GNSS and Barometric Sensor Fusion for Altimetry Applications

Brian Cavagna Rodrigues
Mestrado em Ciência de Computadores
Departamento de Ciência de Computadores
2018

Orientador
Sérgio Armindo Lopes Crisóstomo, Professor Auxiliar, Faculdade de Ciências

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ______/______/_________

Abstract

GNSS systems allow a receiver to the determine its position (including altitude), with a good accuracy if located in open outdoor environment with clear skies. However the accuracy of the estimated position clearly degrades when the line-of-sight from some of the satellites above the horizon to the receiver is obstructed. The position estimation becomes even impossible is the receiver is located indoors.

This dissertation explores a set of methodologies that aim at improving the accuracy and precision of the of the orthometric height based on information from GNSS systems and barometric sensors, in conjunction with forecast data of the atmospheric pressure and temperature. In particular, we explore the fusion techniques that combine the altitude estimation from a GNSS receiver with the altitude estimation derived from the relation between atmospheric pressure and altitude. As a proof of concept, we designed and implemented an Android application which demonstrates the effective operation of the proposed methods in real scenarios. We perform a comparative study which shows that the methods based on GNSS and barometric sensor fusion clearly outperform, in terms of accuracy and precision, the operation of a standalone GPS receiver. Moreover, the proposed methods are able to accurately determine the altitude both in outdoor and indoor environments.

Resumo

Os sistemas GNSS permitem um receptor determinar a sua posição geográfica (incluindo a altitude), com uma precisão tipicamente boa, quando em um ambiente *outdoor* com boas condições meteorológicas. Contudo, a precisão estimada dessa posição geográfica claramente é degradada quando o alcance do satélite é obstruido, tornando-a até inexistente quando o receptor está em um ambiente *indoor*.

Esta dissertação explora um conjunto de metodologias que têm como objetivo a melhoria da precisão da determinação da altura ortométrica, baseado na informação obtida por sistemas GNSS e sensores barométricos, em conjunto com dados de previsão de pressão atmosférica e temperatura. São exploradas técnicas de fusão que combinam a estimativa de altitude obtida por sistemas GNSS com a altitude estimativa da altitude derivada através da informação barométrica. Como prova de conceito, foi arquitetado e desenvolvido uma aplicação Android em que se demonstra a operação dos métodos propostos em cenários reais. Executamos um estudo comparativo que demonstra que os métodos baseados em fusão dos sensores de GNSS e barométrico se destacam claramente,relativamente a operação do isolada do sensor de GNSS. Além disso, os métodos propostos são capazes de determinar com precisão a altitude tanto em ambientes outdoor como em indoor.

Agradecimentos

A determinação e a força para dar continuidade com um projeto desta dimensão não vieram só de mim. Entre um tempo limitado devido à atividade profissional e muitas pedras que foram surgindo no caminho, muita gente me estendeu a mão para dar uma força e continuar focado.

Primeiramente, um forte agradecimento à pessoa que me ajudou e acompanhou esta trajetória em todas as etapas, o meu orientador Sério Crisóstomo. Obrigado pela paciencia, e pelas verdades que eu precisava ouvir durante esta aventura. Se este projeto foi para a frente, foi devido ao seu auxílio e dedicação prestada.

Em segundo lugar, à minha esposa Isabela, que de tudo fez para me apoiar, e acompanhar. Agiu como psicóloga nos piores momentos e assumiu tarefas rotineiras para me liberar tempo para que isto tudo se concretizasse. Você é uma verdadeira companheira.

Um especial agradecimento ao Instituto de Telecomunicações, pelo suporte dado ao longo deste trabalho.

Aos amigos, que fiz ao longo do mestrado, e que ficarão para toda a vida. Amigos que hoje estão no Brasil, na Angola, e aos que ficaram em Portugal. Um enorme obrigado a todos. Um abraço forte, ao meu amigo Ricardo Leite, que por muitas vezes se disponibilizou para ouvir meus desabafos, e, discutia comigo alguns problemas que foram surgindo, visando dar outro ponto de vista à situação.

Por último, mas jamais menos importante, ao meu pai, simplesmente por ser quem ele é, e de ser a inspiração para o meu futuro.

Dedicatória

Pai,

Estamos distantes em cerca de 8231km, e mesmo assim, pareces estar ao meu lado. O incondicional apoio que me dás suporta todas as minhas decisões. Há 3 anos, decidi partir em busca de uma vida melhor, e de realizar um sonho, que no fundo, eu achava que era somente teu.

Dedico inteiramente a ti, este trabalho, pois ele é fruto de todo o esforço e dedicação que eu pude aplicar em um sonho, que no final das contas, também era meu.

Não há uma única página neste projeto, que eu não tenha feito sem pensar em você, imaginando o quão orgulhoso te sentias de mim naquele momento.

Obrigado por ser quem és, e por fazer de mim uma pessoa cada vez melhor. Estamos sempre juntos!

Contents

List of Tables

List of Figures

Listings

Acronyms

API Application programming interface

ATM Atmosphere

CSV Comma-Separated Values

EGM Earth Gravitational Model

GNSS Global Navigation Satellite Systems

GPS Global Positioning System

GRIB General Regularly-distributed Information in Binary form

GT Ground Truth

GTA Ground Truth Adjusted

IMU Inertial measurement unit

JSON JavaScript Object Notation

MAE Mean Absolute Error

MSL Mean Sea Level

METAR Meteorological Terminal Air Report

NOAA National Oceanic and Atmospheric Administration

ORM Object-relational mapping

RMSE Root Mean Square Error

SI International System of Units

UCAR University Corporation for Atmospheric Research

UI User Interface

WGS World Geodetic System

WMO World Meteorological Organization

Chapter 1

Introduction

Alberto Santos-Dumont is known to be one of pioneers to fly in an airplane propelled by own motors [9]. Among all the problems he had with such adventure, certainly one of them was how to determine the altitude in real-time while in a plane. By then, that would be a headache, given that the technology we breath nowadays was absent.

