UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Beach-Volumes:
Design of Software & Hardware

Miguel Alexandre Vaz Cordeiro

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
(2º ciclo de estudos)

Orientador: Prof. Doutor Rui Manuel da Silva Fernandes

Covilhã, Junho 2014

Beach-Volumes: Design of Software & Hardware

To all the people who contributed to this work.

Acknowledgement

I see this project as the conclusion of another cycle that, although individual had the support of several people. I want to express my gratitude to all of them.

First of all, I would like to thank my family, especially my parents, for all the support they gave me during this stage of my life, and the pride they feel for me.

A big thanks to my girlfriend for having followed me in this stage of my life and for understanding, supporting and helping to overcome all the difficulties I had in this path.

I also would like to thank my dissertation advisor, Professor Rui Manuel da Silva Fernandes, for his time availability and for the information and advices he transmitted, allowing me to get the knowledge and skills necessary to successfully complete the project's work plan. I also would like to thank the project SingSubs - Monitoring of Subsidence in Singapore, a collaboration with Tropical Marine Scientific Institute (TMS), part of the National University of Singapore, for the research grant they provided me, allowing the development of this project in the last year.

Finally, I want to thank all my friends and laboratory colleagues especially, João Apolinário and Machiel Bos for the help they have given me over this year. You know, in one way or another, are a part of this work.

Thank you!!!

Abstract

The main objective of the research work described in this manuscript was to develop methods (hardware/software) for three-dimensional data acquisition in order to perform morphological analysis in sandy shore environments. This system, when concluded, intends to estimate the volumetric variations in beaches being eroded at high rates, directly contributing for studies focused on the protection of the coastline. This work is particularly focused on the development of software to estimate the 3D positions using a dedicated low-cost hardware device that was also developed in the framework of this project.

Several methods have been used to study these phenomena, such as classical topography by surveying discrete points with the utilization of theodolites and/or total stations that are highly time-consuming. Other methods that start recently being used, like terrestrial or airborne LiDAR, are very costly and are not suitable to be used in many locations due to governmental requirements (like permits, environment, and security issues). Similar limitations also occur with Mobile Mapping Systems (MMS) with laser scanners that permit the survey when the vehicle is moving on the beach.

In order to minimize the issues presented by the techniques described above, it was developed a simple, economic, efficient and accurate system that can be used to study the erosion phenomena in coastal area. This system should allow one to acquire accurate GNSS positioning data on beaches, using a simple acquisition device in order to enable the creation of a three--dimensional surface of the study area, which temporal variation can permit to quantify the erosion rates. This system is formed by an acquisition device, a GNSS receiver and antenna, and a Android Tablet/Smartphone that allows us to collect the antenna attitude from it's internal orientation sensors. To collect information from the angles of the sensors, an Android application was developed. With the Tablet/Smartphone attitude angles, it is possible to correct the GNSS observations in order to obtain the vertical position - a dedicated C# application was developed with this objective. This corrections allows the creation of surfaces that accurately represent the terrains where the observations were conducted. In the future, using several observations at different times, it is possible to study the variations of sand volumes of the surveyed beaches and analyse the morphological variations. Some preliminary tests were performed concerning the creation of the surfaces, in order to evaluate the internal accuracy of the entire system. For this goal, we developed several scripts, using GMT (Generic Mapping Tools), that can also be used in the future to calculate the variations of the beach volume.

In this respect, the results of some tests carried out to evaluate the accuracy of the developed technology are presented. In particular, by analysing the deviations with respect to a reference surface and by evaluating the sensitivity when measuring on an irregular surface.

Keywords

GNSS, Kinematic Positioning, Natural Hazards

Resumo Alargado

O principal objetivo do trabalho de investigação descrito neste manuscrito foi, desenvolver métodos (software/hardware) para aquisição de dados em três dimensões, de forma a realizar a análise morfológica em ambientes de costa arenosa. Este sistema, quando estiver concluído, pretende estimar as variações volumétricas em praias que estão a ser erodidas a taxas elevadas, contribuindo diretamente para estudos voltados para a proteção da zona costeira. Este trabalho está focado principalmente no desenvolvimento de software para estimar as posições em 3D utilizando um dispositivo de hardware de baixo custo dedicado que também foi desenvolvido no âmbito deste projecto.

Têm sido utilizados vários métodos para estudar estes fenômenos, tais como topografia clássica através do levantamento de pontos discretos, com a utilização de teodolitos e/ou estações totais que são altamente demoradas. Outros métodos que começam recentemente a ser utilizados, é o LiDAR terrestre ou aéreo, são muito caros e não são adequados para serem utilizados em muitos locais, devido a exigências governamentais (como licenças, meio ambiente e questões de segurança). Também ocorrem limitações semelhantes com o Mapeamento de Sistemas Móveis (MMS) com scanners a laser que permitem o levantamento, quando o veículo estiver em movimento na praia.

Para minimizar os problemas apresentados pelas técnicas acima descritas, desenvolveu-se um sistema simples, económico, eficiente e preciso que possa ser utilizado para estudar os fenómenos da erosão da zona costeira. Este sistema deve permitir adquirir dados de posicionamento GNSS precisos nas praias, utilizando um dispositivo simples de aquisição, para permitir a criação de uma superfície tridimensional da área de estudo, o que pode permitir a variação temporal para quantificar as taxas de erosão. Este sistema é formado por um dispositivo de aquisição, um receptor GNSS e uma antena e um Tablet/Smartphone Android que nos permita recolher a atitude da antena apartir dos sensores de orientação internos. Para a recolha da informação a partir dos ângulos dos sensores, foi desenvolvida uma aplicação Android. Com os ângulos da atitude do Tablet/Smartphone, é possível corrigir as observações GNSS, para obter uma posição vertical - foi desenvolvida uma aplicação C# com este objectivo. Esta correcção permite a criação de superfícies que representam com precisão os terrenos onde foram realizadas as observações. No futuro, utilizando várias observações em momentos diferentes, é possível estudar as variações do volume da areia das praias estudadas e analisar as variações morfológicas. Foram realizados alguns testes preliminares sobre a criação de superfícies, para avaliar a precisão interna de todo o sistema. Para este objectivo, foram desenvolvidos alguns scripts, utilizando o GMT (Generic Mapping Tools), que também podem ser utilizados no futuro, para calcular as variações de volume da praia.

Assim, são apresentados os resultados de alguns testes realizados para avaliar a precisão da tecnologia desenvolvida. Em particular, através da análise dos desvios em relação a uma superfície de referência e pela avaliação da sensibilidade na medição sobre uma superfície irregular.

Palavras-Chave

GNSS, Kinematic Positioning, Natural Hazards

Contents

List of Figures

List of Tables

Acronyms List

AJAX	Asynchronous JavaScript and XML
ARFF	Attribute-Relation File Format
ASCII	American Standard Code for Information Interchange
CSS	Cascading Style Sheets
DEM	Digital Elevation Model
DR	Dead Reckoning
DSM	Digital Surface Model
DTM	Digital Terrain Model
ECEF	Earth-Centered Earth-Fixed
EDM	Electronic Distance Measurement
ENU	East North Up
FIG	International Federation of Surveyors
GMT	Generic Mapping Tools
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HTML	HyperText Markup Language
ICEUBI	International Conference on Engineering of University of Beira Interiror
IDE	Integrated Development Environment
IDL	Instituto Dom Luis
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LiDAR	Light Detection And Ranging
MMS	Mobile Mapping Systems
RAM	Random Access Memory
RINEX	Receiver Independent Exchange Format
RMS	Root Mean Square
SEGAL	Space and Earth Geodetic Analysis Laboratory
TBC	Trimble Business Center
TLS	Terrestrial Laser Scanning
TRU	Topcon Receiver Utility
TS	Total Stations
UBI	Universidade da Beira Interior
UML	Unified Modeling Language
USB	Universal Serial Bus
WGS84	World Geodetic System 1984
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Motivation

The main goal to select the subject of this dissertation, was my motivation to work on the field of Geodesy, which is an area where computer science and technology are developing together and where I would like to develop work during my professional activity in the future.

Coastal hazards (like floods and tsunamis) are a major problem all over the world, including Portugal and Singapore, and can cause a large number of deaths and destruction. I think it is important to contribute to the development of technological solutions that allow the efficient monitoring the state of the coast line in a period of time. Another important element that contributed for choosing this project was the fact of being integrated into a research grant where I had the opportunity to expand my knowledge in various areas.

After a previous research of the used methods, was possible to conclude that all of them had serious problems and it was important to contribute with a new solution. Some classical methods such as the use of Total Stations are highly time-consuming. Other methods that started recently being used, as land or air LiDAR, are very expensive and are not suitable to use in many places, because of government requirements (such as licenses, environmental and safety issues). Similar limitations also occur with mobile mapping systems (MMS) with laser scanners that allow the survey, when the vehicle is in motion on the beach (the geo-referencing is done by GNSS + IGS).

All of this gave me a great motivation to perform this dissertation where I pursued to build a system with less restrictions and cheaper than existing systems.

1.2 Description of Problem

Beach volume changes are defined as the mass of sand that enters or leave an area of interest over time. Since it is customary to only make measurements a few times per year, for example in summer and winter, one only can observe the total gain or loss of sand between these measurements.

The volume of the beach is computed by constructing a very high resolution digitial terrain model (DTM) of the area. This DTM can be converted into an estimation of the volume of sand of the beach. The problem is how one can construct an accurate DTM of the beach in a cost effective manner.

1.3 Approach

First of all a research was performed to find the existence of similar systems used to monitor beach morphological variations and, to study and analyse their strengths and weaknesses.

GNSS in kinematic mode is a convenient method to map the topography of the beach in a cost-effective way. However, it still suffers from the problem that the GNSS mast will not always be vertical, introducing an error of the estimated height. The novelty of this research is that we use the tilt-meters of a relatively cheap tablet to correct for this problem.

In order to compute the volumetric variations of the sandy surfaces, a simple, fast and economic system was developed based on an acquisition device with a GNSS receiver and antenna and an Android Tablet/Smartphone. It was developed several applications to complement this innovative system. The first developed application was GyroData, an Android application, that collects the attitude angles of the Tablet/Smartphone (Euler Angles) from the internal gyroscope. With these collected angles it is possible to correct the position of the GNSS antenna in order to precisely calculate the 3D surface from the observed tracks.

To process the acquired GNSS data, the Trimble Business Center (TBC) software was used in order to obtain all observed points along the study area. This software was used because it was available in our Lab, however other commercial software's could have been used to process the GNSS data. After the data processing, it will be necessary to compute the three-dimensional surfaces that represent the observed terrain. Some Linux scripts were developed that in the future can also be used to evaluate the sandy volumetric conclusions.

We started by performing the analysis of the requirements in order to design the system. We will discuss the features, concepts, GNSS basics, programming languages and the IDE's used for the development of the system. Finally, we discuss the tests done.

1.4 Objectives

This dissertation is integrated in a research scholarship in the framework of the project SingSubs - Monitoring of Subsidence in Singapore. It was carried out at SEGAL (Earth & Space Geodetic Analysis Laboratory), a laboratory that it is a scientific collaboration between UBI (University of Beira Interior) and IDL (Instituto D. Luis - University of Lisbon).

The main objective of this project is to develop methods (software and hardware) for data acquisition (vertical profiles) in order to perform morphological analysis in sandy shore environments to detect erosion (or accumulation) rates, which can help to mitigate coastal hazards. Figure 1.1 represents a comparison of beach erosion and re-nourishment in Happisburgh on the East Anglian coast in England, between 1996, 2006 and 2012 [1] [2].

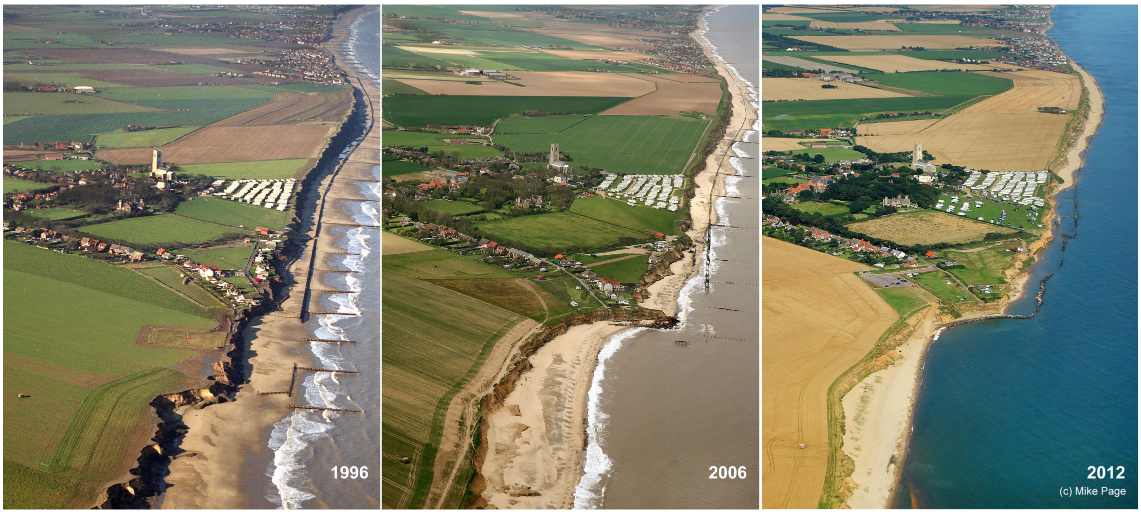


Figure 1.1: Comparison of beach erosion and re-nourishment [2]

In terms of hardware, we intend to developed a system that can cover large areas in a relatively short period, with sufficient accuracy. The system is based on the use of a lightweight device that contains a structure of three wheels with a GNSS receiver and antenna and a tablet solidly attached to the structure. Figure 1.2 shows the developed device.

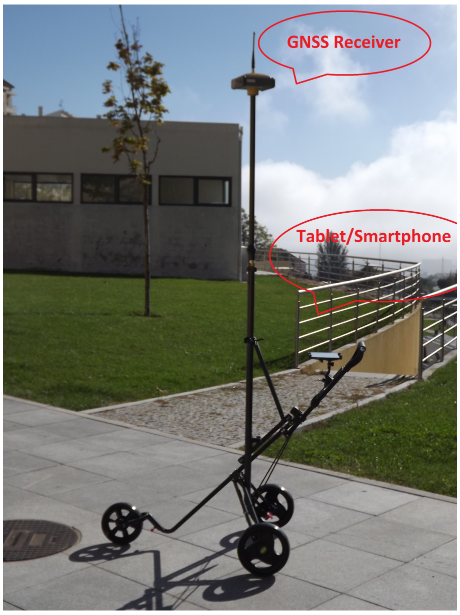


Figure 1.2: Device developed to acquire the GNSS and attitude observations

The GNSS device is used to obtain the 3D positions which are referred to the surface by applying known offsets and corrections estimated using the information provided by the tablet. Figure 1.3 shows the workflow of the methodologies used to compute the three-dimensional surface using the integrated system discussed in this dissertation.

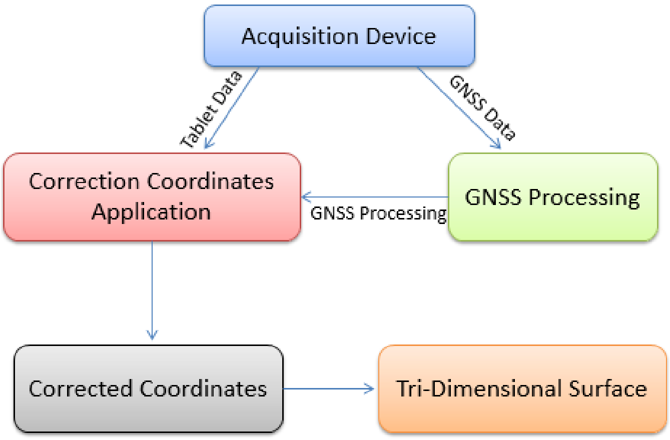


Figure 1.3: Methodologies Workflow

This methodology is composed by the acquisition device that collects GNSS and Tablet data to process the coordinates of the observed points. After the observed GNSS data processing, the coordinates are corrected using the developed application and using the collected attitude angles. The output of the application is a .txt file with the corrected coordinates of each point and with this information it is possible to compute the three-dimensional surface.

The long term objective is to use this prototype to see if there have been any significant changes in the volume of existing sand along the beaches of Singapore, which may have direct consequences on the protection of the coastline.

1.5 Main Contributions

The main contributions of this dissertation are here presented:

- Construction of a lightweight equipment, low cost and easy to carry for acquiring GNSS data and attitude angles of the GNSS antenna.
- Development of a set of applications that permit to estimate volumetric variations using the developed device. In the end, this integrated application can also be used with other devices since the observations are made available in the right formats.

During the course of this dissertation the following conference papers were written:

- Pecha Kucha Presentation with the title "Low-Cost Integrated GNSS/INS for Surface Variation Monitoring" in 1st FIG Young Surveyors European Meeting (http://figysn.ordemengenheiros.pt/pt/).
- An article with the title "Low-Cost Integrated GNSS/INS for Beach Variation Monitoring" was submitted and accepted for the ICEUBI conference (http://iceubi2013.ubi.pt).

1.6 Organization of Dissertation

This report is divided into chapters.

Chapter 1 - Introduction The first chapter is an introduction to the topic of the dissertation, outlining the motivations that led to the development of the same, the objectives to be achieved and finally the approach taken throughout the development.

Chapter 2 - Literature Review and Explication of the Context

In this chapter a reference is made to existing systems which collect Kinematic GNSS data with the same purpose for which the implemented system is proposed.

Chapter 3 - Requirements Analysis

In chapter 3 a superficial description of the operation of the system is made, and information about the requirements and restrictions of each developed module is presented.

Chapter 4 - System Development

Chapter 4 presents a detailed explanation of all the methods used to develop each of the modules of the system.

Chapter 5 - Test and Results

Charter 5 presents some tests made to the system in different terrain types and their respective results.

Chapter 6 - Conclusion and Future Work

Presents the overall conclusions of the development of the project, as well as a description of what can be added to this work.

References

Appendix A - Attachments - User Manual

Manual of the operation of the system.

Chapter 2

Literature Review and Explication of the Context

2.1 Kinematic GNSS

The concept of relative kinematic positioning using GNSS carrier phase observations was described by Remondi [3]. After that, several methods were developed to implement pseudo-kinematic and semi-kinematic GNSS surveying, all based on static initializations [4] [5]. Purely kinematic GNSS methods require the computation of the position of an object in permanent motion, without static initialization.

With the technological advances in hardware and software, several accurate kinematic GNSS precise relative positioning solutions started to be provided by private or governmental agencies. Kinematic GNSS positioning with centimetre accuracy is now a standard product available for a wide range of applications. However, all these solutions are confined to a limited range from a reference station (around 20-40 km) or from a network of stations.

2.2 Classical Methods To Acquire Kinematic GNSS Data

In the development of any system, it is important to identify the required features and functionalities. One approach is to analyze similar products that have comparable functions in order to evaluate the pros and cons.

2.2.1 Total Station

The conventional terrestrial survey methods, such as total station (TS), can be used to perform beach monitoring in a stand-alone mode or by supporting kinematic GNSS. Total station (TS) instruments combine an EDM (Electronic Distance Measurement), an electronic digital theodolite, and a computer in one unit. These devices can automatically measure horizontal and vertical angles, as well as distances, and transmit the results in real time to a built-in computer.

The horizontal and vertical angles and slope distance can be displayed, and then upon keyboard commands, horizontal and vertical distance components can be instantaneously computed from theses data and displayed.

If the instrument is oriented in direction and the coordinates of the occupied station are input to the system, the coordinates of any point sighted can be immediately obtained. This data can all be stored within the instrument, or in an automatic data collector, thereby eliminating manual recording [9].

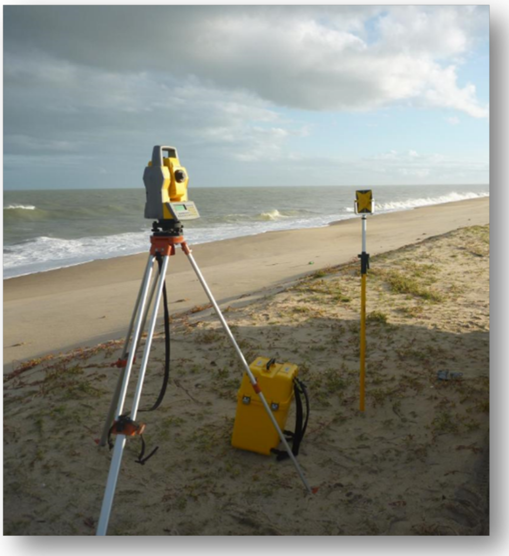


Figure 2.1: Total Station

Local precision is one of the biggest advantages a total station has over a GNSS but, total station has more disadvantages than advantages. Total station are highly time-consuming, still requires several days to monitor a beach which is expensive because of the man-hours involved.

Regarding the accuracy of a total station, it is dependent on instrument type and it is highly dependent on the levelling of the instrument.[10].

2.2.2 Light Detection And Ranging (LiDAR)

The LiDAR technology determines the distance between ground objects and sensors by measuring the time a pulse of transmitted energy takes to return to the LiDAR sensor. When coupled with a ground referencing system, LiDAR sensors make dense, geo-referenced point elevation measurements [12] [13]. Compared with traditional surveying and mapping systems, including photogrammetric systems, LiDAR technology provides a fast and accurate alternative [14] [15] for mapping large areas at high resolution and is gradually being adopted as the primary technique to generate Digital Elevation Models (DEMs) [16] [17]. The United States and some European countries have begun to utilize LiDAR systems to generate regional to national DEM products [12] [18] [19].

LiDAR techniques are arguably superior to traditional methods (e.g., photogrammetric techniques or automated image matching and elevation extraction techniques) for generating DEMs for at least five related reasons, as described by Reference [11]. First, the dense LiDAR point clouds enable generation of highly accurate, high resolution DEMs. Second, surface features can be extracted based on a height context analysis of the LiDAR points [20] [21], enabling accurate mapping of surface features like buildings, trees, and even power lines and pipelines [20] [22]. Additionally, a high density dataset allows users to apply a multi-scale or multi-resolution strategy of feature identification [21]. Third, it is easier to identify slight elevation changes using dense LiDAR point clouds, making it easier to map regions with little textural variations, including variations in the surface of vegetation canopies [20] [23]. Fourth, vegetation canopy structure can be determined because LiDAR pulses penetrate the canopy and create multiple returns [24] [25]. Different multiple return patterns during different seasons can also facilitate vegetation classification [26]. Lastly, LiDAR can be used to map ground elevations even in regions of dense vegetation because of multiple returns [20].

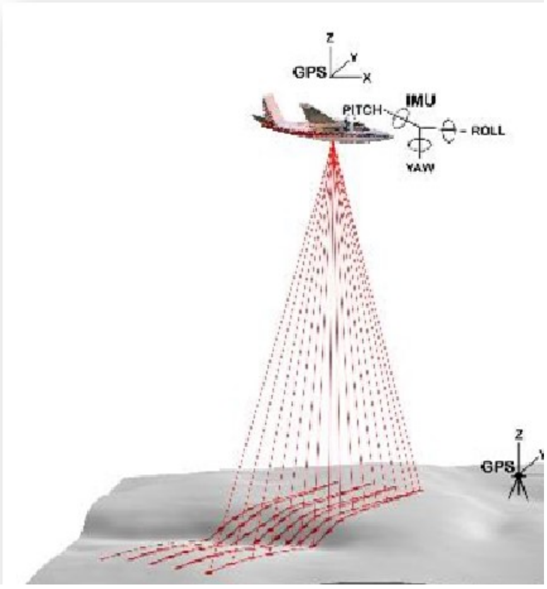


Figure 2.2: Terrestrial and Airborne LiDAR

2.2.2.1 Terrestrial LiDAR

The terrestrial laser scanning (TLS) instruments are a valuable asset for a variety of real-world problems. Beside their traditional use in areas such as digital terrain modelling, geological exploration, surveying, archeological and architectural documentation, they have found applications in forensics and crime scene investigations, deformation monitoring of dams, buildings and landslides, and more recently for scanning movie sets. For many applications of TLS, it is assumed that the point cloud is error-free. Although this is far from reality, numerous efforts have attempted to produce more reliable and accurate point clouds. New and improved filters and measurement algorithms have limited a great deal of outliers and noise in the point cloud.

