He also provided a list of organizations which might have some differential GPS equipment that might be useful for the project.

Olav Nygaard at Leica Geosystems was kind enough to lend us a Leica Geosystems GPS1200 differential GPS receiver and let us use one of their reference stations one week for free.

And last but not least Trøndelag Folkemuseum for allowing me to run around on the site with this odd-looking equipment and for taking time to check out the so far final result.

Contents

Chapter 1

Introduction

This Master's thesis is part of a potentially long running Augmented Reality project by the visualization group at the Department of Computer and Information Science at the Norwegian University of Science and Technology in Trondheim.

The goal of the project is to digitally recreate the former castle at Sverresborg, also called Sion or Zion, in an outdoor environment using equipment the user can carry on him/her, and blend that virtual model into the real environment. Challenges that must be solved include creating a model of the castle, finding a way to shade the model so that it fits the real world light conditions, set up a system for tracking the users position, and perhaps put some interactive and educational content into the scene. The project is purely academic, and will just try out what is possible and perhaps push the frontiers.

Hardware currently available to the project consist of a laptop, an Head Mounted Display, a GPS receiver and an inertial tracker. For deployment in the field, the plan is to carry the laptop in a backpack. The rest will be mounted on the headset.

Chapter 2

Earlier work

2.1 Virtual content

As Sverresborg has been the topic of several earlier projects at the institute, there already exist some virtual content ready to be used. The central piece is a 3D-model of the castle was made several years ago and has been used in several other projects already. Last year, it was imported into Blender[1] and improved by Johannes Odland specifically for this Augmented Reality project.[15] The Blender model is made up of 46,662 vertices and 45,177 faces (some are triangles, some are quadrilaterals). It is exported into OpenSceneGraph's native format for geometry.

There are also some animated 3D-models of people from that era. These have been used in conjunction with Sverresborg in another project. They are not to be included into ZionARA at the moment.

2.2 ZionARA

Odland also made a basic framework for the Augmented Reality application, called *Zion Augmented Reality Application*, or just *ZionARA* or *ZARA* for short. ZionARA was capable of reading video from the cameras on the Head Mounted Display and composite the two video streams into the background of a stereoscopically rendered 3D-scene. For testing purposes, ZionARA could also function with only one camera and normal non-stereoscopic rendering. When rendering stereoscopic, the window was split in two, with each half corresponding to one camera/eye.

Delta3D[2] was used for graphics and PC-style input. It is in turn a layer on top of among other things OpenSceneGraph[3], which in turn is based around OpenGL. Microsoft's DirectShow was used to get streaming video data from the cameras in the HMD.

Although attempts had been made at making ZionARA ready for input from GPS and inertial tracker, it only used visual tracking to match the real and virtual worlds. By using a library called ARToolKitPlus[27], ZionARA could get the camera-relative position and orientation of a square black-and-white marker pattern. ZionARA inverted this transform and applied it to the camera, while the castle was positioned at the origin

[1] http://www.blender.org/
[2] http://www.delta3d.org/
[3] http://www.openscenegraph.org/

of the world coordinate system. It gave the appearance of the castle positioned on top of the marker pattern.

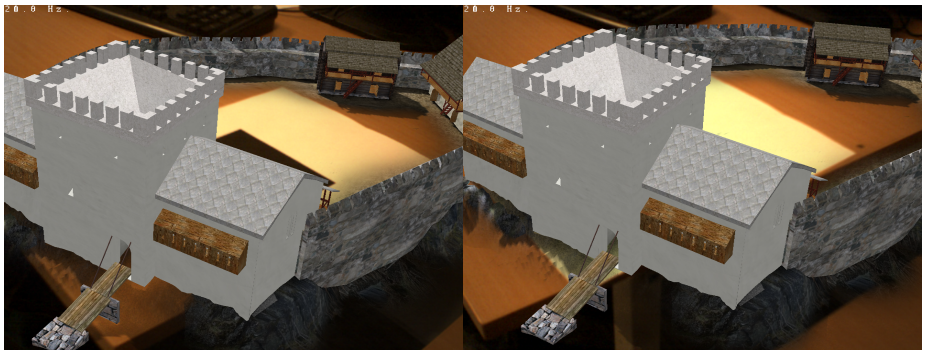


Figure 2.1: Screenshot of ZionARA stereoscopic mixed reality. The castle is positioned on top of a black-and-white marker.

Chapter 3

Other Outdoor Augmented Reality systems

3.1 ARQuake and Tinmith

ARQuake[24, 17] is an augmented reality outdoor game developed at the University of South Australia. It is based on the popular computer game Quake[1] by id Software, which was released as open source[2] under the GNU General Public License. This meant that the team only had to focus on the augmented reality related problems and did not have to make the content.

The hardware system it runs on is called Tinmith[16]. This system has evolved over the years, but some general features are constant. The core is a portable laptop computer which runs the software. A headset is used to overlay the virtual world over what the user sees. Some form of GPS and some form of orientation sensor are used to get the users position and orientation. As of 2004, they used an RTK GPS and an InertiaCube2 orientation tracker. At the time of ARQuake, the system also included a gun-like hand controller. It could not be used for aiming as it was not tracker, but its buttons were used to perform various actions. Recent descriptions and photos of Tinmith include gloves instead.

Early versions of ARQuake also used visual tracking, but this seems to have been left out in later versions. The reason may be that they shifted focus to a 100% outdoor solution, whereas the early ARQuake was labeled as both indoor and outdoor. GPS works very poorly indoors.

One potential drawback with their solution, is how they mix the real and virtual world when presenting it to the user. In 2002, they used a headset with a half-silvered mirror. Light coming from the real world passed through the mirror, where it was mixed with a reflected image from an LCD display showing the virtual world. As only bright objects get reflected strongly enough to overshadow the light from the real world, only brightly colored objects could be used in the virtual world if they were intended to be visible. Since black objects were invisible, recreations of real world buildings were created in black. During rendering, these black buildings would obscure visible virtual objects that were located behind the real world objects they represented, but would be invisible to the user. Although Tinmith now uses a headset which does not have a half-

[1] http://www.idsoftware.com/games/quake/quake/

[2] http://www.idsoftware.com/business/techdownloads/ (retrieved 2008-06-05)

Figure 3.1: Screenshot from ARQuake

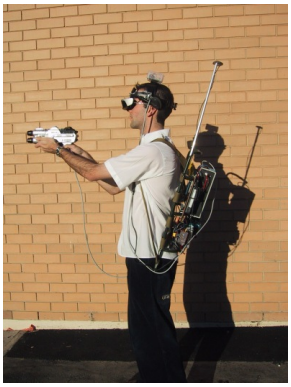


Figure 3.2: Early version of the Tinmith Backpack

Figure 3.3: New Tinmith Backpack (2006)

silvered mirror, they still use black as transparent and mix the camera images with the rendered images in a separate device to off-load the computer.

It should be noted that there have been no updates on the ARQuake homepage since 2002, but the Tinmith system has been developed at least until 2006.

3.2 University of Nottingham

The Institute of Engineering Surveying and Space Geodesy at the University of Nottingham and the School of Chemical, Environmental and Mining Engineering have also been working with outdoor Augmented Reality systems.[20, 19] In 2002, they worked on a system which allowed a person, for instance an excavator, to see the pipes and cables running below ground.

To track the users position and orientation, or rather the position and orientation of the device he or she looks trough, they used an RTK GPS and an inertial navigation system. They tested three different viewing devices: a headset, a binocular-like device and just simply a laptop with a camera. For their purposes, which was to take a look underground now an then, a headset was too cumbersome. The time it took for the computer to update the displayed image also caused some motion sickness. In the end, they settled for the virtual x-ray binocular solution.

Chapter 4

The challenge

The purpose of augmented or mixed reality is to mix a virtual world with the real world in such a way that they become one. While the creation of seamless worlds consisting of a mix of real and non-real elements has long been accomplished in for instance movies, augmented reality poses a set of additional demands.

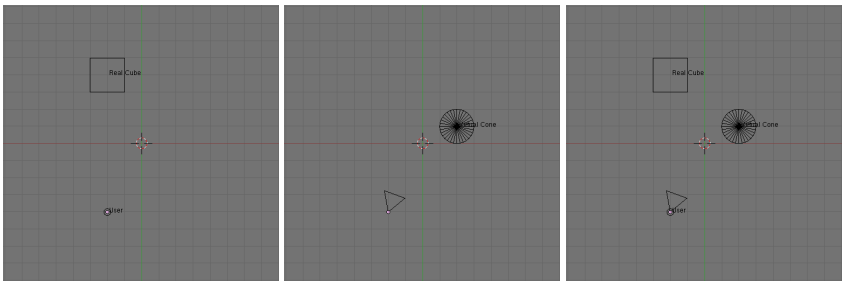
Augmented reality is interactive, which means that in general nothing is preplanned. The user should be free to do things whenever he/she wants and go wherever he/she wants. Certain things can be scripted, whether they are background action or responses to some actions performed by the user, but the user, and therefore the virtual camera, position is not. This is a crucial difference from movies, where one may plan ahead every detail to make sure the real and non-real elements fit together.

Another related difference is that augmented reality is, at least in this case, real-time. When blending real and non-real elements in a movie, the people responsible for that have the advantage of being able to do things over. They can try things out and tweak the results over and over until the line between reality and virtuality has been sufficiently blurred. There is no such luxury in augmented reality. The mixing of real and virtual must happen in real time and without human intervention, at least on a frame-by-frame basis.