Today we have several instruments that can measure altitude and give to the user a relatively accurate estimate of the current elevation. The GPS receiver (with all associated infrastructure) is an example of such an instrument, which is capable of determining the geographic location, including the altitude. On the other hand, we can use the barometer which, also through mathematical calculations, allow us to determine the altitude based on pressure differences. They are different devices, with different purposes, that may achieve a common goal.

In recent years we have witnessed an explosion in the creation and manufacture of new gadgets, and today there are more gadgets than people in the world [2], allowing us to perform simple actions like calling the parents or chatting with friends, to complicated issues like editing a text document and uploading it to a cloud, like you would in a regular computer. The technological era we are living now, are opening spaces to deal with increasingly complex tasks. These tasks can be assisted by electronic sensors, which are getting smaller, to the point of being inside a smartphone, such as is the case of a GNSS receiver and a barometer.

With a smartphone equipped with GPS and barometric sensors, it becomes possible to estimate altitude using both sensors, possibly getting better results than using standalone measurements.

This study describes different strategies that make use of both sensors for altitude determination and compares them to the isolated use of the GPS.

1.1 Motivation

Nowadays there are some intelligent gadgets like drones and quadcopters, that may need to know their altitude precisely, to make decisions at the right time.

Each of the sensors in question (GPS and barometer) have their pros and cons. Using sensor fusion techniques to combine the good features that each of them has to offer, it will possibly bring an improved altitude estimates compared to its standalone measurements.

The barometric and the GPS sensors have a behavior that, for purposes of altitude determination, presumably complement each other. The GPS sensor is affected by obstructions, which decreases the quality of its altitude estimations, but the same is not true in the barometric sensor. Also in favor of the barometer, the sensor is always available and responsive, which ultimately does not occur with its counterpart. However, the determination of altitude based on pressure measurements is affected by atmospheric pressure drifts due to changing weather conditions. This is not the case for the GPS sensor. Moreover, for an altitude determination the barometer needs a base reference of pressure and temperature, or else, a base altitude which can be provided by the calculated height from the GPS sensor. The precision of the altitude estimations are distinct, with the barometric sensor being in the forefront due to the smoothness of the altitude variations.

This study aims at improving the altitude estimations, exploring and combining the particularities of each sensor, so that it can be an asset for applications that require highly accurate and precise altitude measurements.

1.2 Objectives

The aim of this study is to explore fusion techniques applied to barometric and GPS sensors for altitude determination. The barometer has the ability to provide very precise pressure measurements that, when converted to altitude, can lead to an estimated orthometric height more precise than the GPS. However, for this to happen, the barometric sensor must be properly calibrated, and a precise reference pressure and altitude (typically the Mean Sea Level (MSL) altitude) from a reference point must be known (which very often is not possible, for reasons that later will be explained). The barometric sensor may also be used to estimate deferential altitudes from a starting point. The altitude of this starting point can be determined using the GPS, so that it can be used as a base reference for the estimation using the barometric sensor.

The study of fusion methodologies considers different scenarios to which the user may be exposed, such as being absent of internet connection, or being in a place full of trees or buildings where reception of GPS signals is deteriorated or even absent. The sensors in question may contain some noise or outliers in their readings, and methods for signal smoothing will also be explored. We also intend to develop a proof of concept in the form of an *Android* application, which will allow to demonstrate the different results obtained with the study carried out.

The detailed contributions could be described as:

- Use weather forecast data to obtain reference atmospheric pressure and temperature at a reference point, for the barometer based altitude estimation;
- Study of fusion techniques of GNSS and barometer sensor information for altitude estimation;
- Develop an *Android* application - *AccuHeight* - as a proof of concept to the studied methods.
- Compare the results proposed fusion methods with the standalone solutions (GNSS and atmospheric pressure based);

1.3 Organization

As seen, the first chapter went shallow in the subject, introducing the problem and the objectives. The following chapters dive deeper starting with some background notions in the second chapter.

It contains the required knowledge to understand the steps taken along the research, and starts addressing a perspective of altitude in a GNSS constellation. There, it is also possible to acquire some knowledge regarding atmospheric pressure and how to extract the altitude, which may or may not rely on the usage of forecast data, also discussed in the chapter. It ends with some related work, in order to understand how other researches treat the subject.

In the third chapter it can be found the explored methodology. Starting from the first approach, to the last one, it is discussed how each was thought, and how them work behind the scenes, exposing the mathematical details and logic.

The fourth chapter describes the system architecture, showing details about the interaction of the system with external services, and sensors. It is also found the *AccuHeight* details. Starting thought implementation details, to the available features the system offers. Shows details about the interface, configurations and warnings.

Found in the fifth chapter, the results and the discussion are presented, aiming to study the results obtained, when applied to different scenarios.

The last section addresses conclusions, starting with a summary of the research and its conclusion. Limitations are there exposed, along with a discussion regarding future work.

Chapter 2

Background and related work

In this chapter we give an overview of the principles to determine the orthometric altitude based on Global Navigation Satellite Systems (GNSS) systems and based on the use of the atmospheric pressure. In the end we present some related work.