Typical TLS projects involve grouping common points to model a surface or object to reduce the effect of random noise. Modern TLS instruments behave much like total stations; however their systematic errors cannot be eliminated by taking direct and reverse measurements. Instead, users usually have to rely on the manufacturer's black box instrument calibration. The manufacturer's calibration has been shown to be an effective method for improving the scanner's observation precision, but it has also been demonstrated that point-based user self-calibration can further reduce the standard deviation of the scanners observations [27].

2.2.2.2 Airborne LiDAR

Airborne Lidar is an emerging technology and relying on the accuracy of GPS and IMU components in the system, it can produce data of high resolution and accuracy in both horizontal and vertical directions. With this information, researchers can analyse tree volumes, building structures and create 3D urban models. Although applications of Lidar data vary, all these modelling tasks are built around one indispensable procedure: generation of Digital Terrain Models (DTMs) from raw Lidar point cloud.

A Lidar point cloud is a set of points associated with x, y, z positional information. These points include ground points (signals returned from the terrain) and non-ground points (signals returned from objects such as trees and buildings). Elevation information from the whole point cloud, both ground points and non-ground points, forms a Digital Surface Model (DSM) whilst elevation information from ground points forms a DTM. Since ground points are mixed with non-ground points in the point cloud, processing algorithms are needed to generate DTMs from DSMs [28] [29].

Airborne Lidar is the fastest of all mentioned methods, but flight time is expensive and therefore only suitable for large areas of hundreds of kilometres.

2.2.3 Mobile Mapping Systems(MMS) With Laser Scanners

Mobile mapping combines absolute and relative positioning devices and sensors to locate features in a global reference system [30]. The absolute positioning sensors provide the framework for data collected by the relative positioning sensors. GPS is used in most mobile mapping systems, however it can only be used outdoors. Other positioning sensors such as INS or dead--reckoning devices are also employed to compute the position and orientation to fill in the gaps when GPS signal is temporarily interrupted [31]. When data from these sensors are integrated properly, a continuous and accurate tracking of the absolute position of the sensors becomes possible.

The final accuracy is a function of the accuracy of each of the individual sensors and the quality of the calibration and registration of the sensor positions relative to each other. The best absolute accuracy reported is 0.4 m to 1 m (RMS) and significantly better for relative feature-position [30] [32]. Since most mobile mapping systems are without active 3-D sensors, most surface details are not mapped. In navigation systems, cameras are used to track features to estimate the vehicle trajectory and motion and to reconstruct and recognize objects [33] [34].

A common feature of mobile mapping systems is that more than one camera is mounted on a mobile platform, allowing for stereo imaging and 3-D measurements. Direct geo-referencing of digital image sequences is accomplished by the multi-sensor navigation and positioning techniques. Multiple positioning sensors, GPS, Inertial Navigation System (INS) and Dead Reckoning (DR), can be combined for data processing to improve the accuracy and robustness of geo-referencing. The ground control required for traditional mapping is eliminated. The systems can achieve centimeter accuracy of vehicle positioning and meter or sub-meter 3-D coordinate accuracy of objects measured from the geo-referenced image sequences [35] [36].

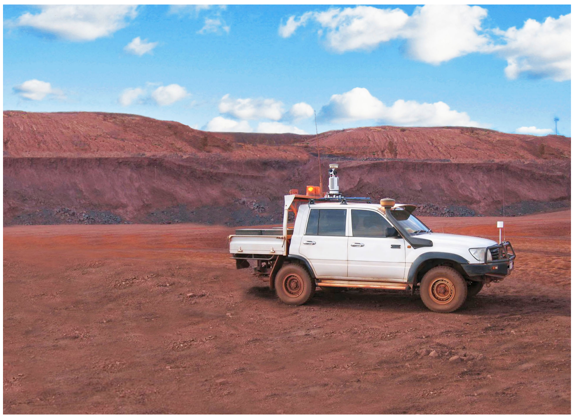


Figure 2.3: Mobile Mapping Systems(MMS) With Laser Scanners

2.2.4 Summary

After the analysis of the above methods, it is possible to verify that total stations are highly time-consuming, the terrestrial and airborne LiDAR, are very expensive and are not suitable for use in many places, due to government requirements (such as licenses, environmental and safety issues). Similar constraints also occur with Mobile Mapping Systems(MMS).

2.3 Android Applications to Acquire Gyroscope Data

Now that we have shown how our GNSS+tablet solution might complement current systems to monitor beach volume, we will now list the existing applications that were used in this research.

To develop the desired Android application was necessary to define the requirements and the features that it should have. With this purpose a research was made focusing the existing Android applications that are not suitable for the requirements of the developed system.

2.3.1 Physics Toolbox Gyroscope

This application measures and displays a graph of Angular Velocity (rad/s) vs. Time (s) that can be exported to Google Drive or in an e-mail as a .txt attachment with comma separated values (.csv). This gyroscope can be used to measure rotational and/or periodic changes in the x, y, and z planes [44].

This application is free but, does not fit what we want to do in this project because, if it is necessary to stop in the middle of an observation, it is not possible to add a note to the file to find out where the observation was stopped or where there is an error in the observation . It also does not save the time when each point is observed to then synchronize with the GNSS data.

2.3.2 Sensor Display & Logging

This application is developed for ENGO500 project for Geomatics Engineering in University of Calgary. Display sensor information, display gyroscope and accelerometer value and logging GPS, GYRO, ACC data in .CSV file [46].

This application has no cost. Does not fit what we want to do this project because it has the same limitations described for Physics Toolbox Gyroscope and it also does not save the data in .txt file, does not save the time when each point observed.

2.3.3 Sensor Kinetics Pro

Sensor Kinetics Pro demonstrates the physics of gravity, acceleration, rotation, magnetism and more as these forces are measured by your phone or tablet. The application includes comprehensive help files with easy to understand information and experiments that you can perform with the sensors [47].

This application is available for Android and iOS operating system. It is paid at a cost of €0.89. It has the disadvantage of not being able to send data to the computer ready to use. Collects information from 2 in 2 seconds and it doesn't serve this work's purpose because it is not possible to associate each observation to a GNSS point. The format in which data is stored is not the most perceptible and each observation mixing the received data with the previous. Also it is not possible to add notes during the data collection.

2.3.4 Data Sensors

Data Sensors captures data from the sensors integrated in your Android terminal. Data Sensors reads data from accelerometer and magnetometer. GPS and microphone are disabled.

At each run, you can selected the acquisition rate, the sensors to be acquired and for how much time you want to grab the data. Data acquired are stored in a text file and they can be sent via e-mail. If a call occurs during the acquisition, the process will not stop. If the device lacks one of the sensors, the sensor will be listed as unavailable.

Data Sensors is used to collect data by the actions: walking, cycling and stand still. The data collected are processed to obtain a series of features. Once completed the capture time, the calculated features are stored in a file named DataSensors-DDMMYYYY-HHMMSS.arff on DataSensors folder, with the ability to send files to an email address. The files created have a structure to analyse ARFF in Weka library, or MyWeka application for Android. Developed at Department of Applied Mathematics and Analysis, University of Barcelona [48].

While it is a freely available application, it lacks all necessary features. Does not save the data from the gyroscope sensor and the orientation sensor. Also it does not save the data in ASCII file and does not allow to add notes over a observation.

2.3.5 Fused Linear Acceleration

Fused Linear Acceleration provides a working code example and Android application that demonstrates how to fuse the acceleration, gyroscope and magnetic sensors to provide a reliable measurement of linear acceleration. Fused Linear Acceleration relies on the gyroscope and magnetic sensors to determine the tilt of the device so the gravity component of the acceleration can be subtracted from the signal.

The linear acceleration measurement is not skewed by periods of actual linear acceleration (i.e, accelerating in a vehicle) unlike a low-pass filter or complementary filter implementation. While this example is implemented with Android/Java, the jist of the algorithm can be applied to almost any hardware/language combination to determine linear acceleration [49].

This application is free but has the disadvantage of not being able to send data to the computer ready to use. The format in which data is stored is not the most perceptible and each observation mixes the received data with the previous.

2.3.6 Sensor Log

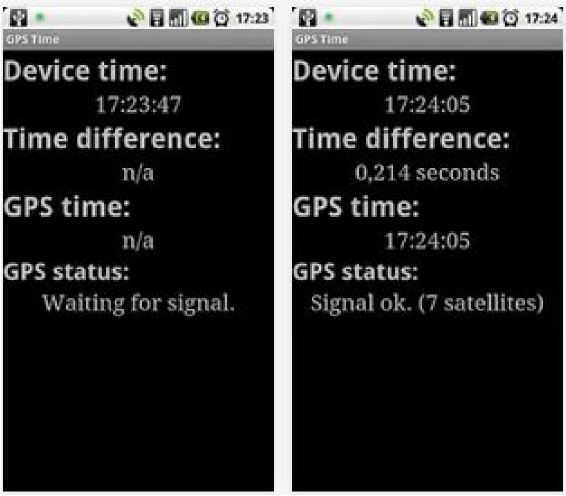
Sensor Log saves data from available sensors (with timestamps) in ASCII files on external storage [50]. This application is free but does not fit for what we want because it does not save the time of each point observed in the file, it does not save the files in ASCI files, it does not allow to add notes during the observation.

There are some other applications to view gyroscope and accelerometer information but don't write and save this values to ASCII files. Apart from this fact, it is also not possible to add notes.

2.4 Android Application "GPS Time"

To perform the necessary corrections to the collected GNSS coordinates it is very important to make an accurate correspondence between this data and the attitude angles collected from the Table/Smartphone. Satellites have very accurate atomic clocks, however portable devices such as Tablets/Smartphones do not have as accurate internal sensors. These differences produce a time offset between the Android device and the GNSS system.

GPS Time is a free Android application and help to keep the internal clock of one Android device accurate by synchronizing it with the GPS time signal. ROOT access is required to actually set the time. The difference (in seconds) between the Tablet/Samrtphone time and the GPS time can be seen without root privileges [51].



Device time:
17:23:47
Time difference:
n/a
GPS time:
n/a
GPS status:
Waiting for signal.

Device time:
17:24:05
Time difference:
0,214 seconds
GPS time:
17:24:05
GPS status:
Signal ok. (7 satellites)

Figure 2.4: GPS Time Application

2.5 Trimble Business Center

Trimble Business Center office software is suitable for processing and analysing satellite and terrestrial survey data recorded in the field, and exporting it to a design package. The software provides numerous innovative and unique features, and it is easy to learn and use.

Trimble Business Center (TBC) is a start-to-finish survey-data processing solution, focusing on measurements and data integrity. Surveyors can use TBC to import and adjust their survey control, process topographic and location data, create digital terrain models and contours, and review all their data before taking the survey to their preferred CAD package for completion. TBC will process topographic survey data, machine control data, and 3D point clouds, to name a few. This review concentrates on the processing of control-survey information from GNSS, bench levels, and traverses. TBC begins at the "start menu" where the user can create a new project, open a project, edit settings, and also run through the tutorials that will be most valuable to new users [37].

Chapter 3

Requirement Analysis

This section is intended to generally describe the operation of the system. The necessary information about the requirements of each module developed are also presented. At the end of this chapter it is expected that requirements definition is well presented in order to move to the development phase of the system with the necessary information.

3.1 General Description - System Perspective

The developed system has as main users engineers and scientists. Thus, the system contains the following modules.

- **Acquisition Device** - This is an integrated system that we developed to collect the GNSS data and attitude angles simultaneously.
- **Andoid Application "Gyrodata"** - It is the application developed to collect the attitude (Euler angles) of GNSS antenna to the millisecond.
- **Equipment used in data acquisition and data processing** - This section describes the equipment used to acquire and process GNSS and the Tablet/Smartphone data and also the corresponding outputs of the processed data.
- **Scripts to Convert Coordinates** - These are scripts to convert ECEF to ENU coordinates and the inverse.
- **Air2Ground** - It is the application developed to transform the coordinates of the points processed by TBC from the base of the antenna to the ground.
- **Surface Computations** - Consists in a set of scripts that were developed to build the three-dimensional surfaces where it is possible to analyse the results obtained in the tests.

3.2 Technologies Utilized

Since the objectives have been identified and presented, it is of utmost importance to understand what is the best technology to use in order to achieve the desired objectives. The main objective of this project involves the construction of a system that has an application that makes corrections to coordinates of GNSS points through the data collected by the Tablet with Android operating system from GyroData Android Application.

For the development of the Air2Ground application we decided to use the Microsoft technologies, particularly Microsoft Visual Studio and C# programming language, because it permits to develop robust applications in a very used Integrated Development Environment that was already known. Regarding the GyroData Android application development, Android Studio was used, because it permits to acquire the attitude angles from the devices where is running easily. In addition, it is currently the most used platform in the world and it does not have associated costs to develop applications. The existing knowledge about development in Java[61] also helped to select this platform. For the processing of GNSS data, it was decided to use the Trimble Business Center (TBC) and the construction of the Surface Computations it was used scripts with Generic Mapping Tools (GMT).

3.2.1 Microsoft Visual Studio

It is a Microsoft technology that has a development environment complete applications. Components remain the same regardless of language, which makes it very easy to change the design and language, and have the same resources at the same place. Has included a set of products inclined for testing, databases, software architecture and design.

Can use various programming languages (Visual Basic, C, C++, C#, J#, F#, HTML, CSS, ASP.NET, AJAX). The debugger is very good because it facilitates debugging of errors during run time. The scheduler takes the exact point of failure code, allowing its correction more efficiently.

In this project, was used the Microsoft Visual Studio 2012 for the development of the application of corrections coordinates of GNSS points to object-oriented language C# [38].

3.2.2 Android Studio

Android Studio is a new Android development environment based on IntelliJ IDEA. Similar to Eclipse with the ADT Plugin, Android Studio provides integrated Android developer tools for development and debugging. This IDE support Java and XML programming languages [52].

This was the IDE that chosen to develop the Android application to collect the orientation sensor data (Euler Angles) from the Tablet/SmartPhone.

3.2.3 Trimble Business Center

Trimble Business Center(TBC) office software is ideal for processing and analysing satellite and terrestrial survey data recorded in the field, and exporting it to a design package. The software provides numerous innovative and unique features, and it is easy to learn and use [43].

For this project was used the Trimble Business Center 2.4.0 for processing de GNSS points and to export the correct coordinates to .xyz file.

3.2.4 Generic Mapping Tools (GMT)

The Generic Mapping Tools (GMT) is an open source collection of about 80 command-line tools for manipulating geographic and Cartesian data sets (including filtering, trend fitting, gridding, projecting, etc.) and producing PostScript illustrations ranging from simple x-y plots via contour maps to artificially illuminated surfaces and 3D perspective views [53].

GMT was used to compute the three-dimensional surfaces obtained from the observed data.

3.2.5 Functional Requirements

- The system should allow the user to easily mount and remove the mast with the GNSS antenna.
- The system should allow the user to adjust the Tablet into the required position (Pitch and Roll equal zero).
- The user should be able to enter the parameters required by the application in order to calculate the corrections to be applied to the GNSS observed data.
- The Tablet or Smartphone must have the orientation sensors very stable, 512MB of memory RAM to make sure that it is capable to store all the collected data and it has to work with the Android operating system.

3.2.6 Non-Functional Requirements

- The system must include a user manual so that users can utilize all the technologies implemented without any difficulty.
- The project should be completed by June 2014.
- The system should be able to be used by engineers and scientists.

3.2.7 Interface Requirements

- The applications must have a appropriate layout to the proposed theme. They should show a simple environment and should be easy for the user to understand all the functionalities.

3.2.8 Restrictions

3.2.8.1 Physical Constraints

- To process the acquired GNSS data, it is necessary to have access to a software that is able to process multiple points. It is very likely that this software will have to be paid.
- In order to enter the time offset it is necessary to have a free or paid Android application where it is possible to see the time difference between the Tablet/Smartphone and the Satellite time.

3.2.8.2 Logical Constraints

For users to use the application Air2Ground, user must work on Windows and to use de Gyrodata it is necessary to have an Android device(Tablet or Smartphone).

3.2.8.3 Restrictions on the Operation of the System Level

Given that there are still several people without minimal knowledge of computer science and Geodesy, a part of the population may have limited access to this system.

3.2.9 Diagrams

3.2.9.1 Uses Cases Diagram

The use case diagrams allow us to represent the interactions of the system with different external agents: typically users whether they are people or other systems [40]. Then it is presented a use case diagram for one user.

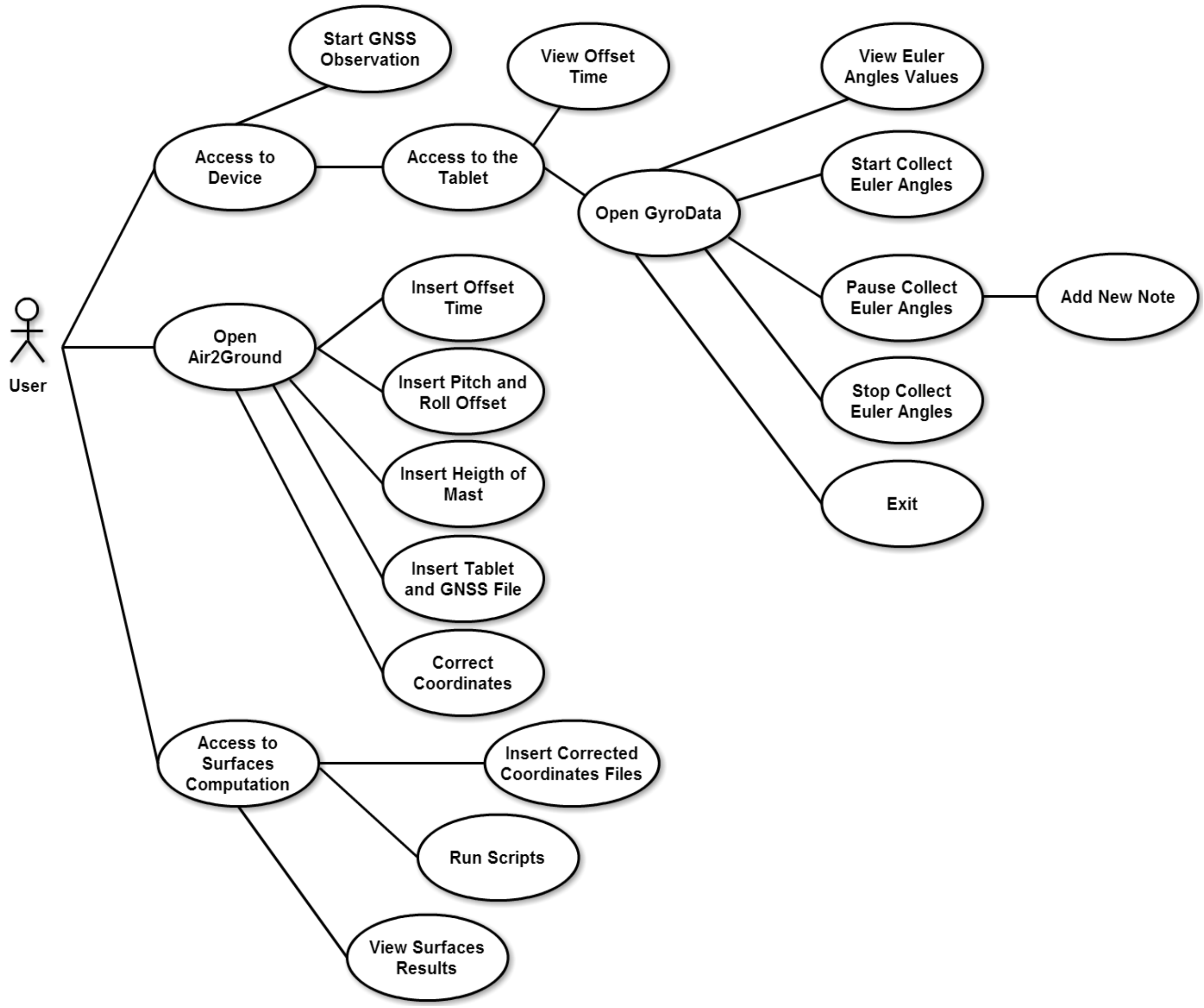


Figure 3.1: Use Case Diagram of the User

Chapter 4

System Development

This chapter describes the methodology used in the implementation phase of the system and the decisions done. The language and terms used in this chapter may not to be easily perceived, and for that reason this section is oriented above all to the programmer. It starts by presenting the hardware developed to collect GNSS data and attitude angles. Then, all applications developed in order to achieve the objective of this thesis are presented and explained step by step.

4.1 Acquisition Device

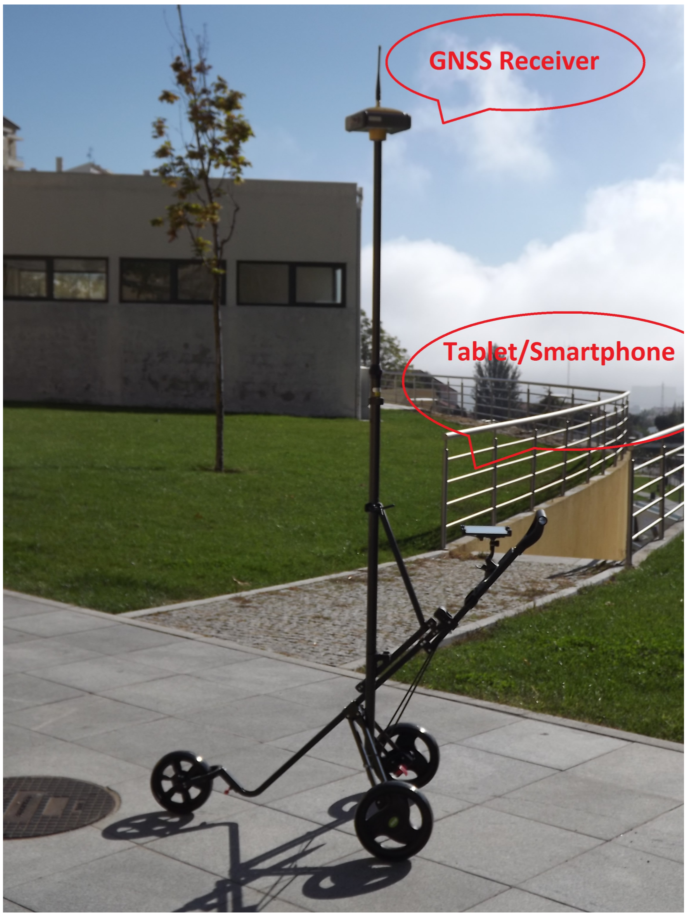


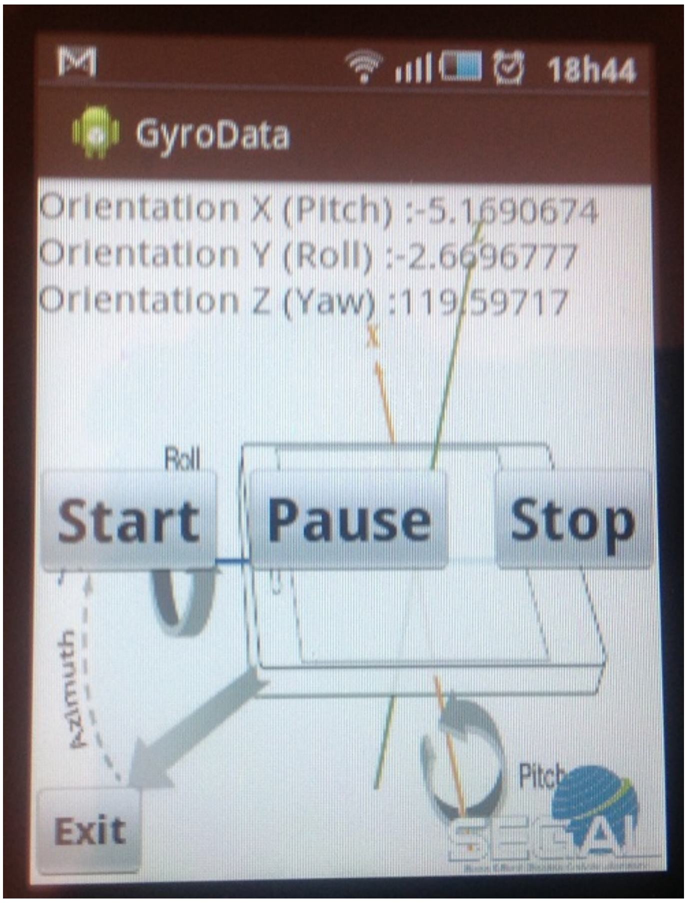
Figure 4.1: Acquisition Device

Figure 4.1 shows the integrated system developed to collect the GNSS data and attitude angles simultaneously. It is a simple and economic system, in comparison with other existing systems, and its advantage is that it can be easily assembled and transported, keeping the ability to collect the data correctly and with the necessary precision to obtain good results.