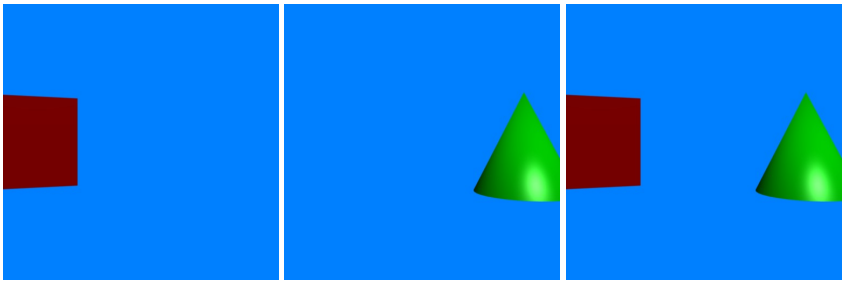
This project is also about mobile augmented reality, which puts a limit on the equipment used. There is no place for of heavy and/or large equipment to bring virtual worlds to life or process the real world for data necessary to merge it with the virtual.

On the other hand, there is no need for augmented reality to be photorealistic, at least not yet. The virtual world could look like a cartoon or simple drawings, but it must seem to occupy the same space as the real world. Examples of this are movies like *Who Framed Roger Rabbit* and the recent *Looney Tunes* movies. While it is generally easy to tell the real actors from the cartoon figures, they still appear to share the same space. When the camera pans left, both real and cartoon elements move towards the right on screen, unless there is a gag requiring something different. The same is true with augmented reality. When the user turns to the left, the virtual world must rotate to the right to match the real world. When the user moves forward, the virtual objects in front of the user must come closer just like objects in the real world.

To achieve this, the system must be able to know the users position in the real world. There must also be a reference point in common between the two worlds. If the tracking system reports that the user is two meters left of the real world reference point, the virtual camera in the virtual world must be position two meters to the left of the virtual world reference point. The problem is how to find the users real world



(a) The real world (b) The virtual world (c) The mixed world



(d) The real world as seen by user (e) The virtual world as seen by virtual user (f) The mixed reality as seen in headset

Figure 4.1: The real, the virtual and the mixed world. By placing the camera in (b) at the same location and facing in the same direction relative to the common reference point as the user in (a) is, the worlds become aligned. This is important for merging the worlds.

position, and that is what this project will try to find a solution to.

4.1 The site

The site where Sion once stood is on top of a steep hill (figure 4.2). Sion covered the entire summit, which is about 80 meters long, 60 meters wide and roughly oval in shape. The summit is generally flat, except for a big depression running from the center and out to one side. Tall cliffs surround the summit on three sides. On the fourth side, the surrounding landscape rises almost up to the height of the hill, giving a much shorter cliff face. A bridge runs from a tiny neighboring hill over to the summit. Earlier, Sion's drawbridge covered this short span.

4.2 The virtual model

As mentioned earlier, the 3D model of the castle has been inherited from earlier projects related to Sverresborg. The model originally featured the hill, which was important in the earlier projects that was not composited on top of the real world. In mixed reality, the goal is to use as much of the real world as possible. While the hill still exists in the model, it gradually fades out away from the buildings. The virtual hill is therefore used to clip the virtual buildings against the terrain, so subterranean parts are hidden from

Figure 4.2: The hill as seen from the northwest

the user, and provide a smooth transition between the virtual and real world.

Although the hill has been modeled using height survey data, it is uncertain exactly how it will fit with the landscape. This can only truly be verified once the tracking system is in place.

Chapter 5

Available tracking systems

There are many ways to track a user's position in 3D space, which is necessary for Virtual Reality, Augmented/Mixed Reality and motion capturing. Typically, these kinds of things happen in an indoor controlled environment where special equipment has been installed. This project is however focused on outdoor use, requiring little, if any, equipment to be rigged up at the site before use.

Earlier studies, both general and specific to this project, have considered the usability of the different types of trackers.[3, 14, 23] Generally there are five types of trackers: GPS/GNSS, inertial trackers, active trackers, passive trackers, and compass and tilt sensors. Some tracking devices are a combination of these types. Examples include GPS receivers with built-in compass, and inertial trackers with magnetic and/or tilt sensors.

Some of the tracking systems are unable to provide all the necessary data on their own. A pure GPS receiver only reports position, while compasses and tilt sensors only report orientations. Inertial trackers may, by integrating linear acceleration, calculate how much the user has moved, but only relative to some starting point. This is unfortunately prone to accumulate errors, decreasing accuracy with time. The solution presented is therefore hybridization, where multiple types of tracking devices are used, each complementing or correcting the others.

5.1 GPS

GPS stands for Global Positioning System[26], and is a system developed and run by the United States Department of Defense. Its real name is actually NAVSTAR GPS. Initially a military project, it has been mostly opened for civilian use. The system is made up of three components: the ground based control centers, the orbiting satellites and the receivers.

There are a minimum of 24 satellites circling the planet every 12 hours in one of six orbits. At least 6 are visible from any point on the planet, unless obscured by buildings, cliffs or other things opaque to radio waves. From orbit, these satellites continuously send a set of signals down to the surface. The satellites also carry atomic clocks so that they can keep an accurate time, which is sent to the receivers along with the satellite's position in orbit, or ephemeris.

The control station monitors the satellites, using monitoring stations around the world, and makes sure their clocks and reported ephemerides are correct.

5.1.1 How it works

From the signal sent from the satellite, a GPS receiver will know when the signal was sent and from where. If it knows when the signal was sent and when it received it, it can calculate how far away it is from the satellite. This pins the receivers position down to anywhere one a sphere surrounding the satellite. By calculating the distance to another satellite, the receiver while limit its possible locations to a circle formed where the two spheres intersect. Three satellites pin the location down to two points. Unfortunately, the clock of the receiver is not normally as accurate as the atomic clocks of the satellites. This introduces some errors into the calculation that may cause the spheres not to intersect. The receiver must therefore tweak the calculations until a solution is found.[11]

Several signals are sent by the satellites, but only the L1-signal is currently available for civilian use. It is transmitted at a frequency of 1575.42 MHz. Part of the signal is covered by the P-code, which is a secret military signal. The other part is the C/A-code, which stands for *coarse acquisition*. It is a 1023 bit Pseudo Random Noise code sent at 1.023 Mbit/s. Each satellite has a unique C/A-code. Overlaid on top of this signal is the Navigation Message. It contains the timestamp, ephemeris and other data.

To make sense of the signal, the receiver generates the PRN-code for each satellite and tries to match it against the incoming signals. Since the signal is delayed, the receiver must shift the generated code until it matches the received code. Once that is done, the receiver knows it has found the signal of that satellite. It then removes the PRN-code from the signal and is left with the Navigation Message. The receiver then has all the data required to measure the distance to the satellite and position it relative to the other satellites when trying to intersect the spheres.

Originally, the NAVSTAR Global Positioning System contained a feature called *selective availability* (S/A). This deliberately degraded the quality of the civilian signal so that their enemies could not use the system against the United States of America. With the increased use of civilian GPS receivers aboard boats, airplanes and even within the military itself, turning S/A on could hurt the United States more than leaving it off. The president therefore signed an order to disable selective availability. New satellites no longer have this feature.

5.1.2 Accuracy and GPS augmentation

Getting an accurate enough estimate of the users position is one of the largest problems in this project. Normal GPS receivers are only accurate to within a couple of meters at best.

The main cause of GPS errors is the ionosphere. As the signal travels through it on the way down from space, it gets delayed. How much it gets delayed vary with time and space depending on the atmospheric conditions. The satellites send some information about this, but errors still remain. Atmospheric effects reduces accuracy down to about 5 meters.

Ephemeris and clock errors make up the next two largest errors sources. Both reduce the accuracy with about 2 meters. Both are corrected by ground stations, but only every two hours. Other errors are caused by signals being reflected off various surrounding objects before reaching the receiver.

The last source of error is the frequency of the signal itself. The C/A-code is sent at about 1 Mbit/s or 1 MHz. Traveling at the speed of light, roughly 3×10^8 m/s, a bit covers almost 300 meters. If the receiver can only lock onto the C/A-code with

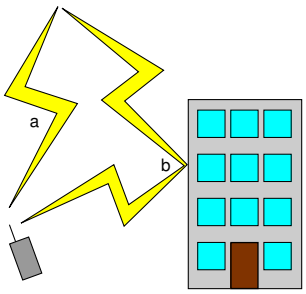


Figure 5.1: Signal being reflected of a building. The receiver must detect that signal b is a delayed version of signal a that has bounced off a building and not use it to calculate position.

1% uncertainty, there would still be an error of 3 meters. In worst case scenario, the receiver could lock onto the signal halfway out of sync, giving a measurement that is 150 meters off.

This project is not the only case where increased GPS accuracy is wanted, and several solutions are available to get a more accurate position. All of them primarily address the ionospheric errors. Some augmentation systems have been ignored since they only cover specific areas, like WAAS, that only covers North America. Developers of Augmented Reality elsewhere should also evaluate local and regional solutions.

5.1.2.1 Military signal

The military itself is one of those who want a more accurate signal, and they already have it. GPS satellites send a signal on at least two different frequencies, all containing a P(Y)-code. Using this code, it is possible to calculate how much the GPS signal has been delayed. Unfortunately, this code is encrypted and still restricted for military use. It is theoretically possible to use the undecrypted codes, but this is slower.

5.1.2.2 Differential GPS

Differential GPS (DGPS) is a GPS augmentation system that employs a reference station to broadcast corrections to the signals from the GPS satellites. The accuracy of DGPS is also only less than 1 meter close to the station. For every 100 km the user moves away from the station, the accuracy drops with about 0.2 meters.[13] There are two reasons why the accuracy drops with distance. The first is because the corrections calculated for one location, are not necessarily valid for other locations. Atmospheric conditions change with space and time, and so does the effects they have on the radio signal passing down from space. The reference station may receive a signal that requires little correction, while the GPS user may get an inaccurate signal which the signal from the station does not correct.

Another reason for the degradation of DGPS accuracy with distance from the station, is that the station and the user will start having a different view of the sky. The two will no longer see the same set of satellites, and DGPS can not correct the signal from satellites it can not see.