2.1 GNSS based altitude determination

A GNSS system is composed by a constellation of satellites [13], that broadcast radio signals that are used by a receiver to determine its position (in a 3D space) and synchronize its clock. The GNSS systems that are currently in operation are described in table 2.1.

Constellation	# of satellites in operation	Origin
GPS	32	United States
BeiDou	32	China
Galileo	24	European Union
GLONASS	26	Russia
NavIC (IRNSS)	8	India
QZSS	4	Japan

Table 2.1: GNSS constellations in operation [8]

2.1.1 How GNSS Works

The operation of a GNSS system to determine the geographic location is given through the measurement of the distance between the satellites and the receptor (derived from the time of flight of the radio signal), in a process designated by trilateration [4]. In figure 2.1, at the right, illustrates this process in a 2D space, where the radius of the circumference centered at each satellite is the estimated distance from it to the receiver. The geographic position of the receptor is calculated by determining the point (or zone) of interception of the circumferences (at least four satellites are needed). At the left of the same figure, the above principle is illustrated for the case of satellites orbiting the earth (3D space), where the location of the receiver lies at the interception of spheres centered at the satellites.

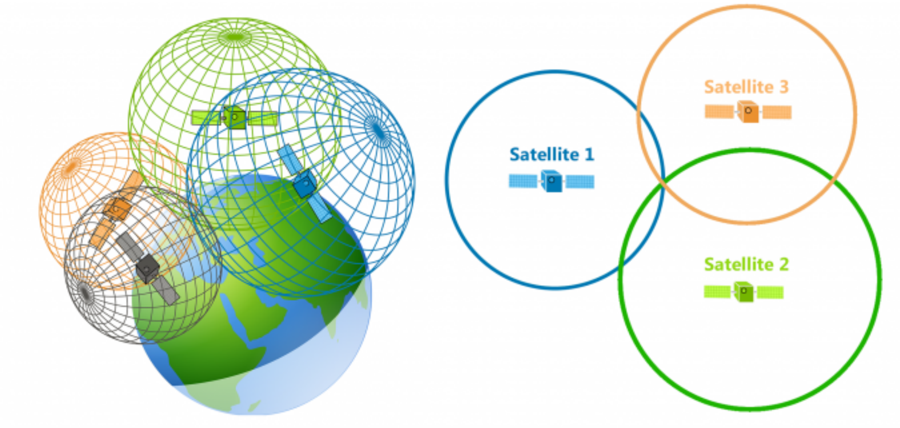


Figure 2.1: Spherical satellites intersection[6]

To measure the distance from a satellite, a receiver uses the expression:

$$Distance = Velocity \times TimeOfFlight$$

where the Velocity is equal to the speed of light, and the TimeOfFlight is the time a signal took propagating from the satellite to it [12]. Therefore, knowing that the radio signal travels at the speed of light [12], and knowing the time the signal departed from the satellite, it is possible to determine the distances between the receiver and the satellites. The estimated position has an accuracy that depends on signal propagation perturbations and on the relative position of the of satellites.

When dealing with vertical measurements, the base reference to calculate the altitude is an ellipsoid. The GNSS reference is the WGS-84 [10], which is a standard that defines the earth as a spheroidal surface - an ellipsoid shape - in order to turn computations simpler than it would referring to the geoid. The geoid is the shape that the ocean surface would take under the influence of the gravity and rotation of Earth alone, if other influences such as winds and tides were absent. This surface is extended through the continents, such that all points on the geoid surface have the same effective potential (the sum of gravitational potential energy and centrifugal potential energy) [14]. The relationship between the geoid and the ellipsoid, is shown in figure 2.2.

When a GNSS receiver determines its altitude $\boldsymbol{H_{GNSS}}$, it corresponds to the vertical distance between its position and the ellipsoid. To derive the orthometric height $\boldsymbol{H_{ORTH}}$ (i.e., the vertical distance $\boldsymbol{H_{GEOID}}$ to the Mean Sea Level (MSL), or geoid), the vertical distance between the ellipsoid and the geoid must be subtracted:

$$\boldsymbol{H_{ORTH} = H_{GNSS} - H_{GEOID}}$$

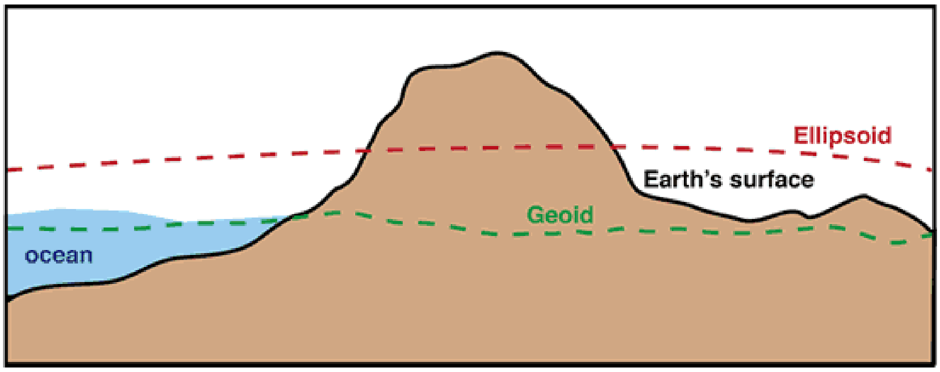


Figure 2.2: Geoid vs Ellipsoid [7]

2.1.2 Error Sources and Accuracy

Some causes that lead to the location estimate to be affected in precision [16] are described below:

1. **Ionospheric and atmospheric delays**
   This effect is caused when the signal is slowed down, causing an effect like the light refracted through water. The calculations may cause an error in the estimations as the velocity of the signal is affected.