The system consists of a golf clubs trolley, adapted with two iron brackets to be able to embrace a two meters mast holding the GNSS antenna with the best possible stability. The golf club cart features two quick release wheels and a third removable wheel, being easy to pack for simple storage.

4.2 Android Application "GyroData"

The development of this application has the main objective to collect the Euler angles (Pitch, Roll and Yaw) through the orientation sensors of a Tablet or from a Smartphone. The application works on any Smartphone or Tablet with the Android operating system. It was developed for Android version 4.4 but it is compatible with all previous versions.

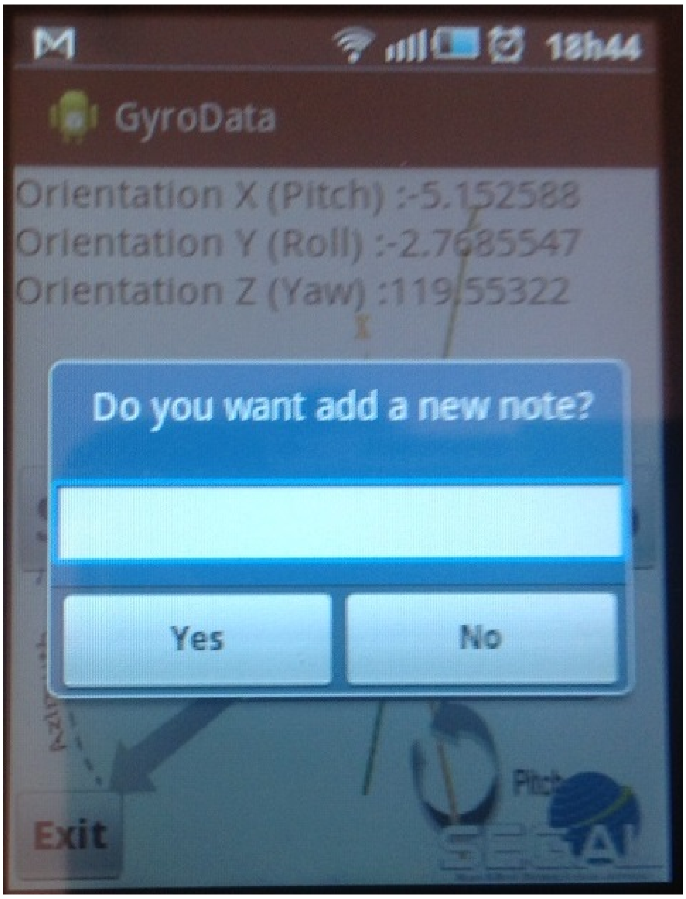


Orientation X (Pitch) :-5.1690674
Orientation Y (Roll) :-2.6696777
Orientation Z (Yaw) :119.59717

Figure 4.2: Android Application "GyroData" - Main Menu

Figure 4.2 shows the Android application "GyroData". This application was built based on the need to solve the problem studied in this dissertation. It allows one to record the values of the Euler angles observed to the second in a text file and it also records the time in the file where the angles Pitch, Roll and Yaw are saved. For each second, the application saves 15 points (time, pitch, roll and yaw) with the milliseconds corresponding to each of the points. This information is then used to synchronize Android application data with the GNSS data that will be explained later in the section 4.5 about the Air2Ground application to correct the coordinates.

The button "Start" starts to record the data. The button "Pause" allows, during a observation, to stop it and add a note(Figure 4.3) to the file. These notes can be something like, "stopped to drink water" or "the points of the last 30 seconds must be repeated."



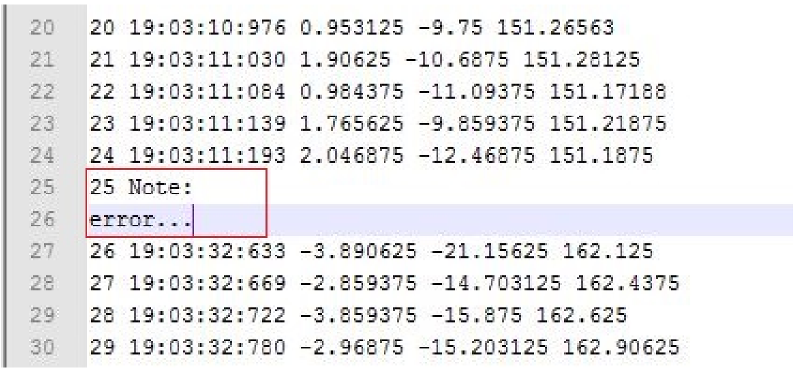
GyroData

Orientation X (Pitch) :-5.152588
Orientation Y (Roll) :-2.7685547
Orientation Z (Yaw) :119.55322

Do you want add a new note?

Figure 4.3: Android Application "GyroData" - Add Note

This feature, was created to provide an option to the user to remember important events. After writing and saving a note, the application allows you to continue the observation. To finish, the user will have to press the button "Stop" to stop the observations and it saves all observed data to a text file. The button "Exit" closes the application and returns to the Tablet or Smartphone menu.



```
20 19:03:10:976 0.953125 -9.75 151.26563
21 19:03:11:030 1.90625 -10.6875 151.28125
22 19:03:11:084 0.984375 -11.09375 151.17188
23 19:03:11:139 1.765625 -9.859375 151.21875
24 19:03:11:193 2.046875 -12.46875 151.1875
25 Note:
error...
26 19:03:32:633 -3.890625 -21.15625 162.125
27 19:03:32:669 -2.859375 -14.703125 162.4375
28 19:03:32:722 -3.859375 -15.875 162.625
29 19:03:32:780 -2.96875 -15.203125 162.90625
```

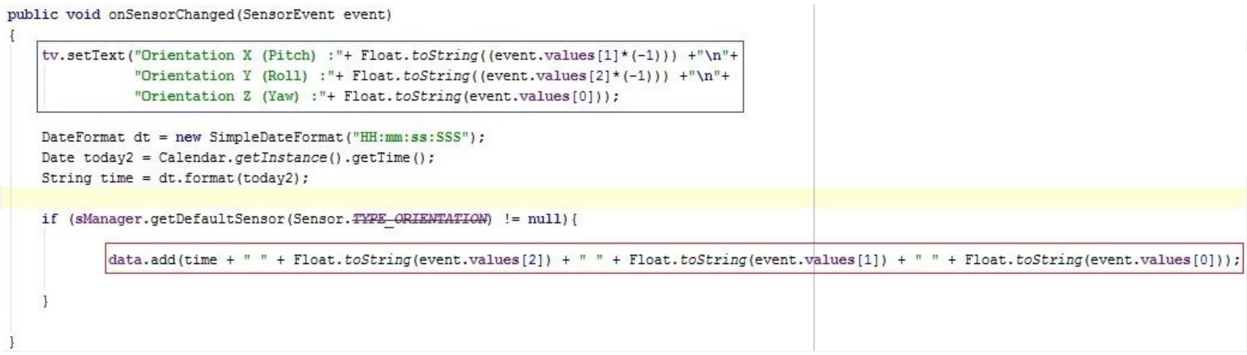
Figure 4.4: Mini-Output File of the GyroData Application

The format of the file that contains the observed data by this application is shown in Figure 4.4. The name of this file is composed by the date and time when the observation ended so we can have an idea of when the observation was made without having to open the file.

The classes *Sensor.TYPE_ORIENTATION* and *SensorEvent* of the Android SDK were to create the *onSensorChanged(SensorEvent event)* function which is shown in Figure 4.5. All values of Pitch, Roll and Yaw angles are in degrees.

- *values[0]:* Yaw or Azimuth, angle between the magnetic north direction and the y-axis, around the z-axis (0 to 359). 0=North, 90=East, 180=South, 270=West
- *values[1]:* Pitch, rotation around x-axis (-180 to 180), with positive values when the z-axis moves toward the y-axis.
- *values[2]:* Roll, rotation around the y-axis (-90 to 90) increasing as the device moves clockwise.

The collected data is stored in *ArrayList* of strings [55]. The instructions that are underlined in blue show the values of the Euler angles in the real time application. The instructions underlined in red represents the format in which data is stored in the text file (Time, Pitch, Roll, Yaw).



```
public void onSensorChanged(SensorEvent event)
{
    tv.setText("Orientation X (Pitch) :"+ Float.toString((event.values[1]*(-1))) +"\n"+
              "Orientation Y (Roll) :"+ Float.toString((event.values[2]*(-1))) +"\n"+
              "Orientation Z (Yaw) :"+ Float.toString(event.values[0]));

    DateFormat dt = new SimpleDateFormat("HH:mm:ss:SSS");
    Date today2 = Calendar.getInstance().getTime();
    String time = dt.format(today2);

    if (sManager.getDefaultSensor(Sensor.~~TYPE_ORIENTATION~~) != null){

        data.add(time + " " + Float.toString(event.values[2]) + " " + Float.toString(event.values[1]) + " " + Float.toString(event.values[0]));

    }

}
```

Figure 4.5: Function "onSensorChanged(SensorEvent event)"

4.3 Equipment Used in Data Acquisition and Data Processing

In the data collection process, a Topcon HiperPro receiver was used. Although, our system is basically independent of the GNSS equipment used, this model has the necessary requirements: it is a multi-function, multi-purpose double-frequency GPS and Glonass receiver [41]. In the tests performed, it was used baselines up to 20km, which is in accordance with the values used for centimeter-level topographic surveys.

The antenna attitude angles are obtained using an Android Tablet/Smartphone. The Tablet/Smartphone has an orientation sensor and together with the devices accelerometer improves motion sensing accuracy and allows to measure in which direction the device is moving/rotating in space (Pitch, Roll and Yaw), how much and how fast.

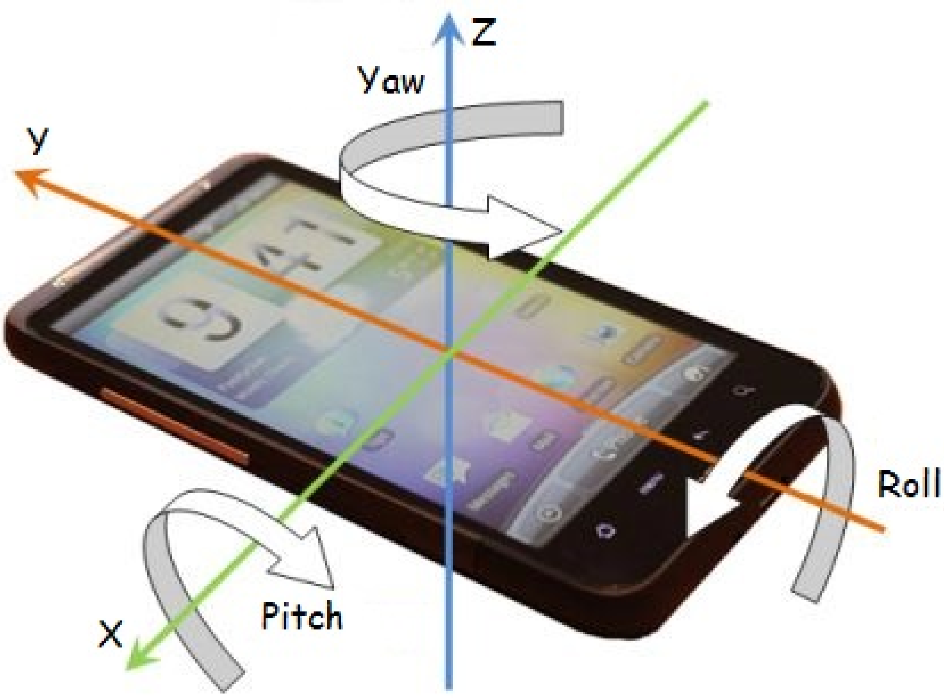


Figure 4.6: Tablet Reference Frame

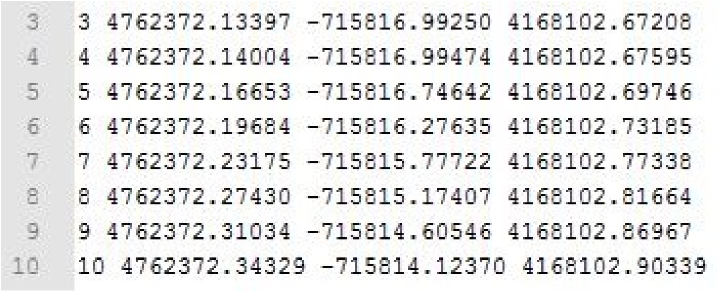
The precision of the orientation and accelerometer sensors is much greater than the precision of the Tablet's/Smartphone's compass and GNSS sensors. The compass and GNSS sensors are used to determine where the device is on the globe, and the orientation and accelerometer are good for finding out where the device has moved in the last few milliseconds. Therefore it depends upon what you're trying to control with the device's movement [56]. However, the measured angles and accelerations tend to drift with time, reason why they are not useful to obtain absolute positions.

It is necessary to use reference stations in order to obtain the coordinates of the observed points in the same coordinate system, even when they are observed at different times. In the tests carried out, two stations were used:

- The station installed in the roof of Block 6 Building of University of Beira Interior.

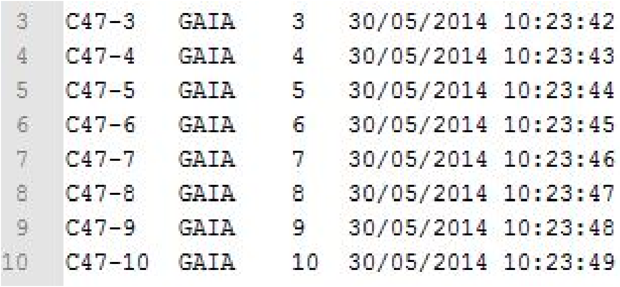
- The station installed in Vila Nova de Gaia which belongs to the National Geodetic Network from the General Direction of the Portuguese Territory.

TBC output is a list of coordinates of each point in the selected datum (reference system). In order to facilitate the conversion, a local coordinate system should be used. In our case, was used the National Reference Frame PT-TM06/ETRS89 - European Terrestrial Reference System 1989 [57].



```
3 4762372.13397 -715816.99250 4168102.67208
4 4762372.14004 -715816.99474 4168102.67595
5 4762372.16653 -715816.74642 4168102.69746
6 4762372.19684 -715816.27635 4168102.73185
7 4762372.23175 -715815.77722 4168102.77338
8 4762372.27430 -715815.17407 4168102.81664
9 4762372.31034 -715814.60546 4168102.86967
10 4762372.34329 -715814.12370 4168102.90339
```

Figure 4.7: TBC Mini-Output File - Coordinates Points



```
C47-3   GAIA   3   30/05/2014 10:23:42
C47-4   GAIA   4   30/05/2014 10:23:43
C47-5   GAIA   5   30/05/2014 10:23:44
C47-6   GAIA   6   30/05/2014 10:23:45
C47-7   GAIA   7   30/05/2014 10:23:46
C47-8   GAIA   8   30/05/2014 10:23:47
C47-9   GAIA   9   30/05/2014 10:23:48
C47-10  GAIA   10  30/05/2014 10:23:49
```

Figure 4.8: TBC Mini-Output File - Time of each Point

Figure 4.7 shows the output file of the TBC. Figure 4.8 shows the second output file of the TBC where it is possible to see where it is stored the time at each point was observed.

4.4 Scripts to Convert Coordinates

In the data acquisition process, it is necessary to know the antenna position with respect to the local coordinate system in order that it will be possible to transform the estimated position using GNSS (bottom of antenna) to the position of interest (position of the ground below the antenna mast).

Two scripts were developed in the C programming language. One to make the conversion from ECEF coordinates which is a Cartesian reference frame with its origin at the centre of the Earth, to local coordinates East-North-Up that are planimetric local coordinates, and the second to do the inverse conversion. This conversion is necessary because, in order to perform the coordinate corrections, the collected coordinates must be in the same reference frame as the Tablet (East, North, Up), because it uses its compass to point towards the north and the tilt meters use gravity to know what the vertical direction is.

4.4.1 Convert ECEF to ENU Coordinates

This script receives as parameters the names of input/output files and also the height of the antenna mast and it's distance to the ground. This coordinate transformation is based on ellipsoidal coordinates ϕ, λ, h and the corresponding rectangular coordinates x, y, z. The representation of this ellipsoid in terms of ellipsoidal coordinates is given by:

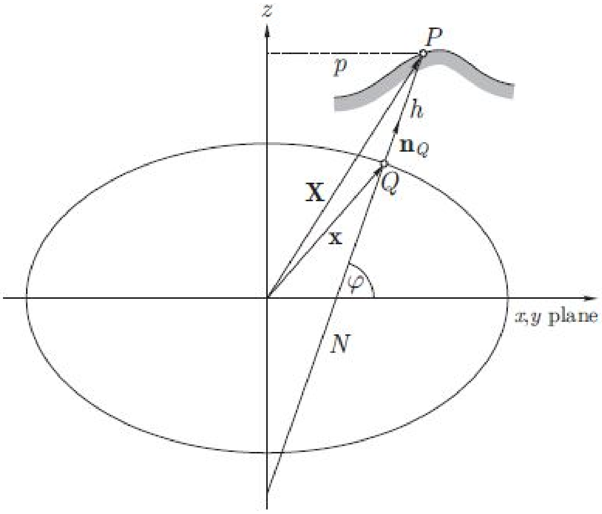


Figure 4.9: Ellipsoidal and Rectangular Coordinates

The origin of the rectangular coordinate system is the center of the ellipsoid, and the z-axis is its axis of rotation; the x-axis has the Greenwich longitude 0º and the y-axis has the longitude 90º east of Greenwich (i.e., λ = +90º) [59]. The transformation is based on mathematical formulas of Figure 4.10.

$$N = \frac{a^2}{\sqrt{a^2 \cos^2\varphi + b^2 \sin^2\varphi}}$$

$$h = \frac{p}{\cos\varphi} - N$$

$$p = \sqrt{X^2 + Y^2} = (N + h)\cos\varphi$$

$$\lambda = \arctan\frac{Y}{X}$$

$$\theta = \arctan\frac{Z\,a}{p\,b}$$

Figure 4.10: Equations of ECEF to Local Coordinates Transformation

After making the transformation and to complete the conversion of coordinates is necessary to use the following rotation matrix:

$$\begin{bmatrix} E \\ N \\ U \end{bmatrix} = \begin{pmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\cos\lambda\sin\varphi & -\sin\lambda\sin\varphi & \cos\varphi \\ \cos\lambda\cos\varphi & \sin\lambda\cos\varphi & \sin\varphi \end{pmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Figure 4.11: Rotation Matrix to East, North, Up

Finally, this script output a file with the coordinates E, N, U and another file with the parameters of the transformation are the ECEF coordinates of the first point of the converted file and the height of the mast for later do the inverse conversion.

4.4.2 Convert ENU to ECEF Coordinates

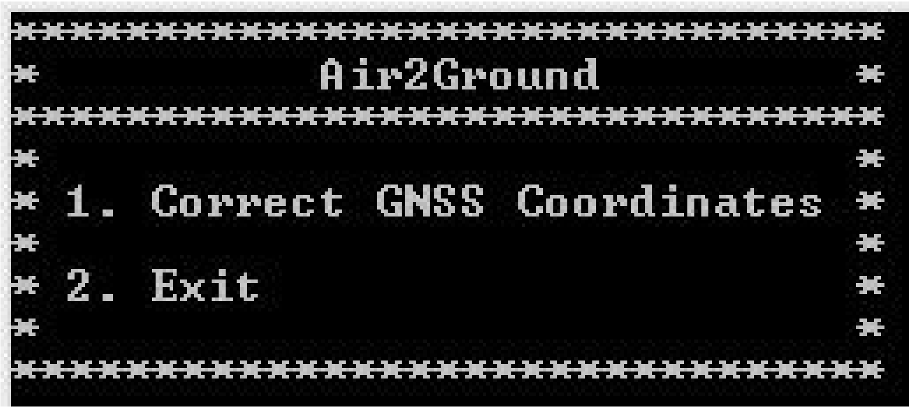
The script of this coordinate conversion receives as parameters the names of input/output files and also, the file with saved parameters on conversion ECEF to ENU. This conversion is done based on the ECEF coordinates of the first point recorded in the parameters file of the previous conversion and then apply the rotation matrix shown in Figure 4.12.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{pmatrix} -\sin\lambda & -\cos\lambda\sin\varphi & \cos\lambda\cos\varphi \\ \cos\lambda & -\sin\lambda\sin\varphi & \sin\lambda\cos\varphi \\ 0 & \cos\varphi & \sin\varphi \end{pmatrix} \begin{bmatrix} E \\ N \\ U \end{bmatrix}$$

Figure 4.12: Rotation Matrix to X, Y, Z

The script returns as output, a file with the coordinates X, Y, Z. This output file will then be used to construct surfaces of the collected data.

4.5 Air2Ground



```
*******************************
*         Air2Ground          *
*******************************
*                             *
* 1. Correct GNSS Coordinates *
*                             *
* 2. Exit                     *
*                             *
*******************************
```

Figure 4.13: Air2Ground Application

When the data is collected, it is very difficult to guarantee the vertically of the mast, because the antenna attitude will be influenced by the terrain topography, and consequently the surveyed points will not be at the vertical position of the ground position.

To correct the acquired data, it will be necessary to know the antenna attitude relative to a local reference frame. In order to make simple the conversion, we use the local horizontal plane (E, N) plus the vertical axis (V). This attitude can be described by three rotation angles known by Euler angles (Ω, ϕ, k or Roll, Pitch and Yaw(Azimuth)) relative to a X, Y, Z reference frame (see Figure 4.14). With this three angles, it is possible to correct each observed point using a simple rotation matrix described in Figure 4.15.

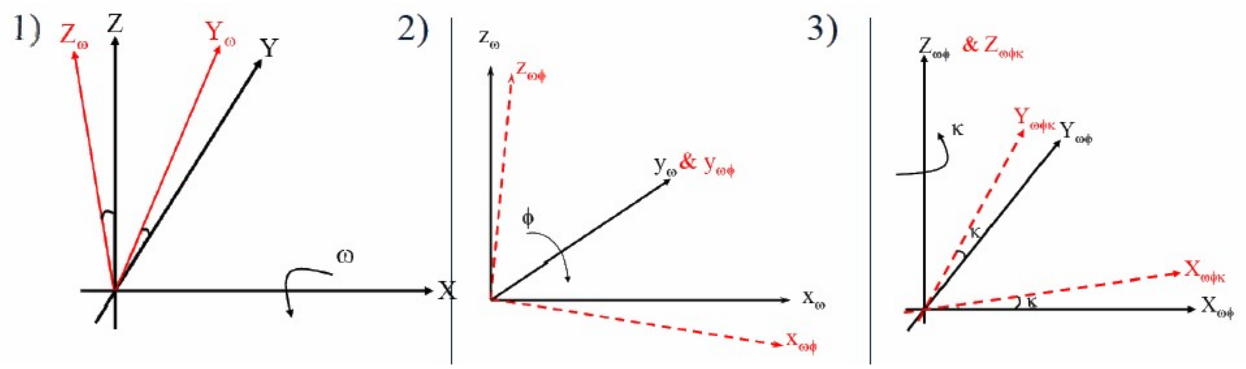


Figure 4.14: Rotation Around Axes X Y Z

$$\begin{bmatrix} x_{\omega\phi\kappa} \\ y_{\omega\phi\kappa} \\ z_{\omega\phi\kappa} \end{bmatrix} = M_\kappa M_\phi M_\omega \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad M_\kappa M_\phi M_\omega = M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

$$m_{11} = \cos\phi\cos\kappa$$
$$m_{12} = \cos\omega\sin\kappa + \sin\omega\sin\phi\cos\kappa$$
$$m_{13} = \sin\omega\sin\kappa - \cos\omega\sin\phi\cos\kappa$$
$$m_{21} = -\cos\phi\sin\kappa$$
$$m_{22} = \cos\omega\cos\kappa - \sin\omega\sin\phi\sin\kappa$$
$$m_{23} = \sin\omega\cos\kappa + \cos\omega\sin\phi\sin\kappa$$
$$m_{31} = \sin\phi$$
$$m_{32} = -\sin\omega\cos\phi$$
$$m_{33} = \cos\omega\cos\phi$$

Figure 4.15: Rotation Matrix

Air2Ground is the application developed to transform the coordinates of the points processed by TBC from the base of the antenna to the ground. It was implemented in console mode in C# [38] using the IDE Microsoft Visual Studio 2012 [39].