Reference stations may be fixed installations, possibly provided by a third-party, or portable devices.

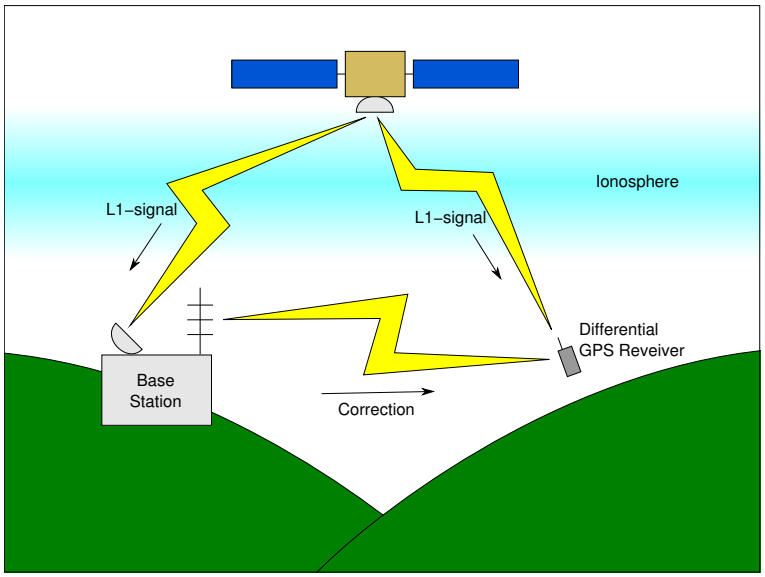


Figure 5.2: Basic operation of Differential GPS. Base station at known location receives GPS signal from satellite and calculates how wrong it is. The result is sent to GPS receiver, which then can correct the now known error in the satellite signal.

Another outdoor Augmented Reality system called ARQuake uses a $4,000 advanced GPS which provides an accuracy of 50 cm at 10 Hz.[2]

5.1.2.3 Real-time kinematic GPS

Real-time kinematic GPS is a type of DGPS that also uses a reference station to broadcast corrections to the signal coming from the satellites.[21] Unlike many other DGPS solutions, RTK GPS is capable of centimeter, or even millimeter, accuracy. The main difference is that most GPS and DGPS receivers lock onto the C/A code transmitted from the satellites, whereas RTK GPS locks onto the carrier frequency used to transmit the C/A code. Since the carrier frequency is higher than the bitrate of the signal, about 1500 MHz and 1 MHz respectively, the receiver can pin-point its position more accurately by synchronizing with the carrier. There does not appear to be any non-differential carrier phase GPS receivers, most likely because the accuracy of the carrier frequency is useless without correction for ionospheric delays and ephemeris errors. The centimeters might be right, but the meters could be way off.

RTK GPS is often used for surveying, and the surveyors usually have their own base station. This base station is placed at a known location, and the so-called rover can then calculate its location based on GPS signals and the signal from the base station. There can be several rovers for each base station. It follows that the accuracy of the positions reported by the rovers are only as good as the accuracy of the base station's position.

Receivers capable of using RTK GPS have already been proven useful in Augmented Reality requiring high accuracy while outdoors.[19] The receiver was used to track a headset, a laptop or a binocular looking viewing device. In their controlled tests, a variation of just a few millimeters were measured in the X-Y plane. This should be more than adequate for ZionARA, as the 3D model of the castle does not have accuracy

at that level. It is also possible that the inaccuracies in the separate orientation tracker will have a greater impact on the virtual world's ability to line up with the real world.

5.1.2.4 EGNOS

The European Geostationary Navigation Overlay Service[1] is a satellite based augmentation system, not only for GPS, but also the Russian GLONASS and eventually the European Galileo. Like the aforementioned augmentation systems, it is based upon base stations at know locations measuring the errors in the received GPS signal and broadcasting this information to compatible GPS receivers. The difference is that the signal is broadcast via satellite, rather than directly.

EGNOS is designed for precision navigation of ships and aircraft. Its expected accuracy is about 2 meters.

5.1.2.5 StarFire

StarFire is a commercial, global, satellite based GPS augmentation system.[7] It was developed by Deere & Company, most known for producing agricultural machines. It is a differential GPS system like the rest, but global. Its accuracy is reported to be about a decimeter.

Note that the StarFire brand also includes an RTK system.

5.2 Inertia based tracking

Inertial tracking uses accelerometers and/or gyroscopes to detect motion.[28] They detect linear and angular acceleration respectively. Accelerometers can also be used to detect angular acceleration, but this requires integration of the result from several accelerometers. Integrating acceleration over time yields velocity, and by integrating once more, distance. The distance reported is the change in position and/or orientation since tracking began.

The main problem with this kind of tracking is that errors add up over time, causing drift. To combat this, the velocity and/or position must be repeatedly corrected using some other data source. Some inertial trackers use a magnetometer to detect the trackers orientation relative to the Earth's magnetic field. The magnetic field is not an entirely accurate data source, but it prevents the orientation values from drifting too much. A gravimeter can also be used as a reference for down.

5.2.1 Angular tracking

Angular tracking generally means gyroscopes. Gyroscopes have been around for a while and are well known. There are three types of gyroscopes: mechanical, optical and MEMS (micro-electro-mechanical systems).

The mechanical gyroscope is what most people think of as a gyroscope. A spinning wheel is mounted on a set of gimbals, allowing for rotation in three dimensions. Conservation of angular momentum keeps the wheel pointing in roughly the same direction as long as it is spinning. When whatever the gimbals are mounted on moves, the gimbals will rotate around their joints while the wheel keeps its orientation. The orientation of the gyroscope's mounting relative to the spinning wheel can be read by sensors located on the gimbal joints. This means that unlike the other gyroscope types, mechanical gyroscopes report orientation.

Optical gyroscopes come in two flavors, but both work on the same principle. One of them is made of a coil of fiber-optic cables. Light enters the cable from both ends. If the sensor rotates, one beam will get a shorter path than the other since the exit is either rotating away from or towards it. This is known as the Sagnac effect and causes an interference which can be measured. The greater the angular velocity, the stronger the interference. The other type of optical gyroscope uses mirrors instead of optic fiber.

MEMS gyroscopes are the newest type. They are small and very simple, with very few parts. They measure the Coriolis effect on some vibrating elements, which is related to the angular velocity of the sensor. The fact that they are small, relatively cheap and use little power, makes them ideal for tracking persons or small devices. They are currently less accurate than the two other types, but this is expected to improve.

5.2.2 Linear tracking

Linear inertial tracking means accelerometers. Linear accelerometers come in three major types: mechanical, solid state and MEMS. Unlike gyroscopes, all accelerometers measure acceleration. This must then be integrated twice to get position. One must remember to subtract the acceleration caused by gravity.

Mechanical accelerometers are again the most straight forward type. A mass is suspended in springs along the axis it measures. When the sensor moves, inertia will cause the mass to lag behind. By measuring how much the mass is displaced in the springs, the acceleration acting on the sensor can be calculated.

There are many different kinds of solid state accelerometers. What seems to be in common is that they measure the effect acceleration has on something that vibrates.

MEMS accelerometers are based on either of the two types above, but are made on a microscopic scale. They are smaller, cheaper and less power consuming. Although not as accurate as the best traditional accelerometers, their accuracy is expected to improve as the technology matures.

5.2.3 Mounting

Inertial tracker assemblies can be mounted to the object being tracked in two ways. One is called stable platform, where the tracker's sensor are mounted on a platform which in turn is mounted on a gimbal with three degrees of freedom. Output from the gyroscopes are used to control motors on the gimbal joints which keep the platform level in the global frame. The angles of rotation on the platform gimbal's joints equals the object's orientation in the global frame. The output from the linear accelerometers are also in the global frame.

The other type is known as a strapdown system. In such a system, the sensor are rigidly attached to the object they track. This means that their outputs are in the local coordinate system of the tracked body. Since the object's position and orientation in the global frame is normally the desired output, the data must be converted into the global frame's coordinate system. This means more computing power is necessary, but the device is simpler mechanically. Computing power is cheap these days and simpler devices are generally cheaper and more robust.

5.3 Active tracking

Active tracking is probably the least suited method. It uses sound, light or electromagnetic fields to track where some device on the user is relative to fixed devices. This means that these fixed devices must be rigged up to surround the site, and the will likely need power as well. While this can be plausible on the top of the hill itself, it makes it impossible to view the hill and the castle from below, unless an even greater area is rigged.

In the interest of having a solution that requires as little modifications of the surroundings as possible and being as mobile as possible, these kinds of trackers have been excluded from further study.

5.4 Passive tracking

Passive tracking is the inverse of active tracking. Whereas in active tracking, external equipment is used to monitor the users movements, passive trackers are mounted on the user. They are typically visual trackers which use features in the surroundings to pin-point the users location and orientation.

A visual tracking system was the first to be developed. It is based around ARToolKitPlus[27], which tracks square black-and-white markers. ARToolKitPlus normally reports a marker's position relative to the camera (or vice versa), but if given information about the markers' true position and orientation in 3D space, it can also calculate the camera's position and orientation in that same space.

The biggest drawback with ARToolKitPlus is that it only detects black-and-white markers. While the interior of the marker can be just about anything, it must be surrounded by a square black border of a certain thickness. This prevents it from using natural features in the surrounding landscape as markers.

Furthermore, ARToolKitPlus requires many markers. When using it for visual tracking, the developers of OpenTracker had to place a marker every two meters.[18] Putting up markers every two meters in two dimensions would require a lot of work and completely clutter up the scene. ARToolKitPlus also gets slower the more unique markers it has to look for. It is possible to reuse makers to some degree based on an estimate of the user's position, as was done by the OpenTracker team, but as there are no walls to restrict movement or visibility, more markers would be visible at the same time. Each marker position would also probably require two to four marker images, since they can be viewed from any direction.