2. **Satellite and Receiver Clock Errors**
   This kind of errors, occurs when the clock of the receiver drifts with respect to the clocks of the satellites, causing a bias in the calculated distances to the satellites.

3. **Multipath propagation / Reflections**
   Reflections and multipath propagation, occurs when the signal does not travel directly from the satellite to the receptor, being reflected in surfaces or objects such as buildings or water, leading to imprecise measurements.

4. **Dilution of Precision**
   This error type, is related to the relative position of the orbiting satellites. If the satellites are well spaced between them, the intersection observed has a low level of uncertainty, whereas when they are closer to each other, the level of uncertainty rises, as the intersection area increases.

The accuracy of the location estimation directly depends on factors like the errors described above, and on the number of satellites used in the calculation of the position. In an open space, the horizontal accuracy is typically as good as 4.9m [20], but this value may get worse when the receptor is near buildings that hinders the signal or obstacles that lead to the occurrence of one or more errors such the ones described above.

The vertical accuracy is typically much worse, with an accuracy that can ascend up to 30m [22]. This difference happens due to the redundant information obtained by the satellites in the line of sight that comes in different directions (typically opposed) for the horizontal approach, that is used to check or validate the measurements from the other satellites. That doesn't occur vertically because the the satellites below the earth are not in the line of sight, leading to an scenario where there are not redundant data to be compared and improved.

2.2 Barometric based altitude determination

Pressure is typically defined as the force applied into a determined surface per unit area, therefore, atmospheric pressure is the weight of the air above the surface [15]. Such pressure is lower while in higher points, and higher as the altitude decreases. One of the instruments that measures the atmospheric pressure is called a barometer. The International System of Units (SI) measures the pressure in Pascal (Pa.) [5].

The barometer was invented in 1643, by Evangelista Torricelli [17], using a glass column full of mercury. Torricelli noticed that the liquid created a vacuum at the top of the column, and the mercury exerted more force in the reservoir as the pressure goes up, and in the other hand, lowering as the pressure decreases, causing a vertical drift able to be measured.

The base value to measure the pressure is the mean sea level, where it registers 1 Atmosphere (ATM), which corresponds in standard conditions to 1013,25 hPa.

Since points at different altitudes have different pressures, it is possible to calculate the vertical distance in meters, using the barometric equation [1].

$$\boldsymbol{Hb(P_o, P, T_o) = \frac{T_o}{L}\left(1 - \left(\frac{P}{P_o}\right)^{\left(\frac{R}{g}\frac{L}{M}\right)}\right)}$$

To use the equation determined by the function $\boldsymbol{Hb}$, where the expected result is the estimated altitude $\boldsymbol{Hb}$, it is necessary to apply some constants expressed in table 2.2, being:

Constant	Description	Value	Unit
R	Universal gas constant for air	**8.31432**	$\boldsymbol{N.m/(mol.K)}$
L	Standard temperature lapse rate	**0.0065**	$\boldsymbol{K/m}$
g	Earth Gravity	**9.80665**	$\boldsymbol{m/s^2}$
M	Molar mass of the dry air	**0.0289644**	$\boldsymbol{kg/mol}$

Table 2.2: Barometric Equation Constants [1]

Considering the constants described above, the input parameters to determine the altitude $\boldsymbol{Hb}$ relies on a reference pressure $\boldsymbol{P_o}$, which is the altitude at the reference point (the starting point), whereas **P** registers the pressure at the measuring level (which is either above or below

$\boldsymbol{P_o}$). Finally, the temperature $\boldsymbol{T_o}$, registered 2 meters above $\boldsymbol{P_o}$ level, expressed in Kelvin.

Unless exploring pressure deltas, $\boldsymbol{P_o}$ and $\boldsymbol{T_o}$ may be provided by an external source, such as a prediction file.

2.2.1 Forecast data

One of the biggest challenges to determine altitude based on pressure is to find a reliable reference to be the base pressure $\boldsymbol{P_o}$. National Oceanic and Atmospheric Administration (NOAA) is an U.S. department that studies the skies and oceans, and provides to public forecast data regarding these areas. Such forecast data, is provided in a specific format defined by the World Meteorological Organization (WMO), to be a standard used by meteorological centers [3], which is the General Regularly-distributed Information in Binary form (GRIB) data format.

For the purpose of this study, it is fetched from NOAA several files in this format, containing pressure and temperature predictions at mean sea level for the current localization under a resolution of 0.25 degree grid, relying on a base hour and the occurrences after that base.

Knowing the mean sea level pressure and the temperature of a localization, it is possible to calculate the estimated altitude if a barometer is present and calibrated on a device, between it and mean sea level.

It is worth saying that, this service is available to the Portuguese territory, where this study takes place.

2.3 Related Work

Using electronic components to measure altitude is not a new subject. Several studies provides different alternatives to deal with optimized altitude measurements. This topic addresses a few approaches that uses sensors to estimate the altitude.

[18] presents a study to fuse Inertial measurement unit (IMU) data with GPS information to allow autonomous flight of a quadrotor. It uses the barometer contained in the IMU to estimate the altitude variations and the GPS to determine the absolute position of the system. Other sensors are also used, to determine angles, velocity and orientation. The sensors values are the input to a kalman filter, and the results showed that the fusion of the sensors using the such filter was a success, and the goals were achieved, allowing the quadrotor to have an autonomous flight.