When all data is collected and processed the next step is the correction of GNSS coordinates using the Euler angles through a rotation matrix, in a process described in Figure 4.16. Where, "h" is the height of the antenna mast plus the distance to the ground.

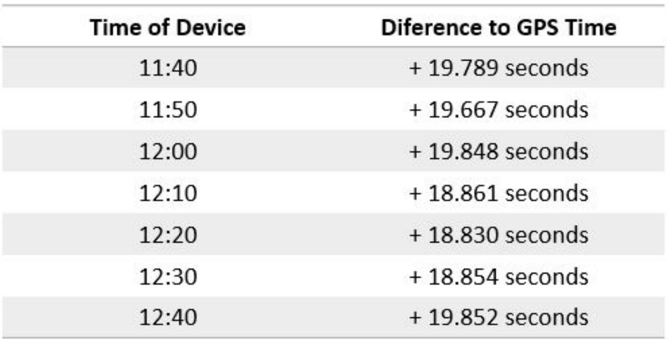
$$\begin{bmatrix} E \\ N \\ U \end{bmatrix} = \begin{bmatrix} E' \\ N' \\ U' \end{bmatrix} - \left(\begin{bmatrix} \cos\phi\cos k & -\cos\phi\sin k & \sin\phi \\ \cos\omega\sin k + \sin\omega\sin\phi\cos k & \cos\omega\cos k - \sin\omega\sin\phi\cos k & -\sin\omega\cos\phi \\ \sin\omega\sin k - \cos\omega\sin\phi\cos k & \sin\omega\cos k + \cos\omega\sin\phi\sin k & \cos\omega\cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} \right)$$

Figure 4.16: Correction Coordinates Equation

The Air2Ground needs the following parameters:

- *The time offset*

  This parameter corresponds to the average of time difference between GPS time and the Tablet/Smartphone time in milliseconds. To get this measure, the free android application "GPS Time" was used where the time difference at the beginning and at the end of the observation is collected, depending on the total time of observation. After some tests with this application it was possible to notice that the differences were not very pronounced. A test was performed for one hour and it was registered the time differences every 10 minutes. And, as it can be seen in Table 4.1, the variation was at maximum 1 second.



Time of Device	Diference to GPS Time
11:40	+ 19.789 seconds
11:50	+ 19.667 seconds
12:00	+ 19.848 seconds
12:10	+ 18.861 seconds
12:20	+ 18.830 seconds
12:30	+ 18.854 seconds
12:40	+ 19.852 seconds

Table 4.1: GPS Time Application Test.

- *The height of the mast, normally 2.000 m.*
- *The mast distance to the ground*

  This parameter is measured in the beginning of the survey.
- *Pitch Tablet/Smartphone sensor error.*
- *Roll Tablet/Smartphone sensor error.*
- *Three files .txt*

  One with the coordinates processed in TBC converted to ENU, another with the hour of each GNSS point exported from TBC and finally the file with the data of Android application "GyroData".

To treat the information contained in the input parameters and get the corrected coordinates, the following methods were used:

- *GPSFileFormat(string file, string file2)*

  This method receives as parameters a file with the coordinates of GNSS points and another file with the hours of GNSS points. Here, the assignment is made of hours in each GNSS point collected by the correspondent their "ID". Returns a List [38] with all the correct information from GNSS points (ID, Time, East, North, UP).

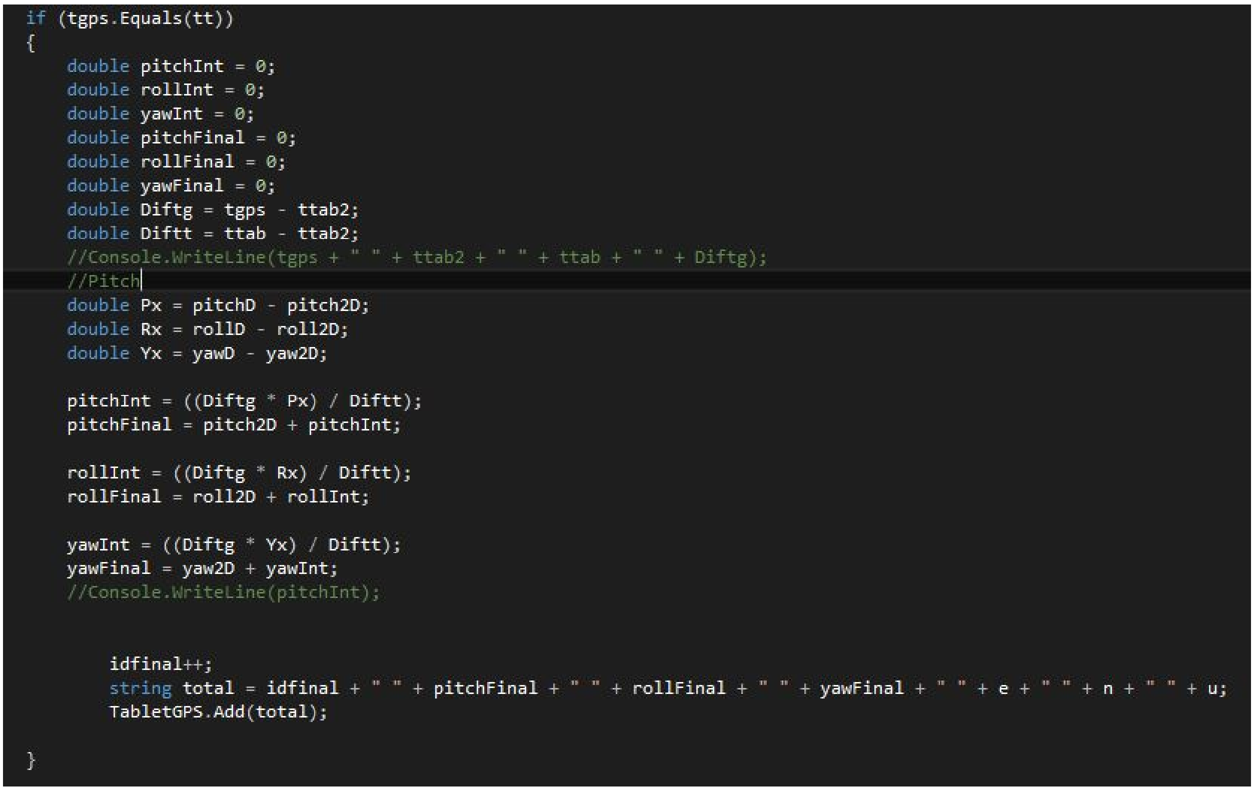
- *TabletFileFormat(string file, double offset, double offsetP, double offsetR)*

  This method processes data coming from the Android application "GyroData" which contains the observed Euler angles. Receives as parameters, the file with the Euler angles, the time offset, the pitch Tablet/Smartphone sensor error and the roll Tablet/Smartphone sensor error. Here the offsets are added or subtracted to data collected so that the correspondence between each instant and each GNSS point is the most accurate possible.

  This method returns one List [38] with the Euler angles corrected (ID, Time, Pitch, Roll, Yaw) and ready to be assigned to GNSS points.

- *TabletGPSFileFormat(List<string> Tablet, List<string> GPS)*

  This is one of the most important methods of this application. It is here that the synchronization is done between GNSS points and Euler angles data set. It receives as parameters the output lists of the two previous methods.



```
if (tgps.Equals(tt))
{
    double pitchInt = 0;
    double rollInt = 0;
    double yawInt = 0;
    double pitchFinal = 0;
    double rollFinal = 0;
    double yawFinal = 0;
    double Diftg = tgps - ttab2;
    double Diftt = ttab - ttab2;
    //Console.WriteLine(tgps + " " + ttab2 + " " + ttab + " " + Diftg);
    //Pitch
    double Px = pitchD - pitch2D;
    double Rx = rollD - roll2D;
    double Yx = yawD - yaw2D;

    pitchInt = ((Diftg * Px) / Diftt);
    pitchFinal = pitch2D + pitchInt;

    rollInt = ((Diftg * Rx) / Diftt);
    rollFinal = roll2D + rollInt;

    yawInt = ((Diftg * Yx) / Diftt);
    yawFinal = yaw2D + yawInt;
    //Console.WriteLine(pitchInt);


        idfinal++;
        string total = idfinal + " " + pitchFinal + " " + rollFinal + " " + yawFinal + " " + e + " " + n + " " + u;
        TabletGPS.Add(total);

}
```

Figure 4.17: Important print of the TabletGPSFileFormat Method

Synchronization begins to be done by comparing the second of time of the GNSS point with the second of time of each Euler angles data set. So, if they are equal, the Pitch, Roll and Yaw are recalculated, in order to assign the most correct value of each of the angles to the GNSS point. This is done using an interpolation [60] as can be seen in Figure 4.18 showing an example of the process that is done for each of the points.

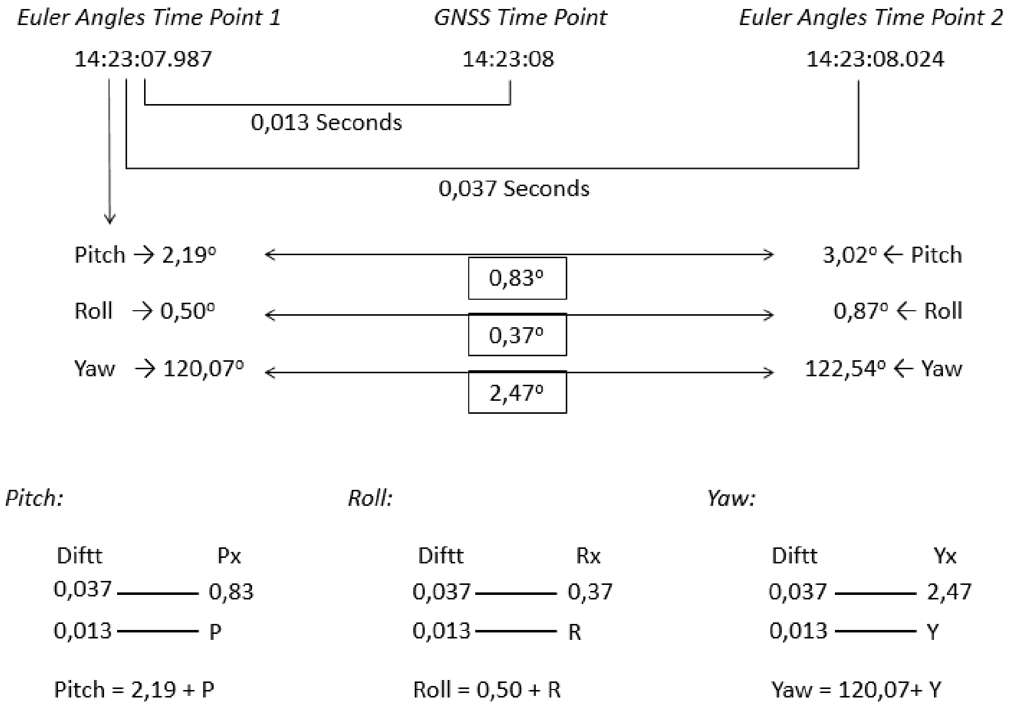


Figure 4.18: Interpolation Example

The procedure is, for each GNSS point, choose the two angles data sets that are nearest, one before and one after. The difference, in seconds, between these two nearest points is calculated (Diftg), as well as the difference in the observed angles (Px). Then, knowing the difference, in seconds, between the GNSS point and the previous angle data set, using a simple rule of three, it is simple to calculate the correction to be made in the attitude angle.

This process is repeated for each observed GNSS point, thus giving them the correspondent correct Euler Angle data set. This method has as output a List constituted by an id, the Euler angles and the corresponding GNSS coordinates (ID, Pitch, Roll, Yaw, East, North, Up) [38].

When the synchronization of the GNSS data with the data from the Android application "GyroData" is done, everything is ready to perform the coordinate corrections using the rotation matrix described before in Section 4.5.

- *MatrixTransformation(double pitch, double roll, double yaw, double e, double n, double u, double h, System.IO.StreamWriter file2)*

In this method, the correction of the coordinates of the GNSS points observed is done using a rotation matrix shown in Figure 4.15. Therefore, this method receives as parameters the "Pitch" and "Roll", "Yaw", "East", "North", "Up" resulting from the previous method, the "h" which is the sum of the height of the mast with the mast distance to the ground and the file "file2" which is where GNSS points with the corrected coordinates are stored.

Finally, the output of this application are two files, one with "ID, Pitch, Roll, Yaw, East, North, Up" ie, with Euler angles attributed to the GNSS points after the synchronization and another with the result of the coordinates corrections after applying the rotation matrix.

4.6 Surface Computations

With the corrected positions, we will be able to generate a three-dimensional surface. Were developed scripts based on GMT(Generic Mapping Tools) programs[15], to generate the surfaces from the set of point positions observed with the system. The developed scripts are:

- *txt2grd:* This script starts by receiving data from GNSS observations, thus determining the limits of surfaces. It defines the limits of the area of interest - the user can be interested in only a sub-area where no extrapolation is done. After using surface (GMT applications), it creates a regular grid from the irregular set of observed points.
- *create_surf:* This script calculate the differences between interest area and internal surfaces, the statistics between all of points, interest area and internal surfaces and were calculated statistics between surveys for different all of points, interest area and internal surfaces. Then, it is possible to check the error and the distance between each GNSS point of the surfaces.
- *surf_all:* Builds the surface of each GNSS observation and a surface with all surfaces built before. To verify in what kind of terrain the GNSS observations were made. The height values are represented by a gray scale.
- *surf_diff:* Here are buitl the surfaces between the differences calculated in the scrip "create_surf" and the surfaces of each of the GNSS observation. It is possible to verify the error of GPS through a colourful range of -10,10 cm and the changes of the surfaces after apply interpolation techniques.

Finally, repeating the process at different times it will be possible to analyze morphological changes in the monitorized surface during a time frame.

Chapter 5

Tests and Results

Some tests were carried out in order to evaluate the robustness of the system to compute the three-dimensional positions. During these tests it was possible to verify a very important factor, which is the sensitivity and stability of the used Tablet/Smartphone. Two tests were performed using two different Smartphones in the same place, where one (Smartphone 1) was more unstable and sensitive to shaking due to terrain variations than the other (Smartphone 2).

The next set of two tests were conducted in an asphalt terrain located about 5.6 km from the base station and had the duration of 5 minutes. Figure 5.1 represents the results after the correction of the coordinates with Euler angles collected by Smartphone 1 and Figure 5.2 represents the results of data collected in the same location using Smartphone 2. The red track is the track with the corrected coordinates and the lilac track has no corrections. The surveys were made in a flat terrain then the corrections should be minimal (2/3 centimetres) but, as can be seen in Figure 5.1 there are differences of about 72 cm. The conclusion is that is necessary to test the quality of the Tablet/Smarthphone sensors to be used with our system. Unstable sensors can introduce significant errors instead of correcting the tilts.

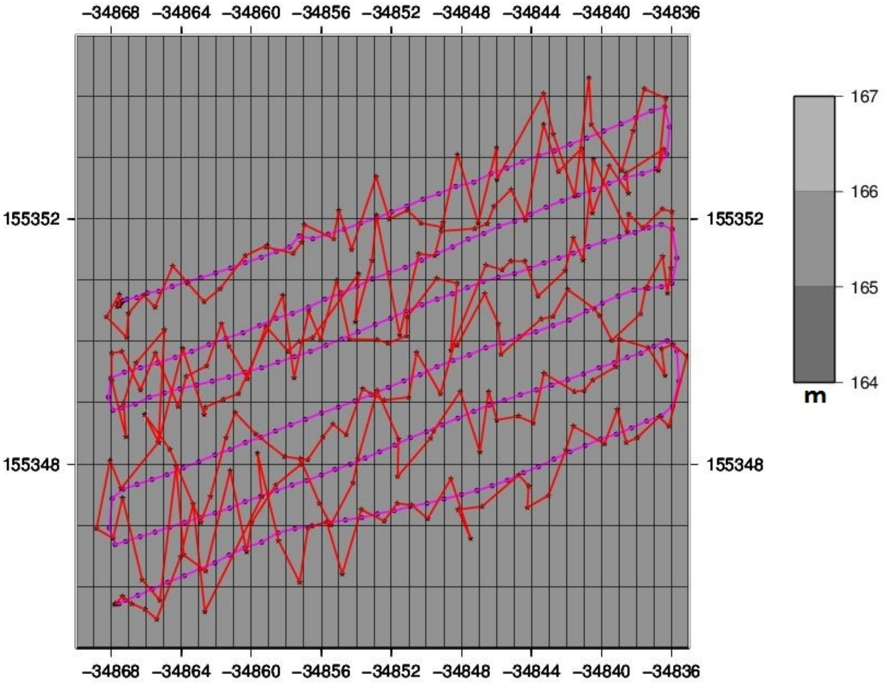


Figure 5.1: Surface with Smartphone 1 (x and y-axis are the east and north in meters).

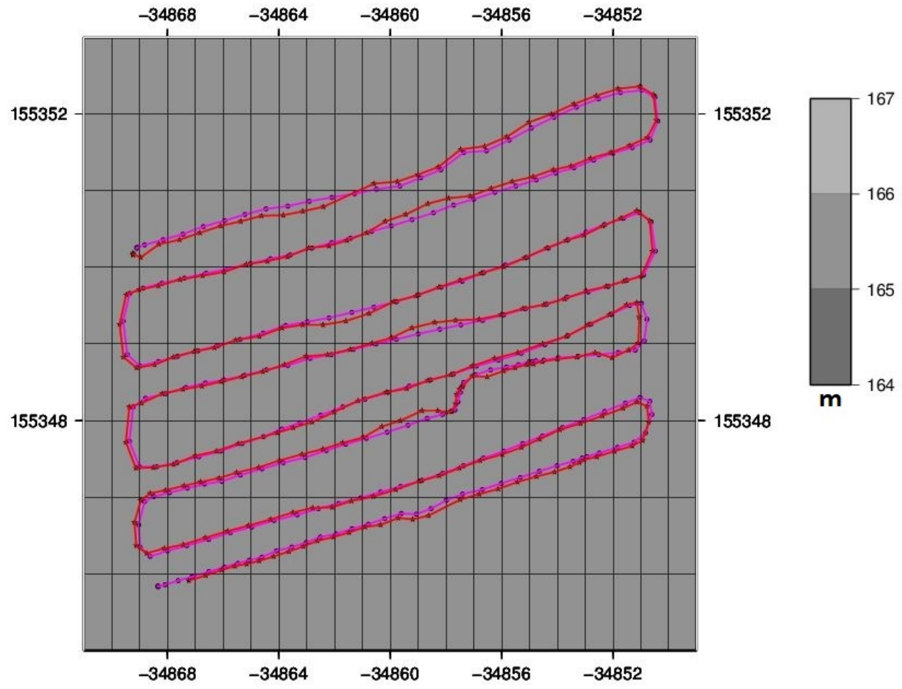


Figure 5.2: Surface with Smartphone 2 (x and y-axis are the east and north in meters).

Based on data of the previous surfaces, two more surfaces were created that are represented in Figures 5.3 and 5.4. In the two figures, the red track is the track with the corrected coordinates and the lilac track has no corrections. In Figure 5.3 it is possible to verify that the distances values between the two surfaces are between 72 cm and -0.5 cm, with media in 6.7 cm. In Figure 5.4 the distances values between the two surfaces are between -2.5 cm and 1 cm, with the average at 0.3 cm which is an good result.

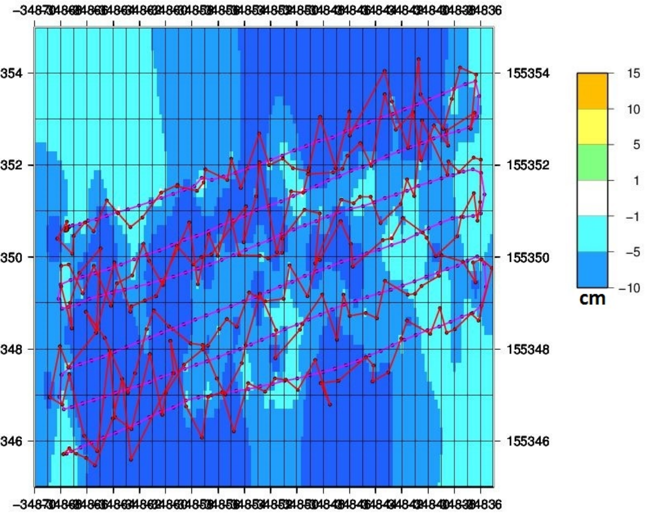


Figure 5.3: Differences between Surfaces of Smartphone 1 (x and y-axis are the east and north in meters).

Another test done was to investigate the influence of the Pitch and Roll errors in the correction of the coordinates. To do that we first perform a survey on a flat terrain and assume

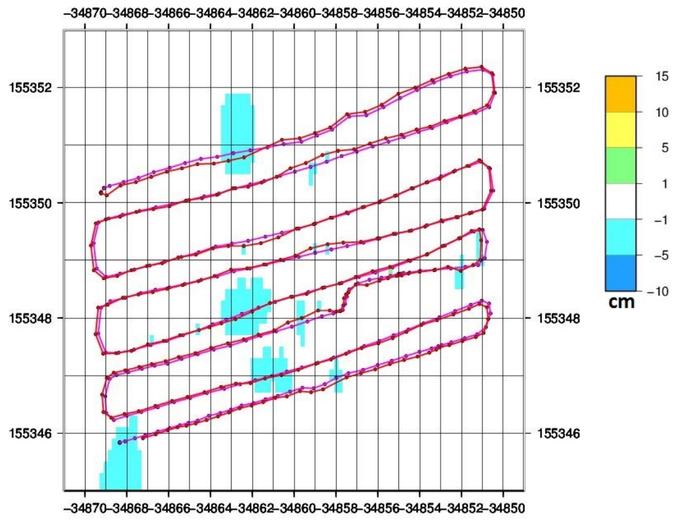
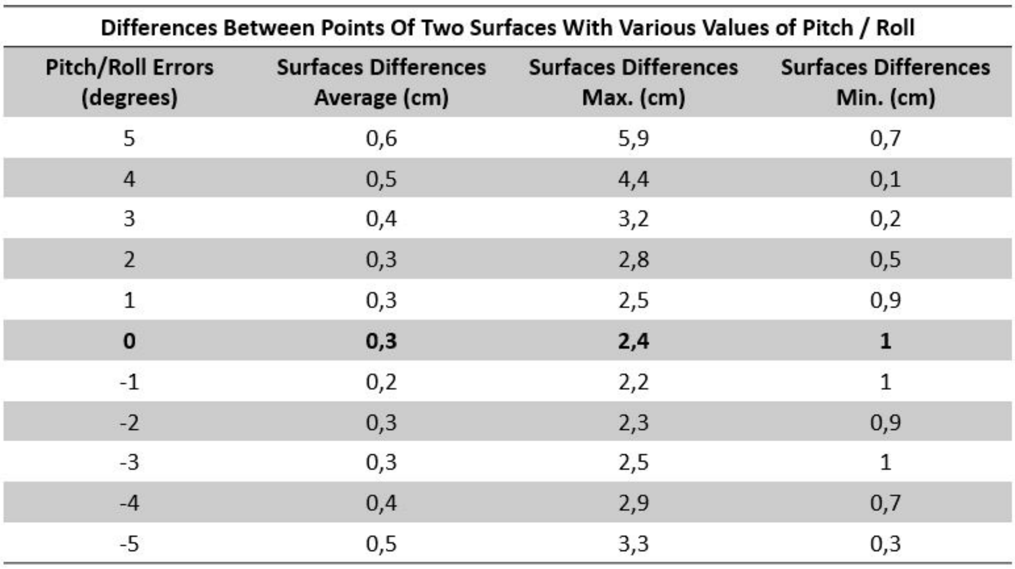


Figure 5.4: Differences between Surfaces of Smartphone 2 (x and y-axis are the east and north in meters).

that the Tablet/Smartphone has no errors in the Pitch and Roll angles. Then two surfaces were created, one without corrections and another after the corrections. The differences between this two surfaces are calculated (see Table 5.1 line 0). After this we gradually increase the values of the Pitch and the Roll by adding values between -5 and 5 degrees (e.g. if Roll = 15°, Pitch = 10° and the error is 2° we assume the Roll = 17° and Pitch = 12°) and we calculate new surfaces and differences for each new set of angles.