Using the natural features at the site presents two problems. At the actual site, there are very few distinctive local features, except some sections of wall and a well in the southern end. The natural features are also subject to change over the course of the year. In winter, they are mostly covered by snow, while in early spring, the grass is dead. As summer approaches, the grass will grow and turn green. Other visitors will also obscure some of the features. Whether a computer could deal with these changes is unknown.

When standing some distance away from the hill, the hill itself may prove a good feature. However at that distance from the castle, the accuracy need no longer be as great.

Distant features, like buildings and mountains, are more constant, but the errors gotten when calculating the users position from them would become huge if they are far away. Few such features are visible nearby when on the summit, at least when

looking in certain directions, and the summit is where accuracy is most needed. Distant features should work fine for tracking orientation.

5.5 Compasses and tilt sensors

These kinds of sensors provide orientation information only, but are simple and generally cheap. Compasses only report heading in the horizontal plane, while pure tilt sensors report anything but that. Earth's magnetic field is relatively weak, so compasses are not entirely accurate. Magnetic and ferrous objects nearby will have a negative impact on its accuracy.

Chapter 6

Equipment

6.1 Laptop

Figure 6.1: Dell XPS M1710.

The laptop is a Dell XPS M1710 laptop with an Intel Centrino Duo 2.16 GHz processor, 3 GB RAM and an NVIDIA GeForce Go 7950 GTX display adapter with 512 MB dedicated video memory. It has ports for both DVI and VGA, and both can be used simultaneously, but then the laptops built-in display stops working.

Like most modern laptops with the default configuration, this laptop goes into a sleep state and eventually hibernation when the lid is closed. It is important that it is reconfigured to ignore the closing of the lid.

6.2 Headset

Figure 6.2: Trivisio ARvision-3D HMD.

The headset is a Trivisio ARvision-3D HMD, which has two cameras capturing video in 640×480 pixels at 48 fps and two microdisplays capable of 800×600 pixels at up to 100 Hz.[25] It can run on battery power and connects to the laptop using USB for camera input, and two VGA cables for monitor output. An Ophit DDA-A001 DVI-to-VGA adapter is used to connect one of the VGA cables to the laptop's DVI port.

Unfortunately, there are several quirks with the headset solution. Sometimes it fails to initialize properly, causing the image on the display to be completely distorted. At other times, the left (DVI connected) display becomes almost completely white. On the right (VGA connected) display, some odd lines are permanently present. It is also difficult to align the viewports so that they match up with the eyes. A small movement in the headset may make it impossible to see, forcing the user to grab the headset and re-adjust it.

6.3 GPS receivers

6.3.1 Standard GPS

(a) Holux GPSlim 240.

Figure 6.3: Holux GPSlim 240 Wireless Bluetooth GPS Receiver

A Holux GPSlim 240 Wireless Bluetooth GPS Receiver had been purchased to provide GPS tracking data. The SiRFstar III chipset quickly establishes a fix on the position and

is supposed to be able to keep track even in an environment with many obstructions. It can track up to 20 satellites and gives a newly calculated position every second. Note that with the current Global Positioning System, no more than twelve satellites will ever be above the horizon at any given place at any given time. Expected accuracy is 5 to 25 meters, or 2.2 to 5 meters using WAAS/EGNOS. GPSlim 240 does not appear to support the latter. The battery is said to last for eight hours or more. Data is sent in NMEA format at a 38400 baud.[8]

6.3.2 Differential GPS

Figure 6.4: Leica GPS1200 RTK GPS receiver

More accurate positioning was provided by a Leica Geosystems GPS1200 differential GPS/GNSS receiver[12]. It is made up of several components. The core component, to which everything else is connected, is a light gray box. It contains two batteries capable of 8 hours of continuous use. A disc shaped antennae is used to receive the satellite signals, while a GSM-antennae is used to receive corrections from the base station. Both can be fixed to a pole. The entire system is controlled by a remote control that can either be docked directly on the core unit, or connected to it with a cable. A USB-adapter is used to transfer the real-time data to a computer. It appears as a regular COM port on the system. Data is sent in NMEA format at 115200 baud.

When connected to the base station, accuracy is at the centimeter level. The level of accuracy is shown on the remote control and during use, it was generally reported to be between 1 and 3 centimeters. If the base station is unavailable, the GPS1200 performs no better than a good non-differential GPS receiver. The reported error then rises to several meters.

It supposedly tracks both NAVSTAR GPS and GLONASS.

6.4 Inertial tracker

(a) InterSense Wireless InertiaCube3.

Figure 6.5: InterSense Wireless InertiaCube3

6.4.1 Features

An InterSense Wireless InertiaCube3 provides inertial tracking data. It is a small device that runs on a 9-volt battery and can sense orientation around all three axes. It detects angular rate of motion, gravity and Earth's magnetic field.[9] The latter means that it actually works somewhat like a compass. Gravimetric and magnetic readings are used to reduce errors caused by drift.

The InertiaCube3 has gyroscopes built using micro-electro-mechanic systems (MEMS) technology. As detailed in section 5.2.1, this means that there are no moving parts in the normal sense. It is also what makes it possible for the tracker to be as small as it is. The accuracy may be somewhat less than traditional gyroscopes.

External measurements of the sensor unit itself is 31.3 mm $\times$ 43.2 mm $\times$ 14.8 mm.[10] It weighs only 17 grams, which does not burden the HMD significantly.

6.4.2 Mounting

The InertiaCube3 must be mounted on the HMD to be able to track which way the user is looking. Because of the shape of the goggles, it is really only practical to mount it at the top. The InertiaCube3 can be mounted many ways, but some are less favorable than others. To be able to monitor the status of the device, particularly the battery, the LEDs must not face towards the HMD. Mounting the device on its side, so that both the face with the wire and the face with the logo are perpendicular to the vertical axis, will cause problems with some operating modes.

That really only leaves one option, and that is to mount the tracker with the logo facing up. It is currently mounted on the left side of the goggles, with the wire coming out to the right. Because of the magnetic sensors, the InertiaCube3 must not be placed close to ferrous or magnetic objects. It has been calibrated to compensate for any static effects the headset may have on the magnetic field.

The receiver, which plugs into a USB port via a long cable, can be kept in the backpack.

6.4.3 Coordinate system

The Wireless IntertiaCube3 uses a right-handed coordinate system. Positive x is opposite of the cable, positive z is opposite of the logo. Rotation about the x-axis is called *roll*, rotation about the y-axis *pitch*, and rotation about the z-axis *yaw*. The API reports rotations as yaw, pitch and roll, in that order.

Placing the tracker with the logo up and the wire going to the right maps the rotations to the following axes in the virtual world: Roll is about negative x, pitch is about positive y, and yaw is about negative z.

The InertiaCube3-series knows what is up and what is down by sensing gravity. It can also remember a direction in the horizontal plane, although this can be reset thought the API and driver control software. With the logo up, holding the tracker level gives zero roll and zero pitch, which is intuitive.

Chapter 7

Experiments

7.1 Initial experiments

7.1.1 Campus trials

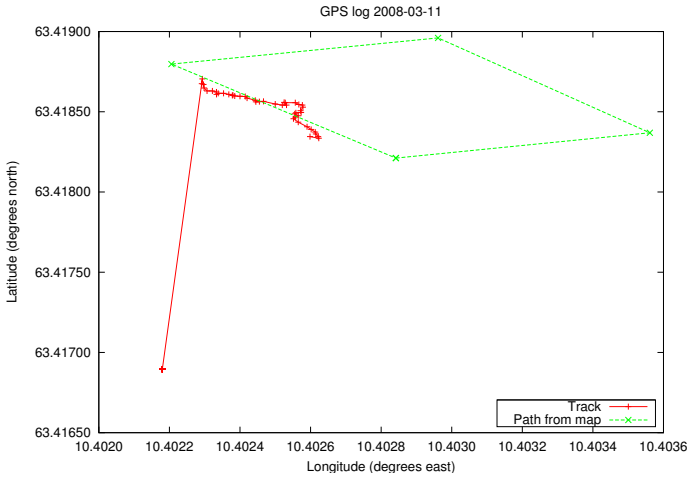
While the GPS receiver's manual gives an estimate of its accuracy, these numbers are very general. It is therefore nice to see how the receiver will perform under more specific circumstances. As an initial experiment, the GPS and the laptop was carried around a rectangular open space at the campus. The area is about thirty meters wide and about sixty meters long. This is far larger than the best-case GPS accuracy range, and even larger than an average accuracy range. Coordinates from the GPS receiver were logged to a file for later analysis, and because it is virtually impossible to see anything on the laptop display when outside.

For the first test run, the GPS did not pick up any satellites at all. It therefor became clear that the GPS will not magically just work. The receiver was carried around in a pocket on the backpack, close to the laptop. Apparently, the laptop, and possibly also the backpack and wearer, was blocking or otherwise disturbing the reception.

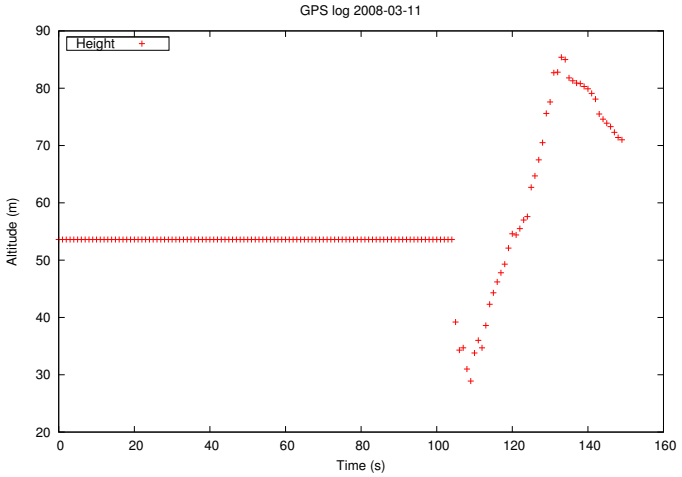
For the second experiment, the GPS was placed on top of the backpack where it should get a more clearer reception. The site remained the same, as it is the only large open space at the university. A new logger was used, which logged all data from the GPS receiver and not just the position. This time, the GPS did pick up satellites and report coordinates which changed over time.