[21] studied the use of multiple barometers and a smartphone to detect the floor position in a building. The central idea is to have a barometric sensor on each floor, that sends the results in real time to a collection server. This causes a floor threshold, that may be determined by the difference between two measured pressures, defining the interval of pressure that floor is. Using a smartphone to measure the current pressure, the user may know the current floor, sending a

request to the collection server and fitting the measured result in the received intervals. The results demonstrates that the method performed well when using a sampling frequency and a time window to remove outliers.

Given the purpose of this study, *Android* applications were also reviewed. Altimeter [19] - developed by EXA tools - is an mobile utility that offers both, online and offline solutions to calculate the altitude. Similar to this study, the application relies on GPS and pressure sensor outputs. In the offline group, GPS is the first method that is used in the solution, and it operates with satellite triangulation. Alone, the offline group offers a solid result if the experiment is taking place in an open space, where the receiving signal tends to have a reduced amount of errors. However, such methodology, by itself, cannot output precise values when dealing with small variations. On the other hand, if including the online resources, the output presents a better result. It uses the GPS to determine location, and uses the location to determine the ground elevation, and the barometer sensor to calculate the altitude through a barometric formula. Although barometer is highly responsive to small variations of pressure, thus leading to a precise altitude measurement, the application doesn't seem to be as responsive as the pressure sensor variations, causing an inflexible output to the user. The developers fused the 3 methodologies using weighted averages.

As another solution to calculate altitude [11], Accurate Altimeter. Like the solution built with this study, the online method uses an external source of data to calculate the altitude with the sensor pressure, relying on a base reference. Instead forecast data from the mean sea level of the current location, this solution uses information from the nearest airport. Unlike AccuHeight, this application shows three different altitudes, the one output by the GPS sensor, the elevation calculated by the current position, and the barometric altitude using the pressure sensor and data available from the airport. All methodologies shows its results, instead fusing them to obtain a most optimized value. This solution has a big gap regarding the second alternative, when using while in a building or an elevator. The elevation output is always regarding the ground level, and for inexperienced users, this information might lead to a bad interpretation of the current situation.

The pressure sensor, when correctly calibrated, along with a consistent source of pressure - like a weather station - can produce a real accurate value of altitude and a sensitive perception to small variations, when applied to a barometric formula. The developers of the previous application, had that into account, and used as a base reference the nearest airport data, contained in a Meteorological Terminal Air Report (METAR) file. Unless dealing with mean sea level, every station or airport are placed in different locations, and relying the reference pressure in only one source may lead to precision errors when moving while the experiment is talking place. Thinking about that, *Altimeter* [1] uses *Delaunay* triangulation as a spatial interpolation to determine the nearest stations to rely the reference pressure on. Such app can determine with an high level of precision the vertical altitude, using only the barometer sensor. In order to evaluate the distance of the nearest weather stations to apply the interpolation, the solution also uses the GPS sensor, but only for location purposes.

AccuHeight comes to use the precision studied in the previous android application *Altimeter*, along with the flexibility to use it in offline mode, like the first solution, where GPS data is used for the purpose. For that, a sensor fusion logic is applied. 2.3 shows a comparison table where it is possible to see that AccuHeight covers all the gaps left by the other solutions.

	Precision to decimeters	Fused Methodologies	Uses external sources
Altimeter (by EXA tools)	No	Yes	No
Accurate Altimeter	No	No	Yes
Altimeter	Yes	No	Yes
AccuHeight	Yes	Yes	Yes

Table 2.3: Android application comparison

Chapter 3

Altitude Methodology

This chapter presents the studied methodologies, and the way they work regardless of any system implementation. Each approach contains all the steps taken to the altitude estimation along with pros and cons observed.

3.1 Method 1

The first approach studied for altitude determination was given through a fixed calibration point at ground level, and starts with three different phases as shown in figure 3.1, and described below:

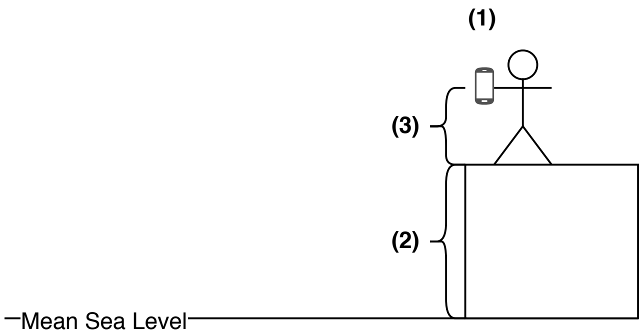


Figure 3.1: Method 1

This method starts (1) by collecting data from location and pressure sensors, simultaneously, during a determined parametrised time, in seconds. As soon as the calibration time runs out, the following task is to extract the average of the different variables, using the gathered samples, resulting in a reference pressure $\boldsymbol{P_0}$, and a reference location, built with the collection of latitudes and longitudes collected from GPS samples. It is also collected, from the forecast data (Gr acronym), the pressure $\boldsymbol{P_{Gr}}$ at the mean sea level, along with the forecast temperature $\boldsymbol{T_{Gr}}$ at the same point, 2 meters above the ground.

The following step (2), is to determinate the altitude at the reference point. To this end, the

Google elevation API was used, which have as input coordinates of latitude and longitude. As the output of the service, we have the the elevation between mean sea level and the calibration point, in meters, to which it is added an offset (3) to compensate the difference between the user equipment position and the ground $\boldsymbol{H_{ORTH}}$.