Table 5.1 presents the results of the tests described above. It is possible to see that the differences are not significant and the maximum differences occur when the error is 5°, however comparing with the value obtained when we assume there are no errors, this difference is only 3.5 cm.



Differences Between Points Of Two Surfaces With Various Values of Pitch / Roll			
Pitch/Roll Errors (degrees)	Surfaces Differences Average (cm)	Surfaces Differences Max. (cm)	Surfaces Differences Min. (cm)
5	0,6	5,9	0,7
4	0,5	4,4	0,1
3	0,4	3,2	0,2
2	0,3	2,8	0,5
1	0,3	2,5	0,9
0	**0,3**	**2,4**	**1**
-1	0,2	2,2	1
-2	0,3	2,3	0,9
-3	0,3	2,5	1
-4	0,4	2,9	0,7
-5	0,5	3,3	0,3

Table 5.1: Test the Pitch and Roll error.

The next set of three tests were conducted in a steep asphalt terrain located about 5.6 km from the base station and had the duration of 4 minutes. This test was to compare the differences between two individual tracks. The red track is the track with the corrected coordinates and the lilac track has no corrections. Figure 5.5 shows the two tracks superimposed on the altimetric map (estimated by averaging the two individual surfaces estimated at each survey) and it is possible to observe the gradual increasing of the heights. It is important to notice that the area to be controlled is only the one limited by the tracks. Outside, only extrapolation is possible which should not be done.

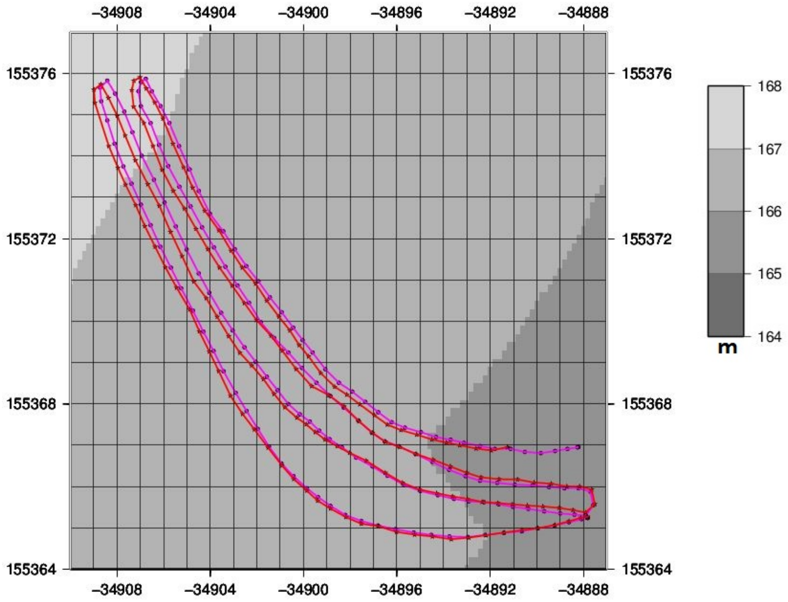


Figure 5.5: Tested Area with 2 tracks at different colors (x and y-axis are the east and north in meters).

In the Figure 5.6, the red track is the track with the corrected coordinates and the of lilac track has no corrections. This figure shows the differences between the points in the two tracks that are within 4 cm of distance from each other. In these marked points, we can see that the biggest differences are between 1 and 5 cm, but most of the results are between 0 and 1 cm. It is important to stress that are absolute differences between two acquired points at different epochs and by different operators, providing us with an estimate of the uncertainty that we can obtain on the surveys.

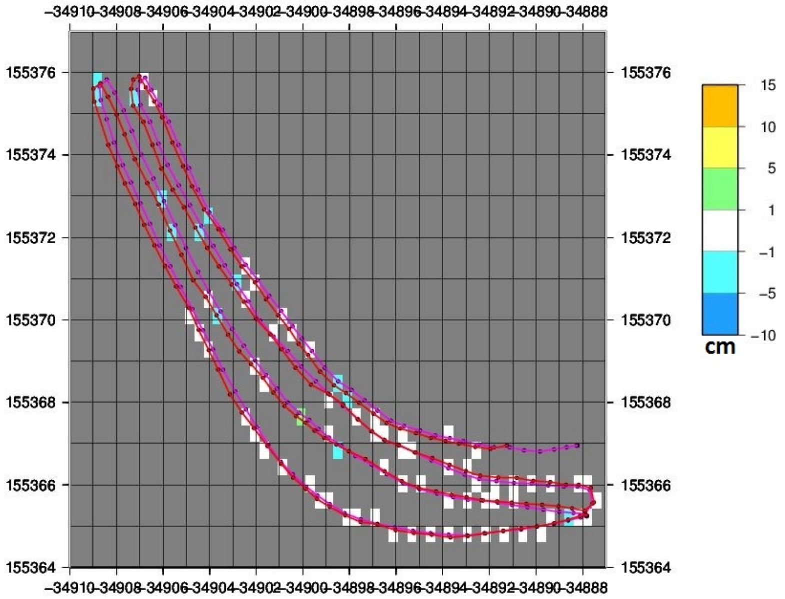


Figure 5.6: Differences between points within 4 cm of distance of each other in two different tracks (x and y-axis are the east and north in meters).

Figure 5.7 present an evaluation of the precision that it is possible to obtain inside of the interpolated area. We have computed the mean surface and Figure 5.7 presents the difference between this averaged surface and the corrected coordinates track. With the exception of a very small part, all differences are below 1 cm.

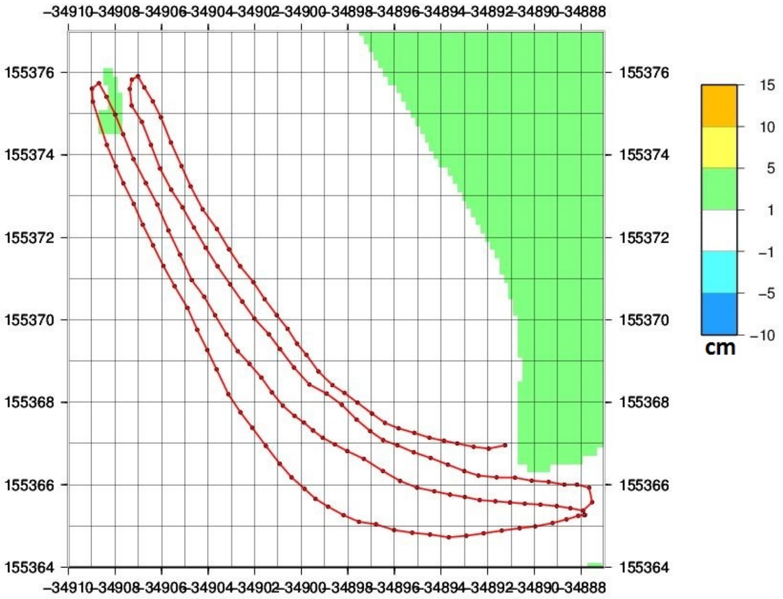


Figure 5.7: Difference between one surface and the average surface (x and y-axis are the east and north in meters).

The last three tests are the result of a survey made on a sand beach in Vila Nova de Gaia, located about 9.1 km from the base station and had the duration of 10 minutes. In the Figure 5.8 the red track is the track with the corrected coordinates and the of lilac track has no corrections. This figure shows two different performed tracks superimposed on the altimetric map (estimated by averaging the two individual surfaces estimated at each survey) and it is possible to observe the gradual increasing of the heights.

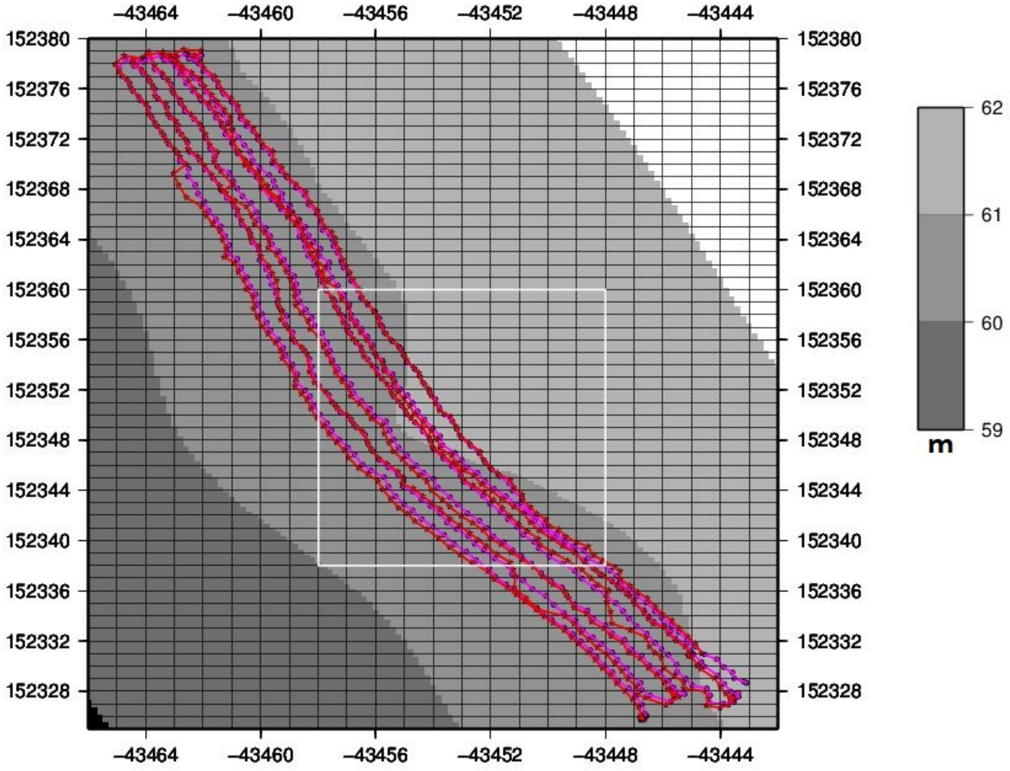


Figure 5.8: Beach Area with 2 tracks at different colors (x and y-axis are the east and north in meters).

In the Figure 5.9, the red track is the track with the corrected coordinates and the of lilac track has no corrections. This figure shows the differences between the points in the two tracks that are within 10 cm of distance from each other. In these marked points, we can see that the biggest differences are between 5 and 10 cm, but most of the results are between 1 and 5 cm and between 0 and 1 cm.

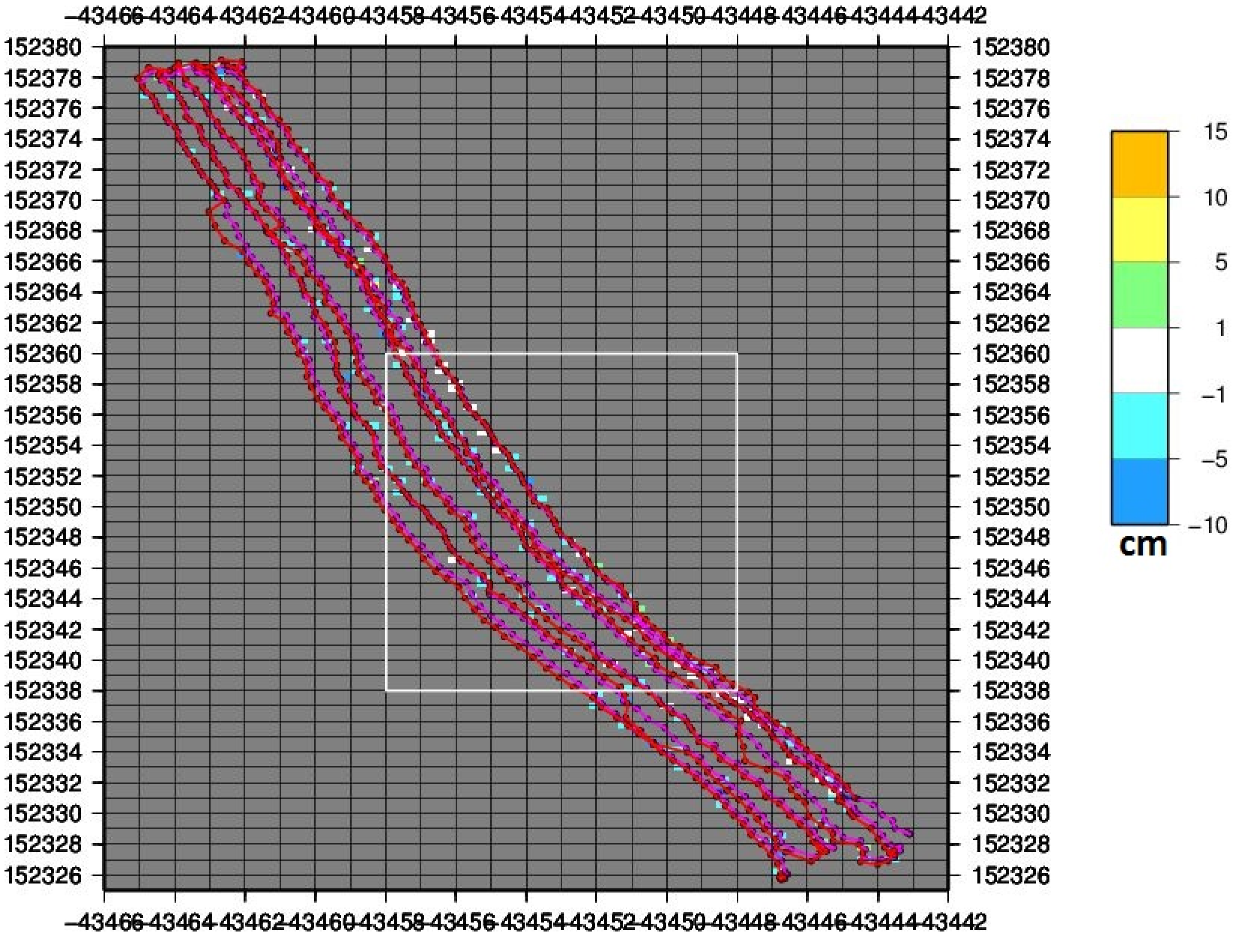


Figure 5.9: Differences between points within 4 cm of distance of each other in two different tracks (x and y-axis are the east and north in meters).

Figure 5.10 present an evaluation of the precision that it is possible to obtain inside of the interpolated area. We have computed the mean surface and Figure 5.10 presents the difference between this averaged surface and the corrected coordinates track. All differences are between 0 and 1 cm and between 1 and 5 cm.

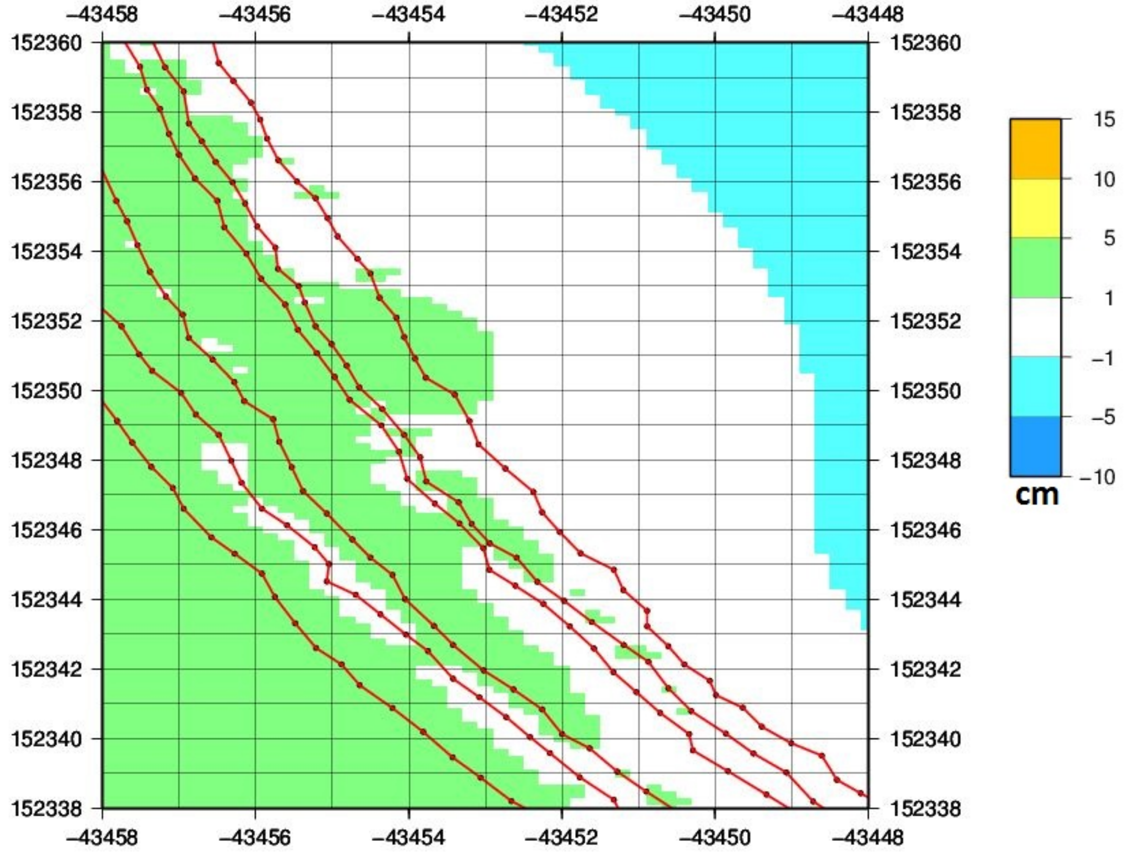


Figure 5.10: Difference between one surface and the average surface (x and y-axis are the east and north in meters).

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The main objective of this project being developed at SEGAL, in the framework of the SingSubs (collaboration with the National University of Singapore), is to develop integrated methods (software and hardware) for data acquisition (three-dimensional surfaces) in order to perform morphological analysis in sandy shore environments.

The comparison of three-dimensional surfaces acquired at different epochs permit to quantify erosion (or accumulation) rates, and consequently, to identify specific areas prone to suffer from catastrophic events in the future.

The particular work carried out in the framework of this thesis proposal has two main defined objectives:

- To develop an hardware device able to position large quantities of discrete points using GNSS data;
- To develop applications able to estimate the three-dimensional surfaces from the collected GNSS data.

Such objectives were established after the definition of the necessary criteria and constraints of the system that was necessary to follow:

- The device should be light and be able to move efficiently on the beach: no motorization was allowed due to practical or official restrictions;
- It should be constructed by integrating cheap components, in particular the attitude acquisition device that could be based on current normal Tablets/Smartphones instead of very expensive gyroscopes devices;
- The data should be obtained quickly, using kinematic approaches - the quality of the computation of the three-dimensional surface should be guaranteed by the number of acquired points in order to obtain accurate surfaces.

Although other systems to monitor such environments have been developed in the recent years like Terrestrial and Airborne LiDAR, Total Station surveying, all of them suffer of the several issues, as detailed in Chapter 1 and Chapter 2, that do not comply with the established criteria for our system.

The first part of the project was to develop the hardware system. The final adopted solution was based on a golf cart (tricycle) where a GNSS system is firmly coupled with a Tablet/Smartphone. All internal distances (GNSS antenna to the ground) and angles (attitude between Tablet/Smartphone and antenna mast) can be accurately determined and used in order to convert the acquired positions (using the GNSS antenna) into the ground on a common reference frame.

In order to perform such conversion, two applications were developed:

- *GyroData* - developed in Android, it permits to collect the Euler angles (Pitch, Roll and Yaw) through the orientation sensors of a Tablet or from a Smartphone. The reason to use Android was because it permits to acquire the attitude angles from the devices where is running easily. In addition, it is currently the most used platform in the world and it does not have associated costs to develop applications. The existing knowledge about development in Java also helped to select this platform.

- *Air2Ground* - developed in C#, it permits to transform the coordinates of the estimated points using the GNSS processing software from the base of the antenna to the ground using a rotation matrix. The reason to use C# was because it permits to develop robust applications in a very used Integrated Development Environment that was already known.

These two applications are also one of the essential components of the system developed in the framework of this work.

In addition, and in order to test the robustness of the developed system, several scripts based on GMT were also created. These scripts permit to generate the three-dimensional surfaces from the set of point positions observed with the system. The scripts are available in the annex. The existing time constraints to finish this component of the entire project did not permit to carry out all desired intensive tests, particularly in what concerns the sensitivity to temporal variation of the observed three-dimensional surfaces. In any case, several tests were already performed that permitted to conclude that the system is already functional and able to perform an accurate monitoring of the morphological evolution of a surface. The tests done that were analysed in this thesis were:

- To test the sensitivity and stability of the used Tablet/Smartphone;

- To investigate the influence of the Pitch and Roll errors in the correction of the coordinates.

- To compare the differences between two individual tracks. For this goal, the data was acquired in rough conditions: a steep asphalt terrain.

- To evaluate the performance of the Acquisition Data Device in a sandy beach environment by comparing the differences between two individual tracks.

The conclusions of these tests were significant:

- The stability of the acquisition of the attitude angles differ significantly between Tablet/Smartphones. It is fundamental to select a device that provides reliable and stable angles otherwise it introduces significant errors on the final coordinates.

- The influence in the initialisation of the attitude angles (Roll and Pitch) are not very significant since with a large error of 5, the differences are inferior to 3.5 cm in the vertical component.
- The differences between derived surfaces vary between 1-5 cm being most of the time inferior to 1 cm in good conditions. In more real conditions, we can obtain larger errors (up to decimetre level) but this can be essentially caused by undetected surface irregularities.

As final comment, this project provided a great satisfaction since I consider that I have contributed for the development of an efficient, inexpensive, and easy-to-use system that can be used to evaluate and quantify the erosion problems in sandy shore environments.

6.2 Future Work

No research project is ever completed... It is always possible to improve it. This is also true in this case. If I consider that the goals of the proposed work was fulfilled, there is space for further developments, namely:

- To perform more surveys over longer periods of time in the same area, in order to better evaluate the sensitivity of the system to temporal changes in the studied areas;
- The Data Acquisition Device can be also improved in order to improve its mobility in the sand. The use of larger wheels is a possibility to be investigated;
- The GMT scripts can be further developed in order to provide more output options in terms of plots. For instance, the creation of vertical profiles and also the optimisation of the area of interest. We have used the national datum of Portugal but, depending of the orientation of the beach, a different local coordinate system can be preferable.

In addition, but already out of scope of this dissertation, the development of tools to evaluate and visualize the volumetric variations over time will be essential to use this system efficiently.

References

[1] *Mark Tebboth*, Understanding intractable environmental policy conflicts: the case of the village that would not fall quietly into the sea., The Geographical Journal, 2013. 2

[2] *Mark Tebboth*, Adapting to coastal change: understanding different points of view in coastal erosion management., Accessed: 15/05/2013, [Online]. Available: http://blog.geographydirections.com/2013/11/01/adapting-to-coastal-change-understanding-different-po
2013. xiii, 2, 3

[3] *Remondi, B.W.*, Global Positioning System carrier phase: description and use., Bulletin Geodesique, vol. 59, pp. 361-377, 1985. 7

[4] *Remondi, B.W.*, Kinematic and pseudo-kinematic GPS, Proceedings of the Satellite Division Conference of the Institute of Navigation, Colorado Springs, pp. 21-23, 1988. 7

[5] *Cannon, M.E. and Schwarz, K.P.*, GPS semi-kinematic positioning along a well controlled traverse., Proceedings of the 5th International Geodetic Symposium on Satellite Positioning, Las Cruces, New Mexico., vol. 2, pp. 621-630, 1989. 7

[6] *IHO*, Manual on Hydrography., International Hydrographic Organization, vol. 1, 2005.

[7] *Kato, T., Terada, Y., Ito, K., Hattori, R., Abe, T., Miyake, T., Koshimura, S. and Nagai, T.*, Tsunami due to the 5 September 2004 off the Kii peninsula earthquake, Japan, recorded by a new GPS buoy., Earth, Planets Space, vol. 57, pp. 297-301, 2005.