Unfortunately, it took about 100 seconds before the receiver picked up some satellites and then it only picked up four. This appears to not be enough to get the required accuracy. The plotted coordinates (figure 7.1a) look nothing like the path walked, which was roughly towards east, north, west and finally south. Altitude varies by more than 50 meters (figure 7.1b), even though the area was flat and level.

This time, the only real obstructions were the buildings and trees surrounding the open space. The buildings are mostly made of solid materials, like stone, which block radio waves and might also reflect them. The head of the person wearing the backpack might also be a hindrance to the signals, but certainly to a lesser degree. That the technology behind this receiver works well in an urban jungle, as promised in some reviews, seems less likely.



(a) First 100 coordinates are on top of eachother in the lower left. The diamond shaped figure marks the actual path walked.



(b) Altitude above mean sea level. Should have been constant.

Figure 7.1: Position and altitude as reported by the GPS.

7.1.2 On-site trial

The next experiment was done at the actual castle site at Sverresborg. Apart from the user, there was nothing preventing the GPS receiver from seeing all the way to the horizon. Just a short walk was made back and forth along the inner wall of the stone building. 9 satellites were picked up straight away, and this number remained constant throughout the short experiment.

The results of the experiment are presented in figure 7.2. As one can see, the plotted coordinates now form a much straighter line, which actually follows itself pretty closely on the return trip. The point to the lower left is the ending point. It may have taken some time before the GPS receiver started sending coordinates, which would explain why the starting point is much further up and to the right.

Similar with the height graph. The position is lower at the start than at the end. Unlike the previous experiment, where the height varied with more than fifty meters, this data only varies with 3 meter. The terrain was not flat this time, and without having measured it, 3 meters difference in height between the start/end and the turning point seems right. There are still some random jumps in height of 10 cm or more (at 12 seconds) which can not be explained by the terrain. It should, however, be possible to filter out such noise.

7.2 First test run

For the next visit to Sverresborg, the inertial and GPS tracking systems were integrated into ZionARA. A bug it the parsing of the height data from the GPS caused the castle to hover 174 meters above the site. To track down this bug at the site could take time and the laptop was running on batteries, so a quick fix was to ignore the height. This caused the castle to hover only a few meters above ground and enabled evaluation of the horizontal tracking.

As expected, the castle moved in sudden jumps once per second. This is because the GPS receiver only sends updates once per second and no interpolator had been implemented in ZionARA. It was also discovered that the axes of the virtual world were not aligned as expected. More about this is explained in section 8.2.1.

It also became clear that some system is necessary to calibrate the system to north. This could be a simple compass which the user could use to align him-/herself towards north and then zero out the heading reported by the inertial tracker. An automated system would naturally be more user friendly.

7.3 GPS replay experiments

To avoid going back and forth between the university and Sverresborg all the time, a simple GPS log replay system was developed. This system is described in more detail in section 8.4.2. The log from the first on-site experiment (section 7.1.2) contains a walk along the inner wall of the main building and so would provide a clue to the accuracy of the system so far.

Unfortunately, when replaying the walk in ZionARA, the virtual camera moved several meters north of the wall, a few meters below ground and possibly some distance to the west. The latter is harder to tell since the extreme positions in the log do not exactly correspond to the corners of the building, nor is the wall aligned fully east-west. This raises some questions. Is this offset caused by GPS inaccuracy? Are the

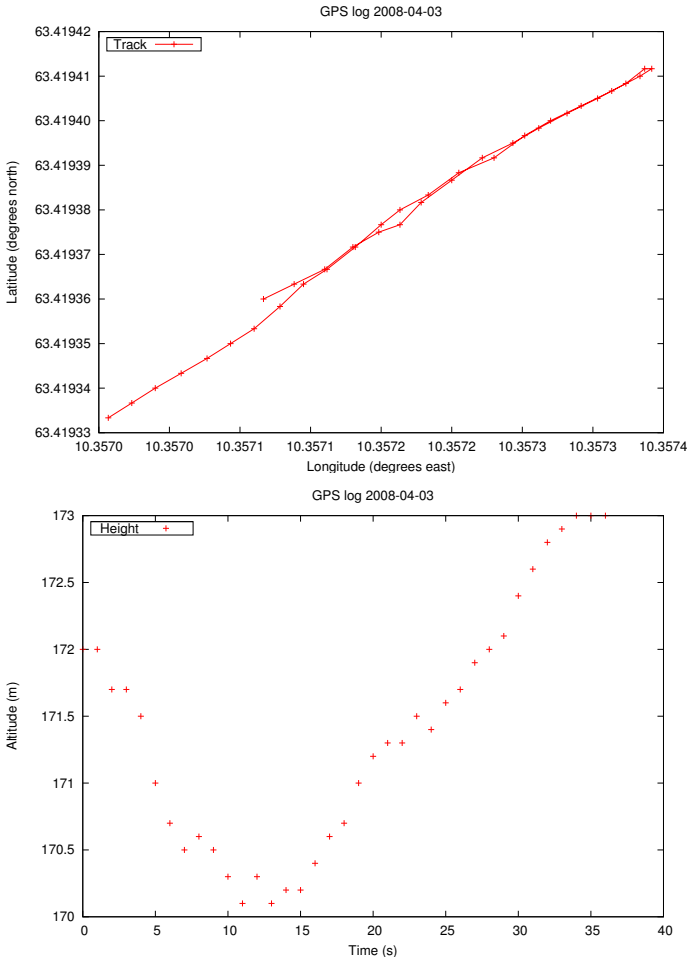


Figure 7.2: Position and altitude as reported by the GPS.

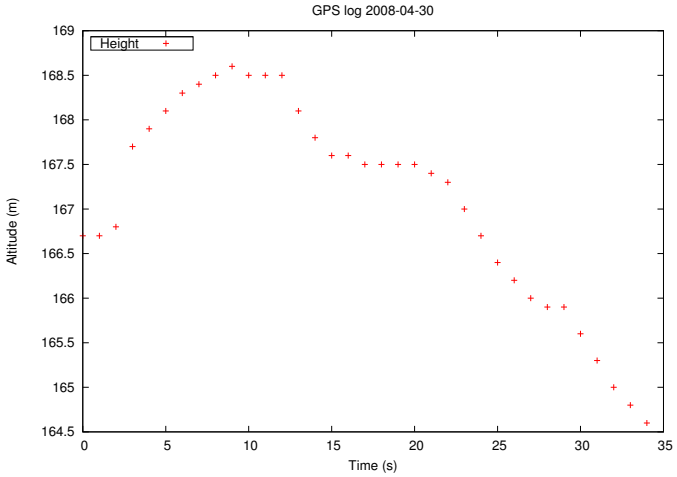


Figure 7.3: Height plot from the GPS log of a walk through the gate.

coordinates of the virtual world reference point correct? The errors are clearly within what one might expect of an unaugmented Global Navigation Satellite System. There are also no references to how the reference point coordinates were found. They may come from a map or by trying to find the spot in the real world and reading the coordinates of a GPS. If the latter is true, then there are two sources for GPS inaccuracy errors.

A new log was recorded of a walk through the gate from the inside out. When playing it back, it once again turned out that the positions were below ground level in the virtual model. Although the previous walk along the wall and this walk through the gate intersect each other, the heights reported this time were much lower (compare figure 7.2 with figure 7.3).

This basically proves it that GPS alone is not accurate enough to track the user in an augmented reality application. Relative movement appears much more accurate, but something is needed to get the GPS receiver onto the right track from the start.

7.4 Differential GPS

7.4.1 Campus trial

Since normal GPS receivers were found to be inadequate, a differential GPS receiver was borrowed. Once connected to the laptop, the first experiment was repeated.

As can be seen in figure 7.4, the log from the differential GPS receiver more closely matches the actual path walked. There are still some errors and times when the GPS receiver completely loses track of the user. The error shown on the remote rose to several meters on these occasions. There are also some fluctuations in height, but they are much lower this time. They also match up with when there were problems getting horizontal position.

When remembering that the ordinary GPS got no fixes at all for most of the walk on campus, but had no such problems at Sverresborg, this is not so bad. There are much fewer obstacles at Sverresborg.