Given that the calibrated pressure $\boldsymbol{P_0}$, can deteriorate due to climatic events, it should be corrected, as so, to fix it, it should be adjusted given a **Δ**P. Since it was stored the forecast pressure value obtained in the mean sea level as soon as the calibration run out $\boldsymbol{P_{Gr}}$, the offset is calculated given the current pressure obtained by the same data, for the current moment and at the same point, and the counterpart past event.

$$\boldsymbol{\Delta P = Current\ P_{Gr} - Calibration\ P_{Gr}}$$

ΔP thus determines the pressure difference occurring at the mean sea level, between the calibration and current moment. The same fact happens with the temperature, resulting in a $\boldsymbol{\Delta\ T}$.

Therefore, to determine the barometric altitude, it is needed a new sample of pressure P from the measuring equipment.

$$\boldsymbol{H_{BARO} = Hb(P_0, (P_{Gr} + \Delta P), (T_{Gr} + \Delta T))}$$

The actual altitude H, estimated by this approach, is given by the sum of the Mean Sea Level (MSL) elevation of the starting point, and the barometric altitude between the same calibration point and the equipment position:

$$\boldsymbol{H = H_{ORTH} + H_{BARO}}$$

This approach has pros and cons. As an huge pro, it is very accurate, since an external service as Google elevation can lead the reference height to be very precise. In the cons side there is the required internet connection, along with it, the need of an external source of data to calculate the height. If the connection is absent, this method cannot be used.

3.2 Method 2

This method as illustrated in figure 3.2, relies on the pressure difference between mean sea level and equipment pressure (1), on which the barometric formula is applied using the read pressures.

Assuming that a forecast file in a General Regularly-distributed Information in Binary form (GRIB) format, parsed to the table 3.1, can reveal the predictions to a variable regarding a

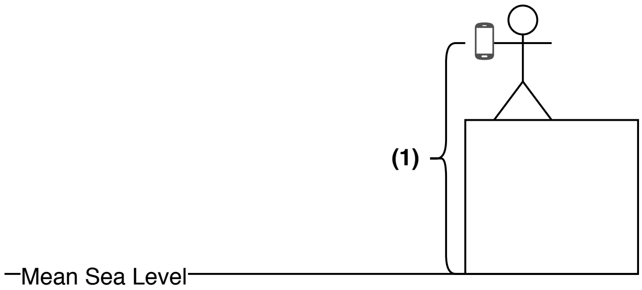


Figure 3.2: Method 2

determined pair of hours, an interpolation is required in order to be able to extract such value everytime there is an pressure update from the sensor.

Variables	**08:00**	**09:00**
Pressure (hPa)	1014	1018
Temperature (K)	287	293

Table 3.1: Decoded GRIB variables example

Therefore, to calculate the altitude estimation between mean sea level and the equipment, for every pressure $\boldsymbol{P}$ update the equipment receives, the barometric formula is applied using as a pressure reference, the interpolated values, pressure at mean sea level, $\boldsymbol{P_{Gr}}$ and temperature in the same point, at 2 meters above ground, $\boldsymbol{T_{Gr}}$ for the same second the pressure event occurred:

$$\boldsymbol{H_{BARO} = Hb(P, P_{Gr}, T_{Gr})}$$

In this specific case, $\boldsymbol{H_{BARO}}$ is the real altitude H.

Unlike the last approach, this method does not need internet, assuming the system was booted when a connection was present, in order to download the latest forecast files only. If the shape is big enough, then the system will flow without any problems even without internet connection, which is considered a pro. As a con, the pressure sensor must be very well calibrated, or else the values calculated may lead to inaccurate measurements.

3.3 Methods 3 & 4

In this section two methods are presented at same time, since both work in the same way, except that, method 3 does not consider the deterioration caused by the climatic events.

Similarly to the first approach described, this method also needs a calibration period, in a fixed point **G** at the ground. In the same way, it collects data from the pressure sensor and the location sensor, however, this approach does not use the GPS to fetch coordinates, but the vertical height instead.

When the calibration timeout is reached, an average of the values is calculated, originating the estimated GPS height $\boldsymbol{H_{GNSS}}$, the estimated pressure $\boldsymbol{P_0}$. Along with these averages, the geoid height $\boldsymbol{H_{GEOID}}$ is also stored, fetched from an external service.

The orthometric height $\boldsymbol{H_{ORTH}}$ at point **G** is obtained through the following calculation, which is the result of subtracting the average GPS altitude with the stored geoid value to the same position:

$$\boldsymbol{H_{ORTH} = H_{GNSS} - H_{GEOID}}$$

At this point, the logic of both methods diverge, where exclusively to method 4 the **Δ**P is calculated, in order to consider the deterioration caused in $\boldsymbol{P_0}$ due to the weather occurrences. Like the first approach, the **Δ**P is the difference between the current pressure at mean sea level, and the pressure at the same point obtained in the calibration timeout. This calculation relies on the data provided by the forecast file. The temperature difference is also calculated on $\boldsymbol{\Delta T}$.

To evaluate the altitude between the calibration point **G** and the current equipment position, the barometric formula is applied, using **G** as a reference pressure, and a pressure **P** obtained by the smartphone

For method 3:

$$\boldsymbol{H_{BARO} = Hb(P_0, P, T_{Gr})}$$

For method 4:

$$\boldsymbol{H_{BARO} = Hb((P_0 + \Delta P), P, (T_{Gr} + \Delta T))}$$

The temperature input to the equations above described, is the real time interpolated temperature at mean sea level, 2 meters above ground.