[8] *Schone, T., Pandoe, W., Mudita, I., Roemer, S., Illigner, J., Zech, C. and Galas, R.*, GPS water level measurements for Indonesias Tsunami Early Warning System., Hazards Earth Syst. Sci., vol. 11, pp. 741-749, 2011.

[9] *Lao-Sheng Lin*, APPLICATION OF GPS RTK AND TOTAL STATION SYSTEM ON DYNAMIC MONITORING LAND USE., Department of Land Economics, National Chengchi University, 64, Section 2, Chihnan Road, Taipei 116, Taiwan, Republic of China. 7

[10] *David Sumner*, CHAPTER 7, Total Station Surveying, University Of Saskatchewan, pp. 249-276, 2012. 8

[11] *Meng, Nate Currit, Kaiguang Zhao*, Ground Filtering Algorithms for Airborne LiDAR Data: A Review of Critical Issues., Remote Sensing, 2010. 8

[12] *Meng, X., Wang, L., Silvan-Cardenas, J.L., Currit, N.*, A multi-directional ground filtering algorithm for airborne LIDAR., ISPRS J. Photogramm. Remote Sens., vol. 64, pp. 117-124, 2009. 8

[13] *Sithole, G.*, Filtering of laser altimetry data using a slope adaptive filter., Int. Arch. Photogramm. Remote Sens., pp. 203-210, 2001. 8

[14] *Liu, X.*, Airborne LiDAR for DEM generation: some critical issues., Prog. Phys. Geog., vol. 32, pp. 31-49, 2008. 8

[15] *Kukko, A.; Hyyppa, J.*, Small-footprint laser scanning simulator for system validation, error assessment, and algorithm development., Photogramm. Eng. Remote Sens., vol. 75, pp. 1177-1189, 2009. 8

[16] *Hill, J.M.; Graham, L.A.; Henry, R.J.; Cotter, D.M.; Ping, A.; Young, P.*, Wide-area topographic mapping and applications using airborne light detection and ranging (LiDAR) technology., Photogramm. Eng. Remote Sens., vol. 66, pp. 908-914, 2000. 8

[17] *Pfeifer, N.; Stadler, P.; Briese, C.*, Derivation of digital terrain models in the SCOP++ environment., Proceedings of OEEPE Workshop on Airborne Laser Scanning and Interferometric SAR for Digital Elevation Models, Stockholm, Sweden., vol. 66, pp. 908-914, 2001. 8

[18] *Vosselmann, G.*, Slope based filtering of Laser altimetry data., Int. Arch. Photogramm. Remote Sens., vol. 33, pp. 935-942, 2000. 8

[19] *Reutebuch, S.E.; Andersen, H.-E.; McGaughey, R.J.*, Light detection and ranging (LIDAR): an emerging tool for multiple resource inventory., J. Forest., vol. 103, pp. 286-292, 2001. 8

[20] *Baltsavias, E.*, A comparison between photogrammetry and laser scanning., ISPRS J. Photogramm. Remote Sens., vol. 54, pp. 83-94, 1999. 8, 9

[21] *Silvan-Cardenas, J.L.; Wang, L.*, A multi-resolution approach for filtering LiDAR altimetry data., ISPRS J. Photogramm. Remote Sens., vol. 61, pp. 11-22, 2006. 8

[22] *Habib, A.; Ghanma, M.; Morgan, M.; Al-Ruzouq, R.*, journal = Photogramm. Eng. Remote Sens., Photogrammetric and Lidar Data Registration Using Linear Features., vol. 71, pp. 699-707, 2005. 8

[23] *Lin, Y.; Mills, J.P.*, journal = Photogramm. Eng. Remote Sens., Factors influencing pulse width of small footprint, full waveform airborne laser scanning data., vol. 76, pp. 49-59, 2010. 8

[24] *Meng, X.; Wang, L.; Currit, N.*, Morphology-based building detection from airborne LIDAR data., Photogramm. Eng. Remote Sens., vol. 75, pp. 427-442, 2009. 9

[25] *Raber, G.T.; Jensen, J.R.; Schill, S.R.; Schuckman, K.*, Creation of Digital Terrain Models using an adaptive Lidar vegetation point removal process., Photogramm. Eng. Remote Sens., vol. 68, pp. 1307-1316, 2002. 9

[26] *Hodgson, M.E.; Jensen, J.; Raber, G.; Tullis, J.; Davis, B.A.; Thompson, G.; Schuckman, K.*, An evaluation of LiDAR-derived elevation and terrain slope in leaf-off condition., Photogramm. Eng. Remote Sens., vol. 71, pp. 817-823, 2005. 9

[27] *Chow, J.; Lichti, D.; Teskey, W.*, Accuracy Assessment of the Faro focus3D and Leica HDS6100 Panoramic Type Terrestrial Laser Scanner through Point-based and Plane-Based User Self-Calibration., Proceedings of the FIG Working Week: Knowing to Manage the Territory, Protect the Environment, Evaluate the Cultural Heritage, Rome, Italy., 2012. 9

[28] *Kraus, K., Pfeifer, N.*, Determination of terrain models in wooded areas with airborne laser scanner data., ISPRS Journal of Photogrammetry and Remote Sensing, vol. 53, pp. 193-203, 1998. 10

[29] *Kraus, K., Pfeifer, N.*, Advanced DTM generation from LIDAR data., International Archives of Photogrammetry and Remote Sensing, vol. 34, pp. 23-30, 2001. 10

[30] *K. Novak*, Mobile mapping technology for GIS data collection., Photogrammetric Engineering & Remote Sensing, vol. 61, pp. 493-501, 1995. 10

[31] *K.P. Schwarz, M.A.Chapman, M.W.Cannon, and P.Gong*, An integrated INS/GPS approach to the georeferencing of remotely sensed data., Photogrammetric Engineering & Remote Sensing, vol. 59, pp. 1667-1674, 1993. 10

[32] *T. Aussems, and M. Braess*, Mobile mapping using a surveying vehicle integrating GPS, wheel sensors and digital video cameras., Integrated Acquisition and Interpretation of Photogrammetric Data WS., Stuttgart, 1995. 10

[33] *E. Kortkov and R. Hoffman*, Terrain mapping for a walking planetary rover., IEEE Trans. Robotics and Automation., vol. 10, pp. 728-739, 1994. 10

[34] *C.E. Thorpe*, Vision and Navigation: The Carnegie Mellon Navlab., Kluwer Academic Publishing, vol. 10, pp. 728-739, 1990. 10

[35] *C. Vincent Tao*, Mobile Mapping Technology for Road Network Data Acquisition, Department of Geomatics Engineering,The University of Calgary, 2500 University Drive, NW, Calgary, Alberta, Canada T2N 1N4. 10

[36] *El-Sheimy, Chapman, M.A. and Tao, C.V.*, An Intelligent Mobile Mapping System., Proc. of Int'l Workshop on Mobile Mapping Technology, April 21-23, Thailand, part 1-2, 1-7, vol. 10, pp. 728-739, 1999. 10

[37] *Professional Surveyor Magazine*, Software Review: Trimble Business Center, 2011. 14

[38] *R. Miles.*, C# Programming., Department of Computer Science, University of Hull, 2011. 16, 27, 29, 30

[39] *B. Johnson.*, Professional Visual Studio 2012. ISBN: 978-1-118-33770-7, Paperback, pp. 1104, 2012. 27

[40] *Professor Doutor Luis Alexandre*, Engenharia de Software, 2008. 18

[41] *T. P. Systems*, Hiper Pro Operators Manual., Rev. C, Printed in USA, 2006. 23

[42] *L. H. Estey and C. M. Meertens*, TEQC: The Multi-Purpose Toolkit for GPS/GLONASS Data, John Wiley & Sons, vol. 3, pp. 42-49, 1999. 52

[43] *Fred Bermudez*, Software Review: Trimble Business Center, Professional Surveyor Magazine, [Online]. Available: http://www.profsurv.com/magazine/article.aspx?i=70991, 2011. 16

[44] *Physics Toolbox Gyroscope*, Accessed: 28/11/2013, [Online]. Available: https://play.google.com/store/apps/details?id=com.chrystianvieyra.android.physicstoolboxgyroscope. 11

[45] *Joe Sass*, GNSS or Total Station??Selecting the Right Tool for the Job, Accessed: 28/11/2013 [Online]. Available: http://machinecontrolonline.com/joe-sass/4903-gnss-or-total-station-selecting-the-right-tool-for-the-job

[46] *Sensor Display & Logging*, Accessed: 28/11/2013, [Online]. Available: https://play.google.com/store/apps/details?id=luxi.ENGO500.App.DispSensor. 12

[47] *Sensor Kinetics Pro*, Accessed: 28/11/2013, [Online]. Available: https://play.google.com/store/apps/details?id=com.innoventions.sensorkineticspro. 12

[48] *Data Sensors*, Accessed: 28/11/2013, [Online]. Available: https://play.google.com/store/apps/details?id=tfc.sensors. 12

[49] *Fused Linear Acceleration*, Accessed: 28/11/2013, [Online]. Available: https://play.google.com/store/apps/details?id=com.kircherelectronics.fusedlinearacceleration. 13

[50] *Sensor Log*, Accessed: 28/11/2013, [Online]. Available: https://play.google.com/store/apps/details?id=de.hs_osnabrueck.asta.sensorlog. 13

[51] *GPS Time*, Accessed: 28/11/2013, [Online]. Available: https://play.google.com/store/apps/details?id=net.sourcewalker.gpstime. 13

[52] *Getting Started with Android Studio*, Accessed: 15/12/2013, [Online]. Available: http://developer.android.com/sdk/installing/studio.html. 16

[53] *What is GMT?*, Accessed: 10/04/2013, [Online]. Available: http://gmt.soest.hawaii.edu/. 17

[54] *Cross Plataform, Open Source .NET Development Framework*, Accessed: 15/03/2014, [Online]. Available: http://mono-project.com/Main_Page.

[55] *ArrayList Class*, Accessed: 20/11/2013, [Online]. Available: http://developer.android.com/reference/java/util/ArrayList.html. 22

[56] *Ellen*, Stabilize compass of iPhone with gyroscope., Accessed: 06/10/2013, [Online]. Available: http://www.sundh.com/blog/2011/09/stabalize-compass-of-iphone-with-gyroscope/, 2011. 23

[57] *Instituto Geografico Portugues*, Informacao Tecnica, Sistemas de Referencia., Accessed: 06/10/2013, [Online]. Available: http://www.igeo.pt/produtos/Geodesia/inf_tecnica/sistemas_referencia/Datum_ETRS89.htm., 2013. 24

[58] *W. Gurtner.*, Rinex: The receiver independent exchange format version 3.10., Accessed: 14/03/2014, [Online]. Available: http://igscb.jpl.nasa.gov/igscb/data/format/rinex301.pdf, 2009. 52

[59] *Bernhard Hofmann-Wellenhof, Helmut Moritz*, Physical Geodesy, ISBN-10 3-211-33544-7, Springer, [Online]. Available: http://www.springer.com/earth+sciences+and+geography/geophysics/book/978-3-211-33544-4, 2006. 25

[60] *Lunardi, Alessandra*, Interpolation Theory, ISBN: 978-88-7642-342-0, Edizioni Della Normale. 30

[61] *Friesen, Jeff*, Beginning Java 7, Apress, 2011. 16

Appendix A

A.1 User Manual

In this section, a step by step description of how to use the developed system, is presented. Having any system able to collect, over any terrain, GNSS data and the antenna attitude with an Android device, it is possible to perform the correction described in this Dissertation.

A.1.1 Organization of the Manual

The user's manual consists of four sections: Acquire GNSS and Tablet/Smartphone Data, Data Processing, Correction Coordinates and Surfaces Computations.

- *Acquire GNSS and Tablet/Smartphone Data:* In this section, we explain how the GNSS data is collected using the device and how the Android Application "GyroData" stores the attitude values of the GNSS antenna(Euler Angles).

- *Data Processing:* This section explains how the GNSS data is processed in TBC, how the correction of the coordinates is done and the structure of the output files they are used as input to the "Air2Ground".

- *Coordinates Correction:* Here, an explanation about the coordinates correction application "Air2Ground", is presented.

- *Surfaces Computations:* In this section, we describe the steps to run the scripts to build the Surfaces.

A.2 Acquire GNSS and Tablet/Smartphone Data

In order to collect the GNSS and Tablet/Smartphone Data it is necessary to use the device shown in Figure A.1. Thus, GNSS and Tablet/Smartphone data are collected simultaneously.

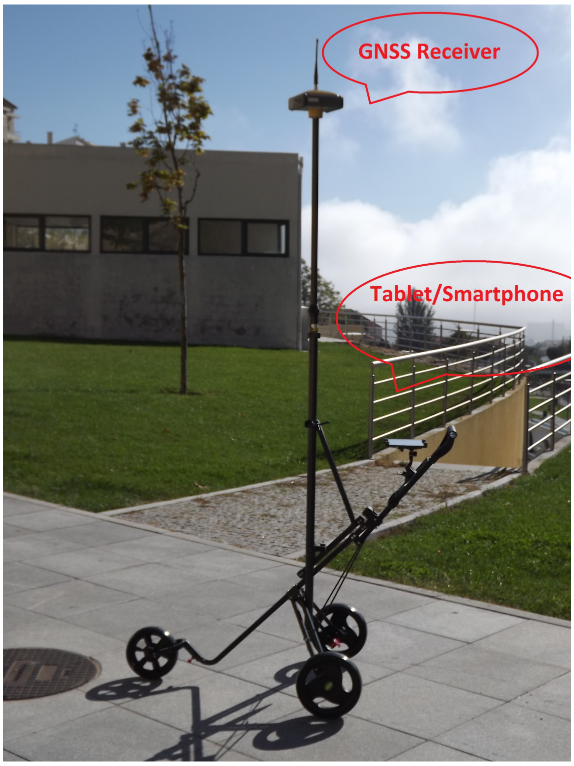
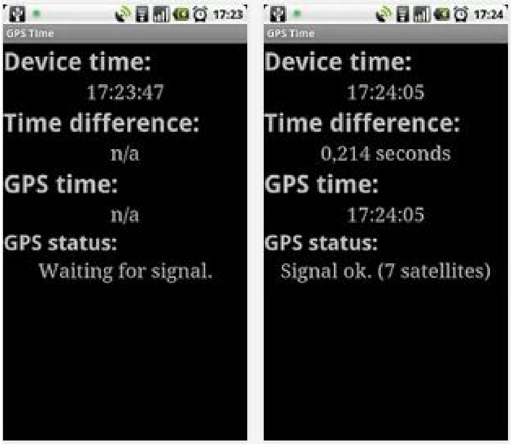


Figure A.1: Acquisition Device.

Before starting a survey, it is necessary to initialize in the Tablet/Smartphone the Pitch and Roll angles to 0. The next step is to, using Android Application "GPS Time" (Figure A.2),check the time difference between the Tablet/Smartphone and the GPS clock. It is also necessary to calculate the error of the Tablet/Smartphone sensor for the Pitch an Roll.

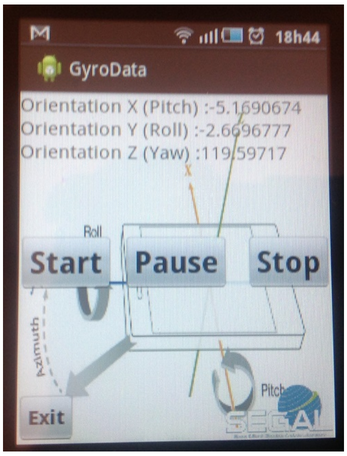


Device time:
17:23:47
Time difference:
n/a
GPS time:
n/a
GPS status:
Waiting for signal.

Device time:
17:24:05
Time difference:
0,214 seconds
GPS time:
17:24:05
GPS status:
Signal ok. (7 satellites)

Figure A.2: GPS Time Application.

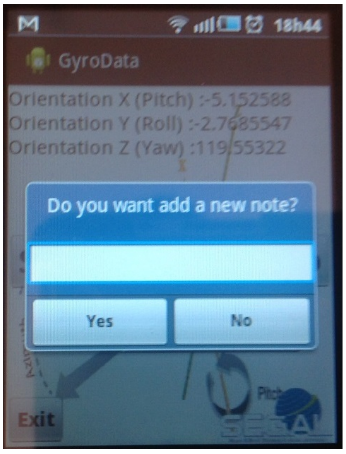
At the beginning of each survey, it is necessary to start the Android Application "GyroData" (Figure A.3) in order to have the correct Euler Angle assigned to the right GNSS point. Since the data collection stops when the Tablet/Smartphone screen is suspended, it is necessary to change the devices settings in order to guarantee we have data for the entire survey.



GyroData

Orientation X (Pitch) :-5.1690674
Orientation Y (Roll) :-2.6696777
Orientation Z (Yaw) :119.59717

Figure A.3: GyroData Application.



GyroData

Orientation X (Pitch) :-5.152588
Orientation Y (Roll) :-2.7685547
Orientation Z (Yaw) :119.55322

Do you want add a new note?

Figure A.4: GyroData Application "Pause" Option.

To start the survey and collect the Euler Angles, press the "Start" button (Figure A.3). To register a note, press the button "Pause"(Figure A.3) the menu shown in Figure A.4 will appear, write a note and then pressing the "Yes" button the note is recorded and the survey continues. If the button "Pause" is pressed and there is no need to add a note, press "No" and the survey will continue. To end the survey press the button "Stop"(Figure A.3)and the button "Exit"(Figure A.3) closes the application.

Figure A.5 shows the Topcon receiver "Hyper Pro" used to collect the GNSS data.

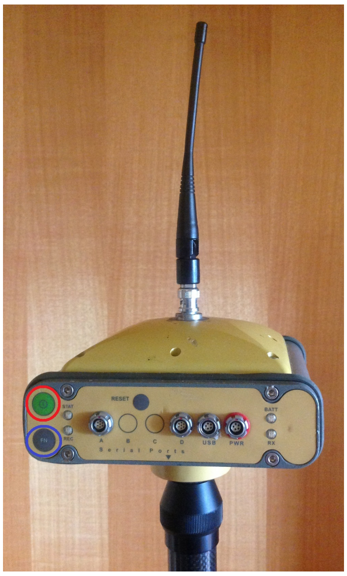
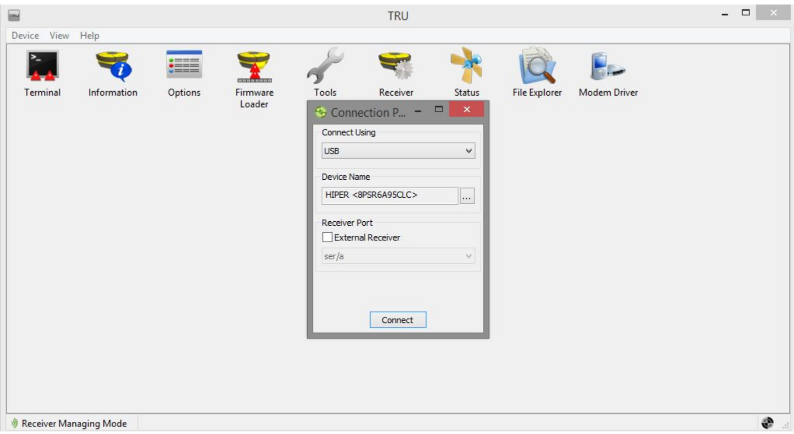


Figure A.5: Topco Hiper Pro Receiver.

To switch the device on, press the button that has the red circle. Before starting the survey, the receiver must be acquiring satellites signals and position for 2/3 minutes. To start the survey, press the button that has the blue circle and it will turn green stating that is collecting data. To finish the survey press the same button until the green light turns off.

To end a survey and to ensure that all GNSS points collected have their Euler angles, finish the GNSS survey first and then the "GyroData" application. At the end of the survey write down the time difference of the Android application "GPS Time"(Figure A.2) again.

After the survey, the acquired data with "HyperPro "must be downloaded using Topcon Receiver Utility (TRU). To do that, the receiver is connected to a computer using a USB or Serial cable.



Connect Using: USB

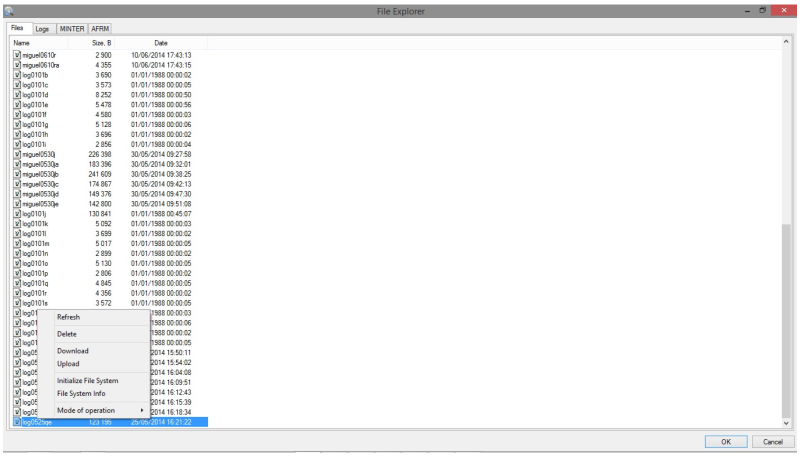
Device Name: HIPER <8PSR6A95CLC>

Receiver Port

☐ External Receiver

ser/a

Figure A.6: Connect to Hiper Pro.



Name	Size, B	Date
miguel0610r	2 900	10/06/2014 17:43:13
miguel0610ra	4 355	10/06/2014 17:43:15
log0101b	3 690	01/01/1988 00:00:02
log0101c	3 573	01/01/1988 00:00:05
log0101d	8 252	01/01/1988 00:00:50
log0101e	5 478	01/01/1988 00:00:56
log0101f	4 580	01/01/1988 00:00:03
log0101g	5 128	01/01/1988 00:00:06
log0101h	3 696	01/01/1988 00:00:02
log0101i	2 856	01/01/1988 00:00:04
miguel0530j	226 398	30/05/2014 09:27:58
miguel0530ja	183 396	30/05/2014 09:32:01
miguel0530jb	241 609	30/05/2014 09:38:25
miguel0530jc	174 867	30/05/2014 09:42:13
miguel0530jd	149 376	30/05/2014 09:47:30
miguel0530je	142 800	30/05/2014 09:51:08
log0101j	130 841	01/01/1988 00:45:07
log0101k	5 092	01/01/1988 00:00:03
log0101l	3 699	01/01/1988 00:00:02
log0101m	5 017	01/01/1988 00:00:05
log0101n	2 899	01/01/1988 00:00:02
log0101o	5 130	01/01/1988 00:00:05
log0101p	2 806	01/01/1988 00:00:02
log0101q	4 845	01/01/1988 00:00:05
log0101r	4 356	01/01/1988 00:00:02
log0101s	3 572	01/01/1988 00:00:05
log01		1988 00:00:03
log01		1988 00:00:06
log01		1988 00:00:02
log01		1988 00:00:05
log05		2014 15:50:11
log05		2014 15:54:02
log05		2014 16:04:08
log05		2014 16:09:51
log05		2014 16:12:43
log05		2014 16:15:39
log05		2014 16:18:34
log0525qe	123 195	25/05/2014 16:21:22

Figure A.7: Download GNSS Data from TRU.

To connect the receiver to the computer, perform the following steps (see Figure A.6), "Device -> Connect -> Select USB -> Ok." To download the collected data, do the following steps (see Figure A.7), "Select File Explorer -> Select the files GNSS -> Click to the rigth mouse button -> Click Download."

When the GNSS data download is complete, files must be converted to "RINEX" (Receiver Independent Exchange Format)observation files. These files contain various information and a full description can be found in [58]. To do this transformation, the program teqc is used, by typing the command: "teqc" [42] using the command "teqc O.mov-1 input_file> output_file.obs" on a Linux machine.

A.3 Data Processing

This section explains how to process GNSS data in Trimble Business Center (TBC), how to run the scripts to convert the coordinates and also how the verification of all data files that will be used as input in the coordinate corrections application "Air2Ground", is made.

A.3.1 Trimble Business Center

In this section, all the necessary steps to process the GNSS data in TBC are presented. To start, create a new blank project(Figure A.8/A.9) following the steps, "File -> New Project -> Ok." Then, in order to have a more precise position, import the region permanent GNSS station files, and also import the collected data Rinex file (see Figure A.10). To do this, follow these steps: "File -> Import -> Select Files to Import -> Ok."