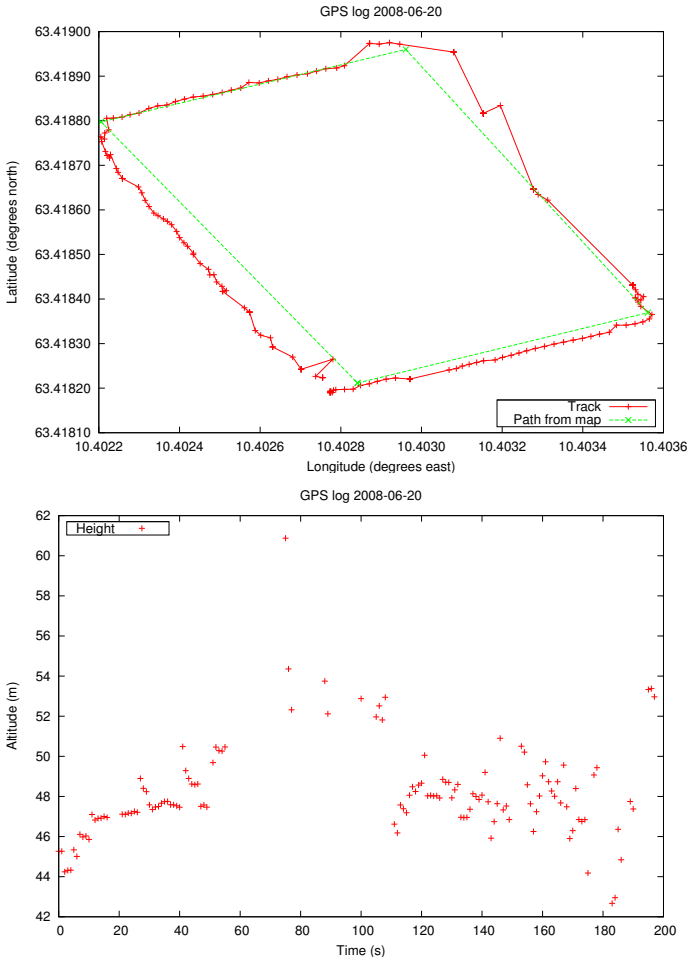


Figure 7.4: Position and height as reported by Leica's differential GPS receiver.

7.4.2 At Sverresborg

After getting the system to work at the castle site at Sverresborg, the system gave good tracking data. It was however discovered that the castle was not positioned where it should be. A new log was captured of a walk along the inner wall of the main building and out through the gate. This log was then imported into the Blender model. The path matched the virtual model in terms of scale and orientation, but was offset by several meters. By translating the path 8 meters along the x-axis, -25 meters along the y-axis and 8 meters along the z-axis, the path matched up with the virtual walls and the terrain height. When returning to Sverresborg, the castle appeared at the correct site.

New problems appeared however. The castle would at times stop "moving" along with the terrain when the user moved. At first, the virtual castle would catch up in less than a minute if the user stood still. Later, a reboot of ZionARA was necessary. It is unknown if this was caused by a bug in ZionARA or something with Leica's equipment. Debugging of the entire system is almost impossible. The laptop is meant to stay in the backpack, and even if one can take it out, untangle the cables and open it, the sunlight makes it very difficult to see what is on screen. While driving the headset, the laptop's own display is blank, so ZionARA must be rebooted to run in a window first anyway. When indoors, making it possible to see what is on the screen, the GPS receiver will not work. Remote debugging would be possible, but there was no time to get the equipment and configure it. Weeding out the bugs will have to wait.

Another problem was the way the InertiaCube was mounted. The velcro allows for some movement, causing the InertiaCube to come out of alignment. Just a few degrees is enough to shift distant objects several meters. As the battery is not attached to anything but the wire connecting it to the tracker, it will dangle by the wire every time the HMD is handled, pulling on the tracker.

Chapter 8

Solution

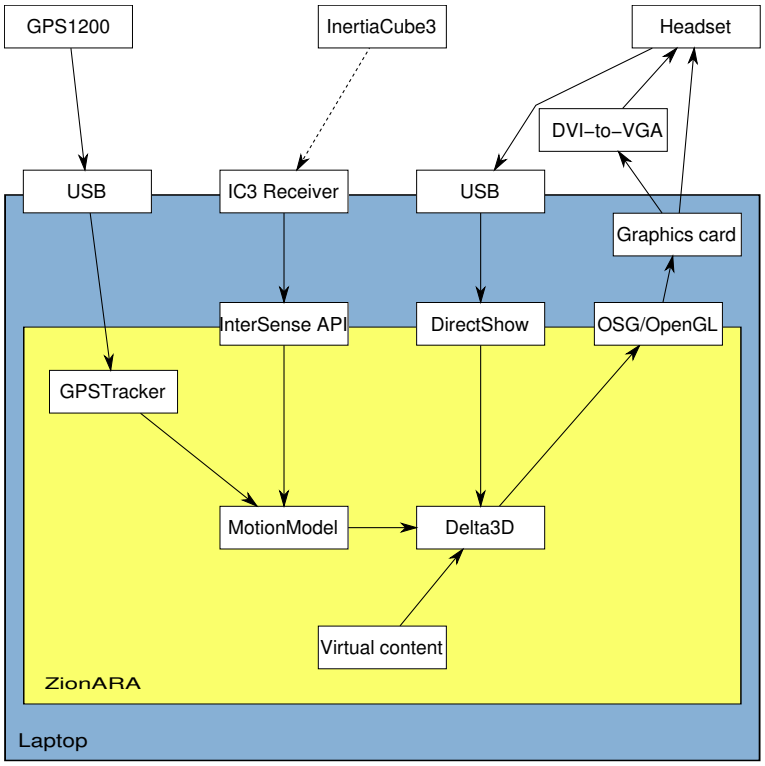


Figure 8.1: Overview of the ZionARA software and hardware architecture.

8.1 Rendering system

ZionARA, as developed by Johannes Odland, did stereoscopic rendering by rendering the view for each eye into the respective half of a vertically split window. The window did not fill the screen, which makes it possible to see the debug console during testing. When using the HMD, this is not ideal. The HMD is driven as two separate displays, and while it is possible to resize the window to full them both, it is not easy.

	Latitude	Longitude
North	111468.30	49937.23
South	111468.20	49948.57

Table 8.1: Lengths, in meters, of a degree of latitude and longitude at the extreme latitudes of Sverresborg museum.

ZionARA has therefor been given another rendering mode in addition to monoscopic and split-window stereoscopic: dual screen stereoscopic rendering. This mode creates two windows, positions one on each display and makes them fill the screen. Each camera, both virtual and real, are then assigned to the correct window. For this to work, the operating system's display settings must be configured to place the primary screen at the output connected to the left display and the secondary screen at the output connected to the right display. The right display must come immediately to the right of the left display. Both screens must run at the HMD's native resolution of 800×600 pixels.

The rendering mode to use can be set in the main configuration file. Monoscopic is still the default mode if no other valid mode is specified.

8.2 Tracking systems

ZionARA no longer uses OpenTracker to communicate with the tracking devices. It did not provide anything useful to the application, as the three tracking devices can easily be used directly by the code. OpenTracker also had no built in way of combining the same kind of input from multiple sources, making it necessary to have at least two OpenTracker graphs. The fact that it converted all rotations to quaternions, whether originally reported as a matrix or Euler angles, made it difficult to use rotations from a different coordinate system, especially when it comes to the InertiaCube3.

8.2.1 GPS tracking

The GPS returns the users position using latitude, longitude and height above mean sea level. This is a spherical coordinate system with the center of the Earth as its origin. Since the rest of ZionARA uses Cartesian coordinate systems, some conversion has to be used. As the Earth is approximately flat on the scale ZionARA is meant to operate, a simple linear conversion should work.

The original definition of the meter as 1/10,000,000 of the distance from the equator to the poles, means that one degree of latitude equals approximately 111,111.11 meters at sea level. This value should not be much different 170 meters above sea level, where Sverresborg is located. The longitude is different. At the equator, one degree of longitude also equals 111,111.11 meters, but this value decreases as the meridians converge towards the poles. The height above sea level should be the same and require no conversion. Unfortunately, these values do not take into account that the Earth is an ellipsoid, meaning that the radius is not the same at the equator and at the poles.

The northern end of the museum area is at about 63.4235635 degrees north, while the southern end is at about 63.4170412 degrees north.[22] Using a calculator[6] available on the Internet, this yields the metric lengths for a degree of latitude and longitude presented in table 8.1. Using a average of these as an approximation should yield a mi-

croscopic error in north-south direction and an error in the fourth decimal for east-west. This should be an acceptable error, at least compared to the errors in the coordinates reported by the GPS.

The 3D-model is created with its positive y-axis pointing north, the positive x-axis pointing east and the positive z-axis pointing up, but for some reason it is rotated 180 degrees in ZionARA so that positive y-axis points south and positive x-axis points west. Odland reports the origin of the model, or the reference point, to be at 63.419253333 degrees north, 10.357181667 degrees east and 174.100 meters above sea-level.[15] Simply subtracting these values from the coordinates reported by the GPS and linearly converting the degrees into meters, should give the user location relative to the reference point.

After getting a differential GPS receiver, the reference point was found to off be several meters. Simply adding $\begin{pmatrix} 8, & -25, & 8 \end{pmatrix}$ to the coordinates fixes this error.

Final coordinate system conversion

$$x = (latitude - 63.419253333) \times \frac{111468.30 + 111468.20}{2} + 8$$

$$y = (longitude - 10.357181667) \times \frac{49937.23 + 49948.57}{2} - 25$$

$$z = elevation - 174.100 + 8$$

In addition, the entire world is rotated 180 degrees around z after applying camera rotation.

8.2.2 Inertial tracking

The InterSense Wireless InertiaCube3 has an easy to use API which makes programming against it easy. This API is the same for a whole range of InterSense trackers, so ZionARA is not bound exclusively to the InertiaCube3 model.

To use the tracker, three different API calls are necessary: One to open the device, one to read the yaw, pitch and roll, and one to close the device. The first and the last API calls are made during application initialization and shutdown respectively. The read calls are made prior to each frame being rendered. Yaw, pitch and roll are then combined into a rotation matrix. With the tracker correctly calibrated, this rotation matrix represents the users orientation relative to a coordinate system where positive y-axis is north, positive x-axis is east and positive z-axis is up.

8.2.3 Visual tracking

Prior to this project, ZionARA only had a visual tracking system, implemented using ARToolKitPlus[27]. This system is still present and has been developed a bit further for testing purposes, but is currently unused. The original system simply detected a marker and used that as the reference point for positioning the virtual world.

Currently, the system is capable of detecting several markers. On initialization, ARToolKitPlus is given a list defining the markers and how they are rotated and positioned relative to the real world reference point. When fed with images from one of the cameras in the headset, ARToolKitPlus will scan through them for markers. If a marker is found, it can then calculate the relative position between the marker and the

camera. Knowing the relative position between one or more markers and the camera, and the relative position between the marker(s) and the reference point, it is possible to calculate the position of the camera, as well as its orientation, relative to the reference point.