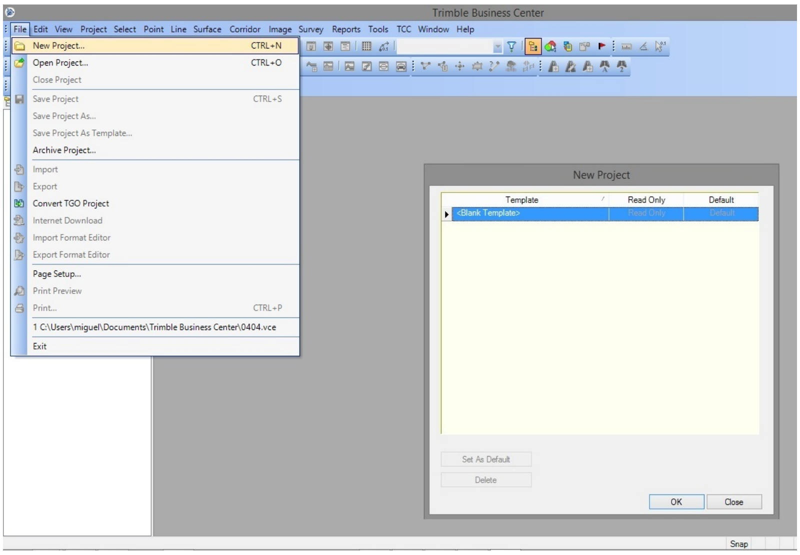


Figure A.8: Create New Project in TBC.

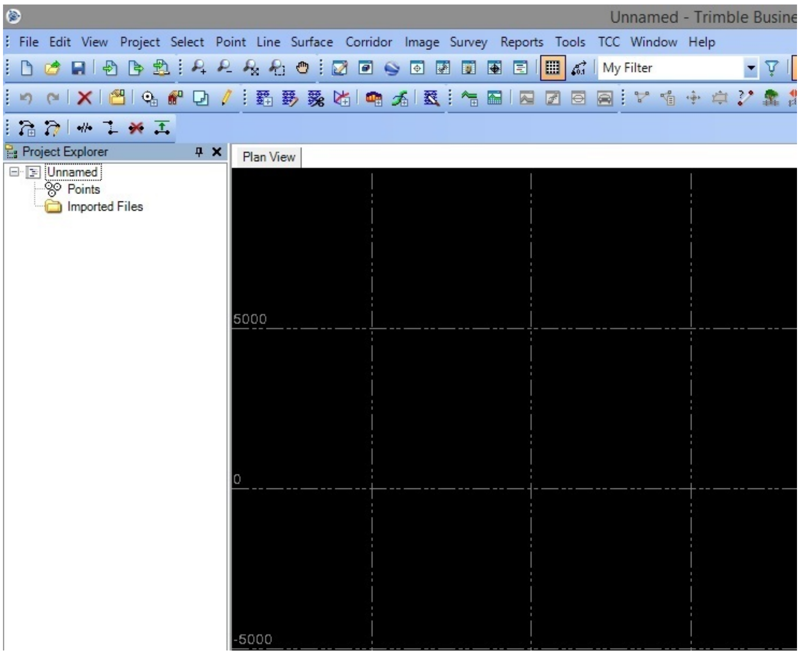
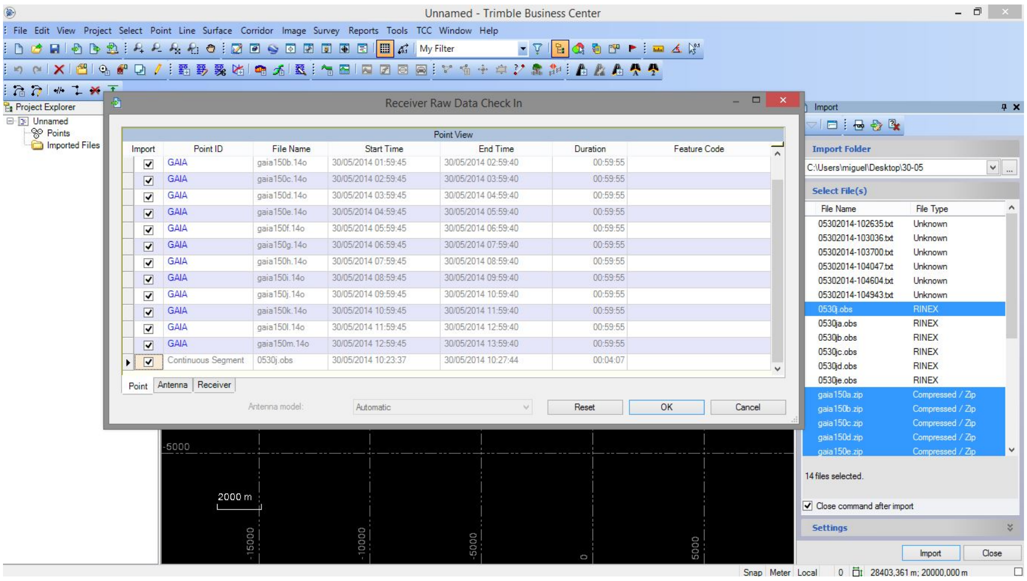

Unnamed - Trimble Busine
File Edit View Project Select Point Line Surface Corridor Image Survey Reports Tools TCC Window Help
My Filter
Project Explorer
Unnamed
Points
Imported Files
Plan View
5000
0
-5000

Figure A.9: The New TBC Project.


Unnamed - Trimble Business Center
File Edit View Project Select Point Line Surface Corridor Image Survey Reports Tools TCC Window Help
My Filter
Project Explorer
Unnamed
Points
Imported Files
-5000
2000 m
-15000
-10000
-5000
0
5000
Snap Meter Local 0 28403,361 m; 20000,000 m
Import
Import Folder
Select File(s)
Settings
Import
Close

Receiver Raw Data Check In

Point View

Import	Point ID	File Name	Start Time	End Time	Duration	Feature Code
☑	GAIA	gaia150b.14o	30/05/2014 01:59:45	30/05/2014 02:59:40	00:59:55	
☑	GAIA	gaia150c.14o	30/05/2014 02:59:45	30/05/2014 03:59:40	00:59:55	
☑	GAIA	gaia150d.14o	30/05/2014 03:59:45	30/05/2014 04:59:40	00:59:55	
☑	GAIA	gaia150e.14o	30/05/2014 04:59:45	30/05/2014 05:59:40	00:59:55	
☑	GAIA	gaia150f.14o	30/05/2014 05:59:45	30/05/2014 06:59:40	00:59:55	
☑	GAIA	gaia150g.14o	30/05/2014 06:59:45	30/05/2014 07:59:40	00:59:55	
☑	GAIA	gaia150h.14o	30/05/2014 07:59:45	30/05/2014 08:59:40	00:59:55	
☑	GAIA	gaia150i.14o	30/05/2014 08:59:45	30/05/2014 09:59:40	00:59:55	
☑	GAIA	gaia150j.14o	30/05/2014 09:59:45	30/05/2014 10:59:40	00:59:55	
☑	GAIA	gaia150k.14o	30/05/2014 10:59:45	30/05/2014 11:59:40	00:59:55	
☑	GAIA	gaia150l.14o	30/05/2014 11:59:45	30/05/2014 12:59:40	00:59:55	
☑	GAIA	gaia150m.14o	30/05/2014 12:59:45	30/05/2014 13:59:40	00:59:55	
☑	Continuous Segment	0530j.obs	30/05/2014 10:23:37	30/05/2014 10:27:44	00:04:07	

Point | Antenna | Receiver

Antenna model: Automatic

Reset | OK | Cancel

Import Folder: C:\Users\miguel\Desktop\30-05

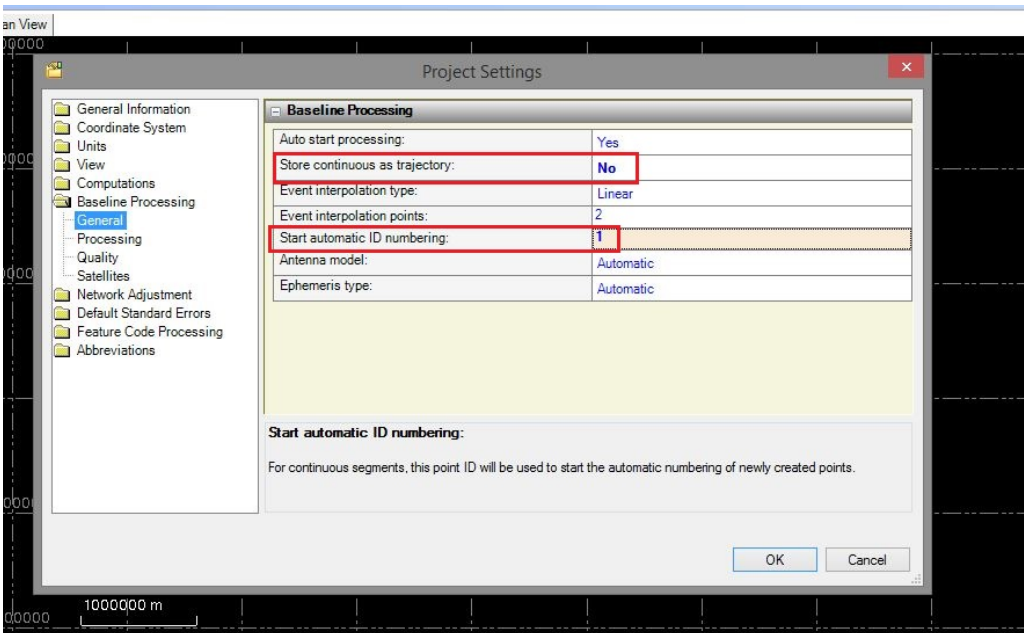
File Name	File Type
05302014-102635.txt	Unknown
05302014-103036.txt	Unknown
05302014-103700.txt	Unknown
05302014-104047.txt	Unknown
05302014-104604.txt	Unknown
05302014-104943.txt	Unknown
0530j.obs	RINEX
0530ja.obs	RINEX
0530jb.obs	RINEX
0530jc.obs	RINEX
0530jd.obs	RINEX
0530je.obs	RINEX
gaia150a.zip	Compressed / Zip
gaia150b.zip	Compressed / Zip
gaia150c.zip	Compressed / Zip
gaia150d.zip	Compressed / Zip
gaia150e.zip	Compressed / Zip

14 files selected.

☑ Close command after import

Figure A.10: Import Files.

After all the necessary files are imported, change the options marked with a red rectangle to "No" and "1" see Figure A.11 and do the steps: "Project -> Project Settings - > BaseLineProcessing -> General -> Ok. "



Baseline Processing

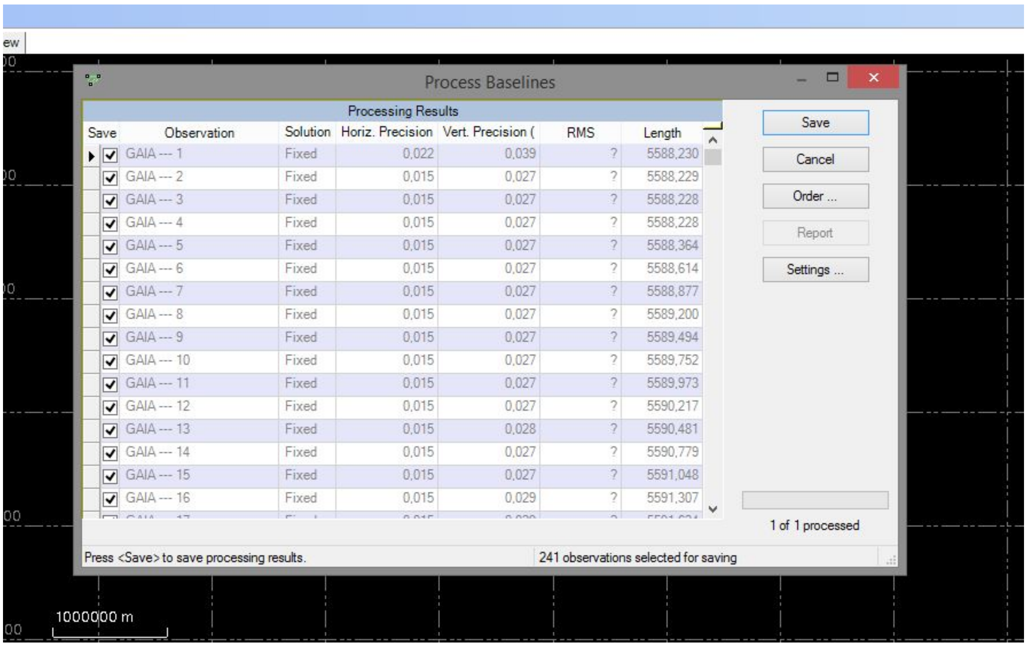
Auto start processing:	Yes
Store continuous as trajectory:	No
Event interpolation type:	Linear
Event interpolation points:	2
Start automatic ID numbering:	1
Antenna model:	Automatic
Ephemeris type:	Automatic

Start automatic ID numbering:

For continuous segments, this point ID will be used to start the automatic numbering of newly created points.

Figure A.11: Project Settings.

Finally, to process the data do: "Survey -> Process Baselines -> Save" (see Figures A.12, A.13, A.14).



Process Baselines

Processing Results

Save	Observation	Solution	Horiz. Precision	Vert. Precision (	RMS	Length
☑	GAIA --- 1	Fixed	0,022	0,039	?	5588,230
☑	GAIA --- 2	Fixed	0,015	0,027	?	5588,229
☑	GAIA --- 3	Fixed	0,015	0,027	?	5588,228
☑	GAIA --- 4	Fixed	0,015	0,027	?	5588,228
☑	GAIA --- 5	Fixed	0,015	0,027	?	5588,364
☑	GAIA --- 6	Fixed	0,015	0,027	?	5588,614
☑	GAIA --- 7	Fixed	0,015	0,027	?	5588,877
☑	GAIA --- 8	Fixed	0,015	0,027	?	5589,200
☑	GAIA --- 9	Fixed	0,015	0,027	?	5589,494
☑	GAIA --- 10	Fixed	0,015	0,027	?	5589,752
☑	GAIA --- 11	Fixed	0,015	0,027	?	5589,973
☑	GAIA --- 12	Fixed	0,015	0,027	?	5590,217
☑	GAIA --- 13	Fixed	0,015	0,028	?	5590,481
☑	GAIA --- 14	Fixed	0,015	0,027	?	5590,779
☑	GAIA --- 15	Fixed	0,015	0,027	?	5591,048
☑	GAIA --- 16	Fixed	0,015	0,029	?	5591,307

1 of 1 processed

Press <Save> to save processing results.

241 observations selected for saving

Figure A.12: GNSS data processing.

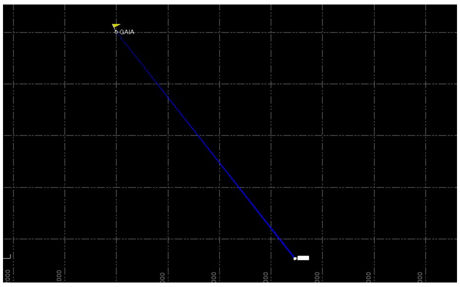


Figure A.13: GNSS data processing results.

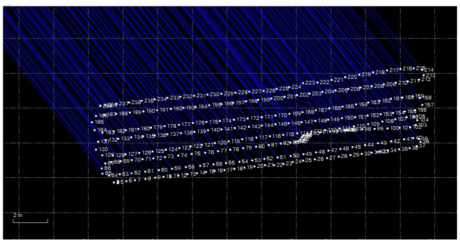
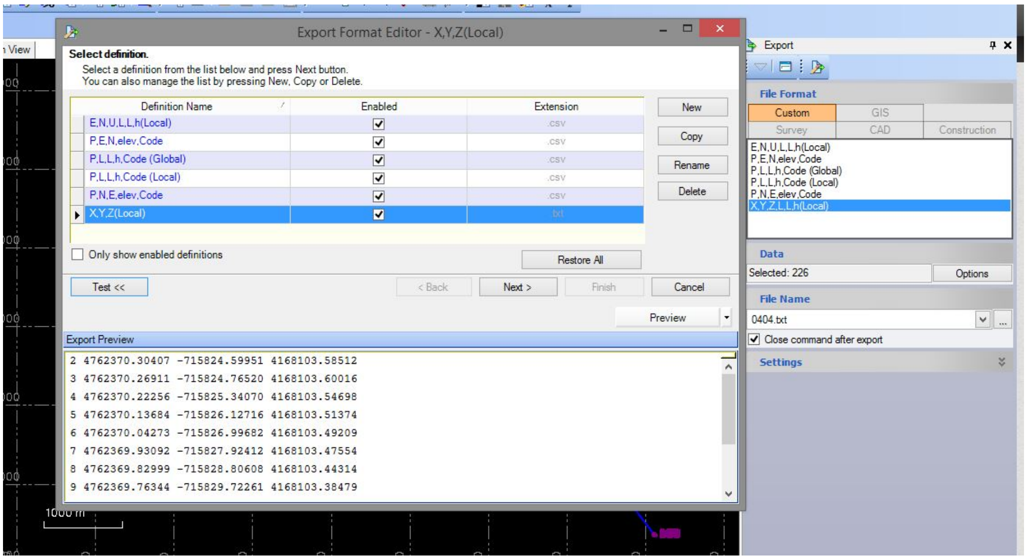


Figure A.14: GNSS data processing results.

After the data processing, it has to be exported by performing the following steps: "File -> Export -> Export Format Edit -> Select the Output File Format -> Select Directory to Save Output File -> Export" (Figure A.15).



Export Format Editor - X,Y,Z(Local)

Select definition.

Select a definition from the list below and press Next button.
You can also manage the list by pressing New, Copy or Delete.

Definition Name	Enabled	Extension
E,N,U,L,L,h(Local)	☑	.csv
P,E,N,elev,Code	☑	.csv
P,L,L,h,Code (Global)	☑	.csv
P,L,L,h,Code (Local)	☑	.csv
P,N,E,elev,Code	☑	.csv
X,Y,Z(Local)	☑	.txt

☐ Only show enabled definitions

Export Preview

```
2 4762370.30407 -715824.59951 4168103.58512
3 4762370.26911 -715824.76520 4168103.60016
4 4762370.22256 -715825.34070 4168103.54698
5 4762370.13684 -715826.12716 4168103.51374
6 4762370.04273 -715826.99682 4168103.49209
7 4762369.93092 -715827.92412 4168103.47554
8 4762369.82999 -715828.80608 4168103.44314
9 4762369.76344 -715829.72261 4168103.38479
```

Export

File Format: Custom

E,N,U,L,L,h(Local)
P,E,N,elev,Code
P,L,L,h,Code (Global)
P,L,L,h,Code (Local)
P,N,E,elev,Code
X,Y,Z,L,L,h(Local)

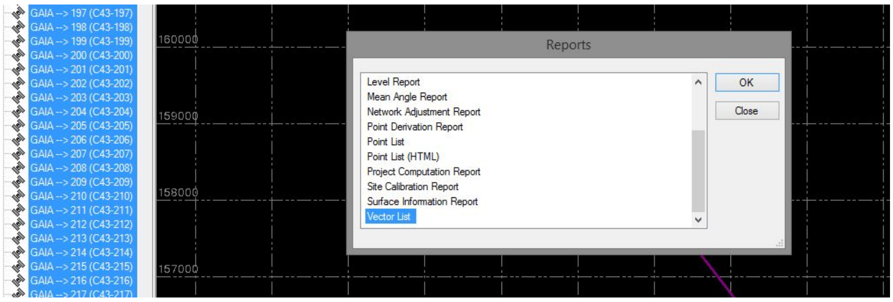
Data: Selected: 226

File Name: 0404.txt

☑ Close command after export

Figure A.15: Export Data.

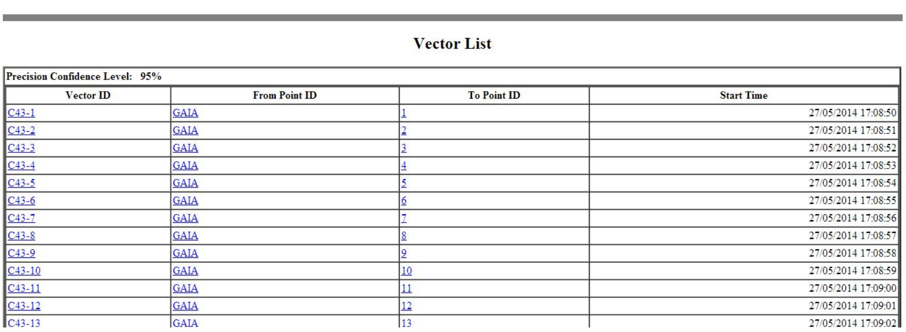
To complete the process it is also necessary to store the time of each observed GNSS point. To do that go to: "Report -> VectorList -> Ok", a window will open in the browser and then you can copy the data to a .txt file. See Figures A.16 and A.17.



Reports

Level Report
Mean Angle Report
Network Adjustment Report
Point Derivation Report
Point List
Point List (HTML)
Project Computation Report
Site Calibration Report
Surface Information Report
Vector List

Figure A.16: Export VectorList.



Vector List

Precision Confidence Level: 95%

Vector ID	From Point ID	To Point ID	Start Time
C43-1	GAIA	1	27/05/2014 17:08:50
C43-2	GAIA	2	27/05/2014 17:08:51
C43-3	GAIA	3	27/05/2014 17:08:52
C43-4	GAIA	4	27/05/2014 17:08:53
C43-5	GAIA	5	27/05/2014 17:08:54
C43-6	GAIA	6	27/05/2014 17:08:55
C43-7	GAIA	7	27/05/2014 17:08:56
C43-8	GAIA	8	27/05/2014 17:08:57
C43-9	GAIA	9	27/05/2014 17:08:58
C43-10	GAIA	10	27/05/2014 17:08:59
C43-11	GAIA	11	27/05/2014 17:09:00
C43-12	GAIA	12	27/05/2014 17:09:01
C43-13	GAIA	13	27/05/2014 17:09:02

Figure A.17: VectorList Example.

A.3.2 Convert Coordinates

The TBC output file is in XYZ ECEF coordinates and it is necessary to convert them to local ENU coordinates. To do so, run the "xyz2enu.c" on a Windows, Linux or MacOs terminal, however there may be some compatibility issues between these different type of files.

First, compile the script using the following command "gcc xyz2enu.c -o xyz2enu-lm". After that, run the compiled script with the command "./xyz2enu xyz_file.txt enu_file.txt 2.??". The file "xyz_file.txt" is the file with the XYZ coordinates, the file "enu_file.txt" is the file where the ENU converted coordinates are stored and "2.??" is the height of the mast plus the it's distance to the ground. These three parameters are necessary to run the script.

To do the reverse coordinate conversion, compile the script "enu2xyz.c" using the following command "gcc enu2xyz.c -o enu2xyz-lm". After that run the compiled script with the command "./enu2xyz enu_file.txt xyz_file.txt". The file "enu_file.txt" is the file with the ENU corrected coordinates in the Air2Groud application and the file "xyz_file.txt" is the file where the converted coordinates in XYZ are stored.

A.3.3 Verification of GNSS and Tablet/Smartphone Data

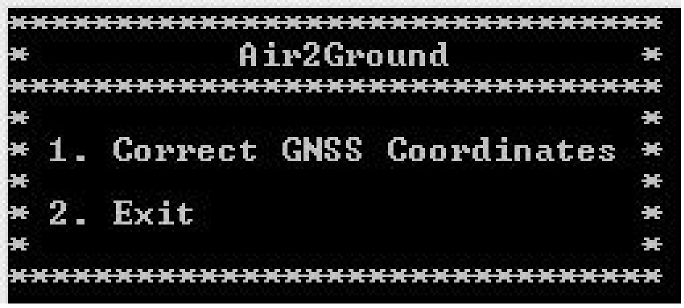
Before correcting the coordinates, always check the files that will be used as input in Air2Ground application. In the Tablet/Smartphone data, alway check for the presence of mistaken values like: "23.097E-10". Delete those mistaken characters, and the data must be something like: "23,097". It is also necessary to delete the lines with the "notes" inserted during the survey. Regarding the TBC output and the coordinates conversion files, it is necessary to verify if all values are correct ie, if it has no character or corrupt data that may affect the corrections of the coordinates in Air2Ground application.

A.4 Correction Coordinates

When GNSS and the Tablet/Smartphone data are in the correct format, the next step is to use the "Air2Ground" application to perform the coordinates correction. In the next chapter, we present all the procedures necessary to perform the accurate correction of the acquired GNSS data.

A.4.1 Air2Ground

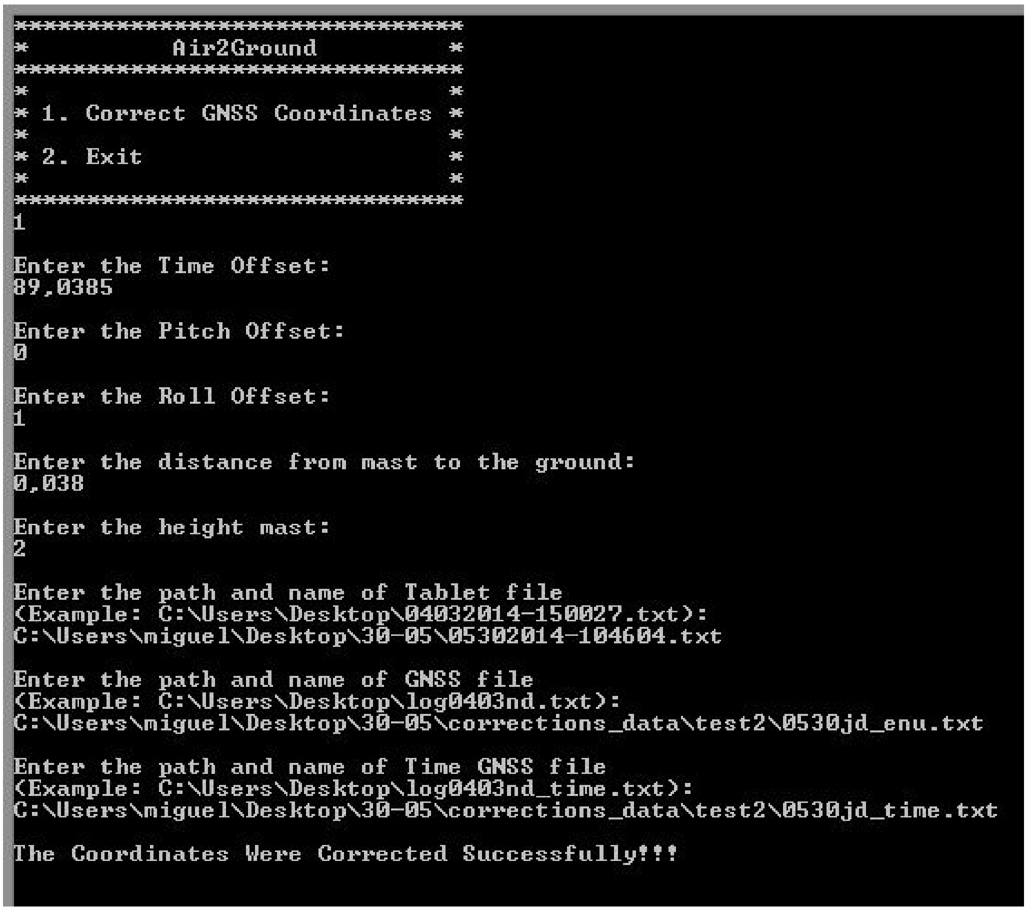
To run this application click 2 times on the executable file named "Air2Ground.exe" and a console with the menu shown in Figure A.18 will open.



```
**********************************
*          Air2Ground            *
**********************************
*                                *
* 1. Correct GNSS Coordinates    *
*                                *
* 2. Exit                        *
*                                *
**********************************
```

Figure A.18: Air2Ground Application.

In Figure A.18 two options are presented. The first option is to perform the coordinates corrections and the second is to exit the application. Figure A.19 shows all the necessary steps to complete the corrections.



```
**********************************
*          Air2Ground            *
**********************************
*                                *
* 1. Correct GNSS Coordinates    *
*                                *
* 2. Exit                        *
*                                *
**********************************
1

Enter the Time Offset:
89,0385

Enter the Pitch Offset:
0

Enter the Roll Offset:
1

Enter the distance from mast to the ground:
0,038

Enter the height mast:
2

Enter the path and name of Tablet file
(Example: C:\Users\Desktop\04032014-150027.txt):
C:\Users\miguel\Desktop\30-05\05302014-104604.txt

Enter the path and name of GNSS file
(Example: C:\Users\Desktop\log0403nd.txt):
C:\Users\miguel\Desktop\30-05\corrections_data\test2\0530jd_enu.txt

Enter the path and name of Time GNSS file
(Example: C:\Users\Desktop\log0403nd_time.txt):
C:\Users\miguel\Desktop\30-05\corrections_data\test2\0530jd_time.txt

The Coordinates Were Corrected Successfully!!!
```

Figure A.19: Air2Ground Application Parameters.

Each of the options inserted, like we can see in Figure A.19 are the parameters that the application need to perform the corrections of the GNSS coordinates. The application starts by asking the Time Offset; The offset of the Pitch and Roll; The distance from the mast to the ground and the height of the mast (the sum of these two parameters will be the GNSS antenna height) and finally the files with the attitude angles(from the Tablet/Smartphone); the file with the GNSS coordinates and the file with the time of observation.

When all the correct parameters are inserted, the application will run and the following message will show: "The Coordinates Were Corrected Successfully!!!". The results are stored in the same directory where the input files are. The file with the corrections will have the name of the input file with the GNSS coordinates plus the following text "_resultfile.txt" and the file where you can check which Euler angles dataset were assigned to each GNSS point will have the name of the input file plus the following text "_TabletGPSformat.txt".

A.5 Surfaces Computations

To create the surfaces it is necessary to have two files with GNSS data collected in a survey. To compare two surfaces it is necessary to define what will be the Surface 1 and Surface 2, defining two files ".txt" with the name "d1a" and "d1b". Then, run the following set of commands:

- txt2srf d1a 0.5
- txt2srf d1b 0.5
- create_surf
- surf_diff.gmt
- surf_all.gmt

Before running "create_surf" it is necessary to define the dimensions of the grid, the gray scale values and altitude scale to plot the surface. These parameters are stored in "parameters.lst" file (grid dimensions) and "GMT_abso.cpt" file (gray scale and altitude values).

Finally, open "OUTDIR" and "OUTDIFF" directories to check if the surfaces were well created.

Appendix B

B.1 GMT Scripts

B.1.1 txt2srf

```
#!/bin/tcsh

if ($#argv != 2) then
  echo 'Usage:    txt2grd file[.txt] (no extension) tensionvalue[0-1]'
  echo 'Example:  txt2grd file.txt 0.5'
  echo 'Purpose:  converts from txt to grd'
  exit 1
endif

set txt  = $1.txt
set xyz  = XYZ/$1.xyz
set llh  = LLH/$1.llh
set enu  = ENU/$1.enu
set blm  = ENU/$1.bm.enu
set grda = GRD/$1.a.grd
set grdc = GRD/$1.c.grd
set grdi = GRD/$1.i.grd
set srfa = GRD/$1.a.srf
set srfc = GRD/$1.c.srf
set srfi = GRD/$1.i.srf

if (! -e $txt) then
  echo $txt does not exist in the current directory
  exit -1
 endif

set tens = $2

# Use proj to compute the positions in PT-TM06/ETRS89
cat $txt | awk '{print($2,$3,$4)}' > $xyz
cat $xyz | xyz2gd - | awk '{print($1,$4,$3)}' > $llh
cat $llh | proj +proj=tmerc +lat_0=39.66825833333333 +lon_0
    =-8.133108333333334 +k=1 +x_0=0 +y_0=0 +ellps=GRS80 +towgs84
    =0,0,0,0,0,0,0 +units=m +no_defs -r > $enu
```

```
# Determine Limits and giving an extra meter
set e_min = ‘sort -nk1 $enu | head -1 | awk ’{printf("%.0f",$1-1)}’‘
set e_max = ‘sort -nk1 $enu | tail -1 | awk ’{printf("%.0f",$1+1)}’‘
set n_min = ‘sort -nk2 $enu | head -1 | awk ’{printf("%.0f",$2-1)}’‘
set n_max = ‘sort -nk2 $enu | tail -1 | awk ’{printf("%.0f",$2+1)}’‘
echo Computed Limits = $e_min/$e_max/$n_min/$n_max

set LIME = ‘grep AreaE parameters.lst | awk ’{printf($2)}’‘
#xyz2grd $enu -G$grd -I0.25/0.25 -R$LIM
blockmean $enu -R$LIME -I0.4 > $blm
xyz2grd $blm -R$LIME -I0.4 -G$grda
surface $blm -R$LIME -I0.2 -G$srfa -T$tens

set LIMI = ‘grep AreaI parameters.lst | awk ’{printf($2)}’‘
echo Area of Interest = $LIMI
#grdsample $grd -G$gr_ -I0.25/0.25 -R$LIM
grdcut $srfa -G$srfc -R$LIMI
grdcut $grda -G$grdc -R$LIMI

blockmean $enu -R$LIMI -I0.5 > $blm
xyz2grd $blm -R$LIMI -I0.4 -G$grdi
surface $blm -R$LIMI -I0.2 -G$srfi -T$tens
```

B.1.2 create_surf

```
#! /bin/tcsh

set DIRS = "XYZ LLH ENU GRD DIFF"
foreach d ($DIRS)
  rm -Rf $d >& /dev/null
  mkdir $d >& /dev/null
 end

set SURVEYS = "d1a d1b"
set TYPES   = "a c i"
set TENSORS = "0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 "
set TENSORS = "0.75"

foreach tens ($TENSORS)
  foreach surv1 ($SURVEYS)
    echo Creating Surface for survey: $surv1
    txt2srf $surv1 $tens
   end
```

```
foreach t ($TYPES)
  grdmath GRD/d1a.$t.srf GRD/d1b.$t.srf ADD 2 DIV = GRD/d1m.$t.srf
  grdmath GRD/d1a.$t.grd GRD/d1b.$t.grd ADD 2 DIV = GRD/d1m.$t.grd
 end

set SURVEYS = "d1a d1b d1m"
echo ""
echo "#####################################"
echo "Differences between Cut (c) and Internal (i) Surfaces "
echo " "
foreach surv1 ($SURVEYS)
  echo Surveys: $surv1
  grdmath GRD/$surv1.c.srf GRD/$surv1.i.srf SUB 100 MUL = DIFF/$surv1.
      ci.srf
  grdmath DIFF/$surv1.ci.srf STD = DIFF/$surv1.ci.std
  set std = ‘grd2xyz DIFF/$surv1.ci.std | tail -1 | awk ’{print($3)}’‘
  set min_max = ‘grdinfo DIFF/$surv1.ci.srf | grep z_min | awk ’{print
      ($3," ",$5)}’‘
  echo $std $min_max | awk ’{printf("Std: %0.1f Min: %0.1f Max: %0.1f\
      n",$1,$2,$3)}’
 end

set SURVEYS = "d1a d1b"
echo ""
echo "#####################################"
echo "Statistics between All(a), Cut(c) and Internal(i) Surfaces "
echo ""
foreach surv1 ($SURVEYS)
  echo Surveys: $surv1 and Average "(d1m)"
  foreach t ($TYPES)
    grdmath GRD/$surv1.$t.srf GRD/d1m.$t.srf SUB 100 MUL = DIFF/$surv1
        .d1m.$t.srf
    grdmath DIFF/$surv1.d1m.$t.srf STD = DIFF/$surv1.d1m.$t.std
    set std = ‘grd2xyz DIFF/$surv1.d1m.$t.std | tail -1 | awk ’{print(
        $3)}’‘
    set min_max = ‘grdinfo DIFF/$surv1.d1m.$t.srf | grep z_min | awk
        ’{print($3," ",$5)}’‘
    echo $t $std $min_max | awk ’{printf("Type: %1s - Std: %0.1f Min:
        %0.1f Max: %0.1f\n",$1,$2,$3,$4)}’

    grdmath GRD/$surv1.$t.grd GRD/d1m.$t.grd SUB 100 MUL = DIFF/$surv1
        .d1m.$t.grd
   end
 end

echo ""
```

```
 echo "#####################################"
 echo " Statistics between Surveys for Different All(a), Cut(c) and Internal(i) Surfaces "
 echo " "
 foreach surv1 ($SURVEYS)
   foreach surv2 ($SURVEYS)
     set menor = `echo $surv1 $surv2 | awk '{if ($1 < $2) print(1); else print(0)}'`
     if ($menor != 0) then
       echo Surveys: $surv1 $surv2
       foreach t ($TYPES)
         grdmath GRD/$surv1.$t.srf GRD/$surv2.$t.srf SUB 100 MUL = DIFF/$surv1.$surv2.$t.srf
         grdmath DIFF/$surv1.$surv2.$t.srf STD = DIFF/$surv1.$surv2.$t.std
         set std = `grd2xyz DIFF/$surv1.$surv2.$t.std | tail -1 | awk '{print($3)}'`
         set min_max = `grdinfo DIFF/$surv1.$surv2.$t.srf | grep z_min | awk '{print($3," ",$5)}'`
         echo $t $std $min_max | awk '{printf("Type: %1s - Std: %0.1f Min: %0.1f Max: %0.1f\n",$1,$2,$3,$4)}'
         echo $tens $std $min_max | awk '{printf(" %0s %5.1f %5.1f %5.1f TENSOR\n",$1,$2,$3,$4)}'

         grdmath GRD/$surv1.$t.grd GRD/$surv2.$t.grd SUB 100 MUL = DIFF/$surv1.$surv2.$t.grd
        end
      endif
    end
  end
end
```

B.1.3 suf_diff

```
#! /bin/tcsh

set LIMI = `grep AreaI parameters.lst | awk '{printf($2)}'`
set LIME = `grep AreaE parameters.lst | awk '{printf($2)}'`

set e_min = `grep Xmini parameters.lst | awk '{printf($2)}'`
set e_max = `grep Xmaxi parameters.lst | awk '{printf($2)}'`
set n_min = `grep Ymini parameters.lst | awk '{printf($2)}'`
set n_max = `grep Ymaxi parameters.lst | awk '{printf($2)}'`
set cola = `grep Color_a parameters.lst | awk '{printf($2)}'`
set colb = `grep Color_b parameters.lst | awk '{printf($2)}'`



echo $e_min $n_min >  $TMPDIR/square.$$
echo $e_min $n_max >> $TMPDIR/square.$$
echo $e_max $n_max >> $TMPDIR/square.$$
echo $e_max $n_min >> $TMPDIR/square.$$
echo $e_min $n_min >> $TMPDIR/square.$$



rm -Rf OUTDIFF >& /dev/null
mkdir OUTDIFF >& /dev/null

set SURVEYS = "d1a d1b d1m"
set TYPES   = "a c i"

echo ""
echo "##################################################"
echo ""
foreach surv1 ($SURVEYS)
  if ($surv1 == "d1a") set col = $cola
  if ($surv1 == "d1b") set col = $colb
  echo " "; echo "#############################################"
  echo Survey: $surv1
  psbasemap -R$LIMI -JX15c -B2g1  -K   > OUTDIFF/$surv1.ci.srf.ps
grdinfo DIFF/$surv1.ci.srf
  grdimage DIFF/$surv1.ci.srf -R$LIMI -CGMT_diff2.cpt -J -O -K  >>
      OUTDIFF/$surv1.ci.srf.ps
#  grdcontour DIFF/$surv1.ci.srf -L-0.2/+0.2 -S3 -C0.05 -A0.1+f0.05 -
    Gd2c -R$LIMI -J -O -K  >> OUTDIFF/$surv1.ci.srf.ps

  if ($surv1 != "d1m") then
    psxy ENU/$surv1.enu    -J -R -B -Sc.1c -W2 -G$col -O -K  >> OUTDIFF/
        $surv1.ci.srf.ps
    psxy ENU/$surv1.enu    -J -R -B -W5/$col -O -K  >> OUTDIFF/$surv1.ci
        .srf.ps
   endif
  psscale -D22c/10c/7c/1c -CGMT_diff2.cpt -L -O     >> OUTDIFF/$surv1.ci
      .srf.ps
 end

echo " "
echo "#############################################"
echo "#############################################"
echo " "
```

```
foreach surv1 ($SURVEYS)
  switch ($surv1)
    case "d1a":
      set col1 = $cola
     breaksw
    case "d1b":
      set col1 = $colb
     breaksw
    endsw
  foreach surv2 ($SURVEYS)
    set menor = ‘echo $surv1 $surv2 | awk ’{if ($1 < $2) print(1); else
        print(0)}’‘
    if ($menor != 0) then
      switch ($surv2)
        case "d1a":
          set col2 = $cola
         breaksw
        case "d1b":
          set col2 = $colb
         breaksw
       endsw
      echo " "
      echo "###############################################"
      echo Surveys: $surv1 $surv2

      foreach t ($TYPES)
        switch ($t)
          case "a":
            set LIM = $LIME
           breaksw
          case "c":
          case "i":
            set LIM = $LIMI
           breaksw
         endsw
        psbasemap -R$LIM -JX15c -B2g1  -K   > OUTDIFF/$surv1.$surv2.$t.
            srf.ps
        grdimage DIFF/$surv1.$surv2.$t.srf -R$LIM -CGMT_diff2.cpt -J -O
            -K  >> OUTDIFF/$surv1.$surv2.$t.srf.ps

        if ($surv1 != "d1m") then
          psxy ENU/$surv1.enu    -J -R -B -Sc.1c -W2 -G$col1 -O -K  >>
              OUTDIFF/$surv1.$surv2.$t.srf.ps
          psxy ENU/$surv1.enu    -J -R -B -W5/$col1 -O -K  >> OUTDIFF/
              $surv1.$surv2.$t.srf.ps
         endif
        if ($surv2 != "d1m") then
```

```
          psxy ENU/$surv2.enu    -J -R -B -Sc.1c -W2 -G$col2 -O -K  >>
              OUTDIFF/$surv1.$surv2.$t.srf.ps
          psxy ENU/$surv2.enu    -J -R -B -W5/$col2 -O -K  >> OUTDIFF/
              $surv1.$surv2.$t.srf.ps
         endif

        if ($t == "a") then
          psxy $TMPDIR/square.$$ -J -R -W5/255/255/255 -O -K   >>
              OUTDIFF/$surv1.$surv2.$t.srf.ps
         endif
        psscale -D22c/10c/7c/1c -CGMT_diff2.cpt -L -O     >> OUTDIFF/
            $surv1.$surv2.$t.srf.ps
######################################################
        psbasemap -R$LIM -JX15c -B2g1  -K   > OUTDIFF/$surv1.$surv2.$t.
            grd.ps
        grdimage DIFF/$surv1.$surv2.$t.grd -R$LIM -CGMT_diff2.cpt -J -O
            -K  >> OUTDIFF/$surv1.$surv2.$t.grd.ps

        if ($surv1 != "d1m") then
          psxy ENU/$surv1.enu    -J -R -B -Sc.1c -W2 -G$col1 -O -K  >>
              OUTDIFF/$surv1.$surv2.$t.grd.ps
          psxy ENU/$surv1.enu    -J -R -B -W5/$col1 -O -K  >> OUTDIFF/
              $surv1.$surv2.$t.grd.ps
         endif
        if ($surv2 != "d1m") then
          psxy ENU/$surv2.enu    -J -R -B -Sc.1c -W2 -G$col2 -O -K  >>
              OUTDIFF/$surv1.$surv2.$t.grd.ps
          psxy ENU/$surv2.enu    -J -R -B -W5/$col2 -O -K  >> OUTDIFF/
              $surv1.$surv2.$t.grd.ps
         endif

        if ($t == "a") then
          psxy $TMPDIR/square.$$ -J -R -W5/255/255/255 -O -K   >>
              OUTDIFF/$surv1.$surv2.$t.grd.ps
         endif
        psscale -D22c/10c/7c/1c -CGMT_diff2.cpt -L -O     >> OUTDIFF/
            $surv1.$surv2.$t.grd.ps
       end
     endif
   end
 end
rm -f $TMPDIR/square.$$
```

B.1.4 surf_all

```
#! /bin/tcsh

set LIME = ‘grep AreaE parameters.lst | awk ’{printf($2)}’‘
set LIMI = ‘grep AreaI parameters.lst | awk ’{printf($2)}’‘

set e_min = ‘grep Xmini parameters.lst | awk ’{printf($2)}’‘
set e_max = ‘grep Xmaxi parameters.lst | awk ’{printf($2)}’‘
set n_min = ‘grep Ymini parameters.lst | awk ’{printf($2)}’‘
set n_max = ‘grep Ymaxi parameters.lst | awk ’{printf($2)}’‘

set cola = ‘grep Color_a parameters.lst | awk ’{printf($2)}’‘
set colb = ‘grep Color_b parameters.lst | awk ’{printf($2)}’‘


echo $e_min $n_min >  $TMPDIR/square.$$
echo $e_min $n_max >> $TMPDIR/square.$$
echo $e_max $n_max >> $TMPDIR/square.$$
echo $e_max $n_min >> $TMPDIR/square.$$
echo $e_min $n_min >> $TMPDIR/square.$$


rm -Rf OUTDIR >& /dev/null
mkdir OUTDIR >& /dev/null

set SURVEYS = "d1a d1b"

foreach surv1 ($SURVEYS)
  if ($surv1 == "d1a") set col = $cola
  if ($surv1 == "d1b") set col = $colb
  echo " "; echo "##############################################"
  echo Survey: $surv1
  psbasemap -R$LIME -JX15c -B4g1 -V -K   > OUTDIR/$surv1.a.srf.ps
  grdimage GRD/$surv1.a.srf -R$LIME -CGMT_abso.cpt -J -O -K -V >> OUTDIR
      /$surv1.a.srf.ps

  psxy ENU/$surv1.enu    -J -R -B -Sc.1c -W2 -G$col -O -K -V >> OUTDIR/
      $surv1.a.srf.ps
  psxy ENU/$surv1.enu    -J -R -B -W5/$col -O -K -V >> OUTDIR/$surv1.a.
      srf.ps

  psxy $TMPDIR/square.$$ -J -R -W5/255/255/255 -O -K -V  >> OUTDIR/
      $surv1.a.srf.ps
  psscale -D22c/10c/7c/1c -CGMT_abso.cpt -L -O    -V >> OUTDIR/$surv1.a.
      srf.ps
 end
```

```
echo " "; echo "##############################################"
echo Survey: d1m
psbasemap -R$LIME -JX15c -B4g1 -V -K   > OUTDIR/d1m.a.srf.ps
grdimage GRD/d1m.a.srf -R$LIME -CGMT_abso.cpt -J -O -K -V >> OUTDIR/d1m.a.srf.ps

psxy ENU/d1a.enu    -J -R -B -Sc.1c -W2 -G$cola -O -K -V >> OUTDIR/d1m.a.srf.ps
psxy ENU/d1a.enu    -J -R -B -W5/$cola -O -K -V >> OUTDIR/d1m.a.srf.ps
psxy ENU/d1b.enu    -J -R -B -Sa.1c -W2 -G$colb -O -K -V >> OUTDIR/d1m.a.srf.ps
psxy ENU/d1b.enu    -J -R -B -W5/$colb -O -K -V >> OUTDIR/d1m.a.srf.ps


psxy $TMPDIR/square.$$ -J -R -W5/255/255/255 -O -K -V  >> OUTDIR/d1m.a.srf.ps
psscale -D22c/10c/7c/1c -CGMT_abso.cpt -L -O    -V >> OUTDIR/d1m.a.srf.ps

rm -f $TMPDIR/square.$$
```

B.1.5 parameters

```
Xmini -43458
Xmaxi -43448
Ymini 152338
Ymaxi 152360
AreaI -43458/-43448/152338/152360
Xmine -43466
Xmaxe -43442
Ymine 152325
Ymaxe 152380
AreaE -43466/-43442/152325/152380
Color_a 255/000/255
Color_b 255/000/000
Color_c 000/255/000
Color_d 000/000/255
```

B.1.6 GMT_abso

```
#cpt file created by: makecpt -Cgray -T57/64/1
#COLOR_MODEL = RGB
#
59   91   91   91   60   91   91   91
60   128  128  128  61   128  128  128
61   164  164  164  62   164  164  164
B  0  0  0
F  255  255  255
N  128  128  128
```