The visual tracking system is more or less vestigial. No practical use of it in the final Zion Augmented Reality Application is currently planned. It may still be useful for testing parts of ZionARA not related to tracking.

8.2.4 Merging the data

The rotation matrix from the inertia tracker (**R**) is then combined with data from the GPS (x, y and z) and then multiplied with a matrix representing a 180 degree rotation around the z-axis to match the coordinate system used for rendering. This yields the final transform (**T**) applied to the virtual camera setup.

$$\mathbf{T} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{R}(yaw, pitch, roll)$$

When using visual tracking, both the GPS and inertia trackers are ignored, but the data still needs some conversion. Here **A** is the data from ARToolKitPlus.

$$\mathbf{T} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{A}^{-1}$$

8.3 Headset calibration

8.3.1 Exposure

The light conditions indoors and outdoors are very different, and both vary with the weather. Early on, while development was taking place inside, the headset worked fine. This changed drastically when first using the headset outside. It was a sunny day, and the images captured by the cameras contained almost nothing but white.

It turned out that the cameras were getting overexposed because the exposure setting was set for regular indoors illumination. Even with some indirect sunshine in the office, the cameras were getting slightly overexposed. Fortunately the cameras have an auto-exposure feature. After enabling that, the headset captured good images both outside in the sun and inside the office.

This solution is adequate for now, but when the time comes for implementing virtual illumination that matches the real light conditions, other solutions may be necessary. As it is now, the brightness of the real world backdrop may go up an down, while the foreground virtual castle is always rendered at the same brightness. How to fix this is however beyond the scope of this report.

8.3.2 Viewports

The convergence of the cameras has been adjusted so that they are parallel and pointing straight forward. The transformation matrices for their virtual equivalents are set up

accordingly. Unfortunately, the cameras are not properly aligned vertically, with the right pointing slightly lower than the left. There is no way of adjusting this short of trying to bend them into place or picking apart the headset. This does not seem to be much of a problem and the headset might be replaced anyway. There are also something else going on, since calibration yields two different projection matrices for the two cameras. The difference is not that big, though.

8.4 External helper programs

In addition to ZionARA, a few other applications were also developed to assist the research and development.

8.4.1 GPS logger

A really simple program which logs all data coming from the GPS to a file, as well as listing them in real time. The main motivation for creating this program, as other GPS logging programs no doubt exist, was to be able to start and stop logging with the laptop shut and placed in the backpack. At the time, the only hand-held input device available was a standard optic computer mouse. Since the cursor could drift away from any start or stop buttons while the computer is tucked away, the GPS logger application responds to mouse clicks wherever they happen. It does not even have to be in focus, which it possibly will not be after the first click. The right mouse button starts and stops the logging, since it is the least harmful when performed outside the window.

8.4.2 GPS emulators

Since GPS reception is poor at the campus because of the tall buildings, it is necessary to go somewhere else to test out the GPS tracking system. If something is wrong, one must then go back to campus to fix it. This can become tedious. To be able to try and move around at Sverresborg using the GPS tracking code in ZionARA without actually going to Sverresborg, a set of fake GPS applications were developed. Since ZionARA communicates with the GPS using basic operating system I/O operations on a COM port, only minor changes were needed to make it communicate with a named pipe.

The fake GPS application sits at the other end of the pipe. It emulates the GPS by sending the GPS commands necessary for the operation of ZionARA. One of the programs simply lets the user enter latitude, longitude and height. It then wraps them up in a fake GPS sentence and writes it to the pipe. ZionARA interprets the sentence as valid and extracts the coordinates as if they came from a GPS.

The other program reads in a file which contain a log of sentences possibly received from a GPS. When the user hits play, the program starts streaming the GPS sentences to the pipe. Every time it reaches a GPGGA sentence, which is what ZionARA uses to read the coordinates, it pauses for a second to simulate the rate at which the GPS produces coordinate updates.

Figure 8.2: The packed backpack ready for use.

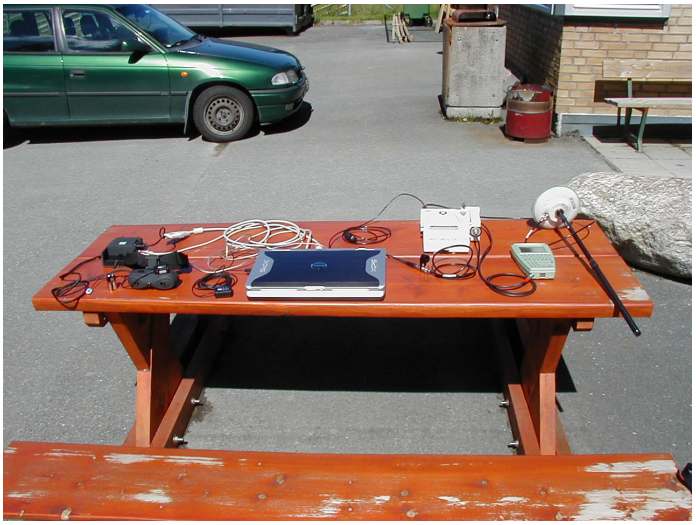


Figure 8.3: All the equipment connected together, but not packed. From left: HMD with Wireless IntertiaCube3 and battery, Wireless InertiaCube3 receiver, laptop, GSM antennae, DGPS core unit, DGPS remote control and GNSS antennae.

Chapter 9

Conclusion

These experiments show that non-differential Global Navigation Satellite Systems are in practice useless for mixed reality where the user is close to the virtual objects. Without the ability to compensate for ionospheric delays, ephemeris errors and clock errors, the reported coordinates will inevitably be several meters off. That is simply unacceptable.

Differential GNSS, and especially real-time kinematic versions, are capable of providing the required accuracy for mixed reality. The equipment used was meant for surveying and is a bit bulky and unwieldy to operate, but that is a minor drawback with the particular receiver used. If an augmented reality system enters production, it would be able to use a custom built receiver that integrates better into the overall system.

Modern laptops are powerful enough and small enough to make mobile augmented reality possible. The frame rate is good, although there is some latency. A noticeable amount of time passes from the time the images are captured by the cameras until they are presented to the user. This may feel unpleasant to the user.

Battery life is a concern. Running ZionARA at full power quickly drains the laptop's batteries. A single tour of the site drains them down to about 50%. One must then spend an hour recharging them before one can be certain that they will last another tour. The HMD's battery seems to last about just as long, but without an indicator telling how much power remains, this is more of a guess. Some sort of unified power supply capable of providing power to all the different devices would be a lot easier to manage.

Although the exact weight is unknown, and may change as equipment is changed, it is well within the carrying capacity of an adult human and older children. It is probably around 10 kilograms.

Chapter 10

Further work

Although the tracking system works, further tweaks may be necessary to get a perfect match between the virtual geometry and the real geography. There are also some problems with getting a correct heading. The InertiaCube must be re-calibrated to north every now and then, possibly because the Earth's magnetic field fluctuates a bit. It is also not firmly enough attached to the headset due to the ad hoc nature of the experimental equipment and may easily come out of alignment when handled. While the error in the matching of the two worlds is small in terms of degrees, just a degree or two becomes several meters some distance away from the viewer.

The method currently used to combine the real and the virtual world is very simple. When rendering, the image data from the cameras is used as a backdrop. All virtual objects are drawn on top. This means that all virtual geometry will appear in front of all real world geography and objects, which was noted as odd or not right by museum staff during a demonstration. For a convincing mixing of realities, the virtual world must blend *into* the real world, not just sit on top of it. Real world objects must obscure or be obscured by the virtual objects depending on which is closer to the viewer. This means that ZionARA must learn how to see the world and figure out the distance to every object. As the system already has stereo vision with moving pictures, most algorithms for depth estimation are possible. Once the depth of each pixel has been found, these depths can be written to the z-buffer before rendering the virtual geometry. The z-buffer algorithm[5] will then take care of making sure virtual objects do not obscure real world objects closer to the viewer.

To achieve full integration between the virtual and the real world, the two worlds must be equally illuminated. Since controlling the real world illumination outdoors, it is the virtual illumination which must be set up to match the real world light conditions. This will change over time as the sun travels across the sky and clouds pass overhead.

Nomenclature

COM port Originally a name for the RS-232 serial port on IBM PCs. ZionARA only deals with virtual COM ports, which are serial communication channels emulating RS-232 ports.

DGPS Differential Global Positioning System. GPS or GNSS corrected for errors by using a reference station.

fps Frames per second

GNSS Global Navigation Satellite System

GPS Global Positioning System. The GNSS developed by the US Military and the most well known.

HMD Head Mounted Display

NMEA National Marine Electronics Association. In this document, and in most GPS related situations, it is synonymous with the NMEA 0183 standard for a whole range of marine electonic devices, from sonars to GPS receivers.

RTK Real-time kinematic

Bibliography

[1] European Space Agency. What is EGNOS? http://www.esa.int/esaNA/GGG63950NDC_egnos_0.html (retrieved 2008-03-05), July 2007.

[2] ARQuake FAQ. http://wearables.unisa.edu.au/projects/ARQuake/www/faq/index.html (retrieved 2008-01-29).

[3] Ronald T. Azuma. The challenge of making augmented reality work outdoors. In Yuichi Ohta and Hideyuki Tamura, editors, *Mixed Reality: Merging Real and Virtual Worlds*, pages 379–390. Springer-Verlag, 1999.

[4] Glenn Baddeley. GPS - NMEA sentence information. http://home.pacific.net.au/~gnb/gps/nmea.html (retrieved 2008-05-20), June 2007.

[5] Edwin E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. PhD thesis, Dept. of CS, U. of Utah, December 1974.

[6] Computer Support Group. Length Of A Degree Of Latitude And Longitude Calculator. http://www.csgnetwork.com/degreelenllavcalc.html (retrieved 2008-04-09).

[7] Kevin Dixon. StarFire^{TM}: A Global SBAS for Sub-Decimeter Precise Point Positioning. Technical report, NavCom Technology Inc., January 2007. Available at http://www.navcomtech.com/Support/Download/StarFireGlobalSBASforSubDecimeterPrecisePointPositioning.pdf (retrieved 2008-03-05).

[8] Holux. *HOLUX GPSlim240 Wireless Bluetooth GPS Receiver User Guide*, 2006. Available at http://www.holux.com/JCore/en/products/products_download.jsp?pno=253 (retrieved 2008-15-19).

[9] InterSense, Inc. *Product Manual for use with InertiaCube3 and InertiaCube3 Processor*, 2005.

[10] InterSense, Inc. *Supplemental Product Manual for use with Wireless InertiaCube3 Serial and USB Interfaces*, 2005.

[11] P. F. Lammertsma. Satellite Navigation. Technical report, Institute of Information and Computing Sciences, Utrecht University, February 2005. Available at http://paul.luminos.nl/documents/show_document.php?d=6 (retrieved 2008-03-04).

[12] Leica Geosystems. Leica GPS1200 product page. http://www.leica-geosystems.com/corporate/en/products/total_stations/lgs_4521.htm (retrieved 2008-06-30).

[13] Chris Hill Luís Sardinha Monteiro, Terry Moore. What is the accuracy of DGPS? *The Journal of Navigation*, pages 107–225, 2005.

[14] Johannes Odland. Towards augmented reality for field use. Project report at the Norwegian University of Science and Technology, 2006.

[15] Johannes Odland. Zion Augmented Reality Application. Master's thesis, Norwegian University of Science and Technology, 2007.

[16] Wayne Piekarski, Ross Smith, and Bruce H. Thomas. Designing backpacks for high fidelity mobile outdoor augmented reality. In *ISMAR '04: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 280–281, Washington, DC, USA, 2004. IEEE Computer Society.

[17] Wayne Piekarski and Bruce Thomas. ARQuake: the outdoor augmented reality gaming system. *Communications of the ACM*, 45(1):36–38, 2002. Also available at http://wearables.unisa.edu.au/arquake/papers/index.html (retrieved 2008-03-31).

[18] Gerhard Reitmayr and Dieter Schmalstieg. OpenTracker: A flexible software design for three-dimensional interaction. *Virtual Real.*, 9(1):79–92, 2005. Also available at http://studierstube.icg.tu-graz.ac.at/opentracker/Documents/opentrackervrst2001final.pdf.

[19] Gethin Wyn Roberts, Andrew Evans, Robin Hollands, Bryan Denby, Simon Cooper, and Alan Dodson. Look Beneath the Surface with Augmented Reality. *GPS World*, February 2002.

[20] G.W. Roberts, A.J. Evans, A.H. Dodson, B. Denby, S.J. Cooper, and R.J. Hollands. The use of augmented reality, GPS and INS for subsurface data visualisation. In *XXII International Congress of the FIG, Washington DC, USA*, 2002.

[21] Septentrio. DGPS vs RTK. http://www.septentrio.com/about_dgps_rtk.htm (retrieved 2008-03-04).

[22] Statens Kartverk. Norgesglasset. http://ngis2.statkart.no/norgesglasset/default.html.

[23] Tor Egil Riegels Strand. Further work on Augmented Reality at Sverresborg. Project report at the Norwegian University of Science and Technology, 2007.

[24] Bruce Thomas, Ben Close, John Donoghue, John Squires, Phillip de Bondi, Michael Morris, and Wayne Piekarski. Arquake: An outdoor/indoor augmented reality first person application. In *ISWC '00: Proceedings of the 4th IEEE International Symposium on Wearable Computers*, page 139, Washington, DC, USA, 2000. IEEE Computer Society.

[25] Trivisio Prototyping GmbH. *User Manual ARvision-3D*.

[26] United States Air Force. GPS Facts Sheet. http://www.losangeles.af.mil/library/factsheets/factsheet.asp?id=5325 (retrieved 2008-03-04).

[27] Daniel Wagner and Dieter Schmalstieg. ARToolKitPlus for pose tracking on mobile devices. In *Proceedings of 12th Computer Vision Winter Workshop 2007*. Graz Technical University, February 2007. Available at http://www.icg.tugraz.at/Members/daniel/Publications/ARToolKitPlus.

[28] Oliver J. Woodman. An introduction to inertial navigation. Technical Report 696, University of Cambridge, August 2007. Available at http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf (retrieved 2008-05-22).

Appendix A

GPS messages

The GPS receivers sends several data messages[4] to the listening application, in this instance ZionARA. The Holux GPSlim 240 sends the following subset of messages specified in the NMEA 0183 standard:

GPGGA Global Positioning System Fix Data. Sent once every second. This is the only sentence parsed by ZionARA. See section A.1.

GPGSA Dilution of precision. Sent once every five seconds.

GPGSV Satellites in view. Sent every five seconds as a group of GSV-sentences each detailing up to four satellites.

GPRMC Recommended minimum specific GPS/Transit data. Contains just about everything, except elevation. Sent once every second.

GPVTG Track Made Good and Ground Speed. Sent once every second.

Leica's DGPS system offers an even greater range of messages, some of which may be unique to their equipment or even just that model. The unit used here was configured to start the messages with *GN* instead of *GP*. This is why ZionARA has been programmed to accept both.

A.1 GGA

```
$GPGGA,111537.000,6325.1467,N,01021.4417,E,1,09,0.9,165.0,M,41.5,M,,0000*5C
```

Figure A.1: Example GGA sentence from one of the GPS logs.

The GGA sentence (figure A.1) is the only one parsed by ZionARA and the only one generated by the GPS emulator. A GGA sentence is made up of 15 parts separated by commas, including the GGA prefix, and a checksum separated from the rest by an asterisk. The elements are:

1. Prefix which identifies the type of sentence. The initial *GP*-part is used by standard GPS messages, but can be configured to be something else.
2. Time. Two digits for hour, two for minutes and two digits plus fraction for seconds.
3. Latitude. Two digits for degrees and two digits plus fraction for arcminutes. Parsed.
4. Hemisphere. “N” for northern and “S” for southern. Not parsed. Assumed to be “N”.
5. Longitude. Three digits for degrees and two digits plus fraction for arcminutes. Parsed.
6. Hemisphere. “E” for eastern or “W” for western. Not parsed. Assumed to be “E”.
7. Quality. 0 for no fix, 1 for normal GPS fix, and 2 for differential GPS fix.
8. Number of satellites used to get fix.
9. Horizontal dilution of precision.
10. Altitude above mean sea level.
11. Units used for altitude. Seems to be fixed at “M” for meters.
12. Geoidal separation.
13. Units used for geoidal separation. Seems to be fixed at “M” for meters.
14. Seconds since last update from differential GPS station (if any).
15. ID of differential GPS station (if any).
16. Checksum. Ignored.

Appendix B

GPS coordinate extraction scripts

These are the scripts used to transform the GPS logs into data files used to create the diagrams used in the report.

B.1 Latitude and longitude extraction

```
#!/usr/bin/python

import sys
from decimal import *

for line in sys.stdin:
  if line[1:6] == "GPGGA" or line[1:6] == "GNGGA":
    elements = line.split(",")
    if elements[2]:
      la_deg = Decimal(elements[2][0:2])
      la_min = Decimal(elements[2][2:])
      la_dec = la_deg + la_min / Decimal("60.0")

    if elements[4]:
      lo_deg = Decimal(elements[4][0:3])
      lo_min = Decimal(elements[4][3:])
      lo_dec = lo_deg + lo_min / Decimal("60.0")

    sys.stdout.write(str(la_dec) + " " + str(lo_dec) + "\n")
```

Reads GPS data from standard input and writes lines of space separated latitude and longitude pairs as decimal degrees to standard output.

B.2 Height extraction

```
#!/usr/bin/python

import sys

i = 0
for line in sys.stdin:
  if line[1:6] == "GPGGA" or line[1:6] == "GNGGA":
    elements = line.split(",")

    sys.stdout.write(str(i) + " " + elements[9] + "\n")
    i = i + 1
```

Reads GPS data from standard input. Each line written on standard output contains a seconds since start counter and, separated by a space, the height above mean sea level.

Appendix C

Contents on the disc

- master.pdf - This report
- ZionARA/
  - ZionARA.sln - Microsoft Visual Studio 2005 solution
  - ZionARA.vcproj - Microsoft Visual Studio 2005 project for ZionARA
  - config/ - Configuration files for ZionARA
  - data/ - Textures and mesh models
  - FakeGPS/ - Source code for GPS emulator
  - GPSLogger/ - Source code for GPS logger
  - GPSPlayback/ - Source code for GPS log player
  - include/ - Header files for ZionARA
    - zara/ - Miscellaneous headers
    - zaraConf/ - Headers related to the configuration system
    - zaraTrac/ - Headers related to the trackers
  - src/ - Source files for ZionARA
    - zara/ - Core and rendering
    - zaraConf/ - Code for reading configuration files
    - zaraTrac/ - Modules interfacing with the tracking systems
- logs/ - The raw GPS logs referred to in this report

Appendix D

Dependencies

No compiled binaries are provided as they only work with the laptop and other equipment used in the project, and they already lie there. To compile new versions of ZionARA, the following dependencies must be installed:

- Delta3D (sub-dependencies are bundled with binary releases)
- ARToolKitPlus (for vestigial visual tracker)
- InterSense software (runtime dependency)
- Microsoft DirectShow baseclasses (part of Windows SDK, but must be compiled)