Therefore, to estimate the current altitude

$$\boldsymbol{H = H_{ORTH} + H_{BARO}}$$

As an advantage, this approach does not require an internet connection, as long as the forecast files are present in the system, and the shape is big enough to cover the area of the experiment, however, the previous method already achieved that. The difference relies on the barometer calibration, which is not a must to run this method, since we are only interested in the pressure deltas that occur between **G** and the equipment. On the other side, the GPS seems to be the bottleneck, if the experiment takes place in a location where the signal arrives poorly or full with errors, hence, the determination of the orthometric altitude may have a drift.

3.4 Fusion Based Approaches

The fusion methodologies use smoothing techniques of the sample given by a sensor, and for a deeper understanding of the content that will be presented below, it is necessary to address Kalman Filters.

3.4.1 Kalman Filters

Kalman filter is a mathematical process to estimate real values given samples $\boldsymbol{MEA}$ carrying noise and outliers. The filters' prediction is based in a weighted average, between the measurement and prediction. It starts by defining a kalman gain $\boldsymbol{K_G}$, which is an estimator to place the weight in which the filter should rely, then it calculates an estimation $\boldsymbol{EST_t}$ for the actual observation, based on the previous state $\boldsymbol{EST_{t-1}}$, and finally, calculates the error in the estimate $\boldsymbol{E_{EST}}$, that is used in sequence evaluations. Such filters, take into account the apparatus error $\boldsymbol{E_{MEA}}$, namely, the average error of the measurements read by the equipment. The way to evaluate those values are described below

1. $$\boldsymbol{K_G = \frac{E_{EST}}{E_{EST} + E_{MEA}}}$$

2. $$\boldsymbol{EST_t = EST_{t-1} + K_G[MEA - E_{EST_{t-1}}]}$$

3. $$\boldsymbol{E_{EST_t} = [1 - K_G](E_{EST_{t-1}})}$$

The current work uses such filters in order to smooth the GPS signal, ignoring the noise and outliers caused by the several obstacles between the GPS signal and the receptor.

3.4.2 Method 5 variants

As seen, one of the biggest disadvantages of previous method is the confidence level of the orthometric altitude, measured by the GPS during a finite amount of time. Satellites may not output the most accurate results depending on signal quality that the measuring equipment is receiving, nevertheless, if an determined sample - in these variants the very first - is constantly refined along an infinite period, we may reach the real value at that point. This method contains two different approaches, that intend to converge to similar results, the first one that uses kalman filters, and the second one that uses an simple average to determine the real altitude at the initial point.

Although both approaches behave independently there are a few common steps regarding the data collection.

For each location update obtained in the equipment, an orthometric altitude should be estimated, using the location and the geoid value the system holds for the current position.

$$\boldsymbol{H_{ORTH} = H_{GNSS} - H_{GEOID}}$$

As soon as this calculation takes place, the pressure $\boldsymbol{P}$ at that moment is fetched, along with the interpolated temperature $\boldsymbol{T_{Gr}}$ contained in the GRIB files.

Such values are then wrapped in an *observation*, **t** that holds the three variables above described, resulting in

$$\boldsymbol{t_i(H_{ORTH}, P, T_{Gr})}$$

Each observation created is then processed according the selected approach.

3.4.2.1 Method 5.0

There is a chain of execution regarding the evaluation of this method, shown in the figure 4.9, which contains few steps to estimate the altitude.

Calculating the barometric altitude is the first step, as so, when an observation arrives to be processed, the first step is calculate the altitude between the mean sea level and the equipment position, based on the interpolated pressure observed in the forecast file, $\boldsymbol{P_0}$, and the data contained in the observation:

$$\boldsymbol{H_{BARO} = Hb(P_0, P, T_{Gr})}$$

The output of this calculation already defines the estimated altitude between the mean sea level and the equipment position, however, the goal of this approach is to rely the reference altitude on GPS, ignoring possible miss-calibrations with the pressure sensor, and leading the altitude measurement to a most precise value. Thus, one of the most important values to calculate is the fixed GPS altitude, created to nullify the altitude changes occurred between the measurements, with the goal to have a dataset of possible altitudes at the first point of the experiment $\boldsymbol{T_0}$, along all observations. Submitting this dataset to kalman filters, it is possible to filter noise and outliers of possible GPS measures that may not be precise, providing also a smooth variation of this reference which will be used as the base altitude.

As so, the next operation in the chain is to calculate the fixed GPS altitude, $\boldsymbol{GPS_{Fixed}}$, which is defined by the orthometric measured in the observation, subtracted the delta altitude $\boldsymbol{\Delta}$b estimated with the pressure, between observations, being:

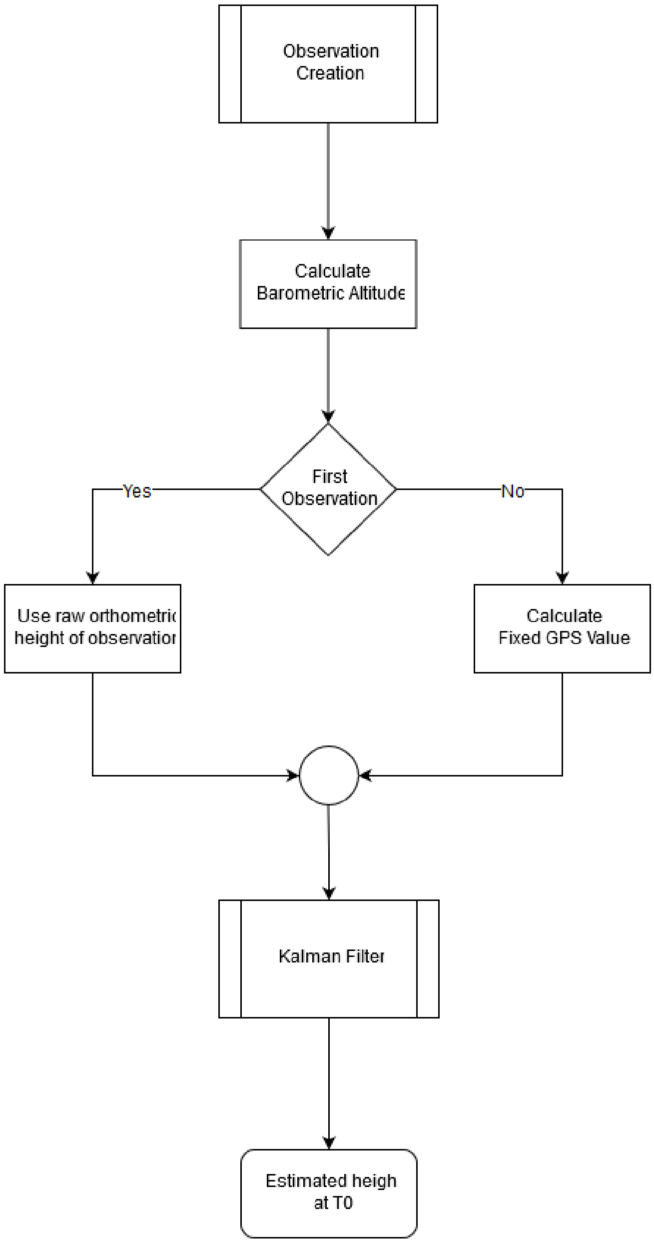


Figure 3.3: Processing workflow of method 5.0

$$\Delta b = H_{Barometric} - H_{Barometric_0}$$

$$GPS_{Fixed} = H_{ORTH} - \Delta b$$

With such evaluations set, the third step in the chain is to submit the result to kalman filters. As shown in figure 4.9, there is a slightly difference between the first observation processing and the subsequent ones. This occurs because the first observation has no past events estimated, nor weights to address the estimations, therefore, the calculations are initialized with default values. Instead submitting the GPS_{Fixed} in the first case, it is submitted the raw orthometric observation, and once there is not an error in the estimate it is used to calculate the gain, the vertical average GPS error $\boldsymbol{GPS_{err}}$, which is a parameterised value defined by the user, as such:

$$\boldsymbol{K_G} = \begin{cases} \dfrac{\boldsymbol{GPS_{err}}}{\boldsymbol{GPS_{err} + GPS_{err}}} & \text{when } \boldsymbol{t} \text{ is 0,} \\ \\ \dfrac{\boldsymbol{E_{EST_{t-1}}}}{\boldsymbol{E_{EST_{t-1}} + GPS_{err}}} & \text{otherwise} \end{cases}$$

$$\boldsymbol{EST_t} = \begin{cases} \boldsymbol{GPS_{Fixed} + K_G[GPS_{Fixed} - GPS_{Fixed}]} & \text{when } \boldsymbol{t} \text{ is 0,} \\ \\ \boldsymbol{EST_{t-1} + K_G[GPS_{Fixed} - E_{EST_{t-1}}]} & \text{otherwise} \end{cases}$$

$$\boldsymbol{E_{EST_t}} = \begin{cases} \boldsymbol{[1 - K_G]GPS_{err}} & \text{when } \boldsymbol{t} \text{ is 0,} \\ \\ \boldsymbol{[1 - K_G](E_{EST_{t-1}})} & \text{otherwise} \end{cases}$$

At soon as these calculations take place, is its created a kalman filter object which holds the estimations, and a relation with the observation object through an id, and the processing of the observation $\boldsymbol{t_i}$ is finished. At this point, if t>0, there is enough data to determine the real altitude. Since the output of kalman filter at this experiment is expected to be, along the observations, the most close to the real GPS altitude at $\boldsymbol{t_0}$, and thus used as a base reference, for each pressure update event occurring, may be calculated the delta altitude $\boldsymbol{\Delta h}$ between the current altitude estimation and the altitude at that point, as such:

$$\boldsymbol{\Delta h = Hb(P_0, P, T_{Gr}) - H_{Barometric_0}},$$

Finally, estimating the real altitude being evaluated as the delta calculated above and the last kalman filter output,

$$\boldsymbol{H = EST_t + \Delta h}$$

The pros and cons achieved with this approach are the same of method 4, with the addiction of the GPS values, that tends to be more accurate when using a time-series to constantly evaluate a height of a determinate point, and applying a filter to cancel outliers. As one con observed, is that kalman filter tends to converge to the real values with some delay in cases of a sudden difference in the altitude.

3.4.2.2 Method 5.1

As a counter-proposal to the last shown variant, this approach eliminates the usage of the kalman filter, with the goal to converge the real values faster, and follow the variations almost in real time. As a replacement, in order to estimate the real altitude value at $\boldsymbol{t_0}$, a simple average of the samples is estimated, regarding the $\boldsymbol{GPS_{Fixed}}$ value. Behind the scenes, the mechanics implemented in this method is very similar comparing to the last one, changing only the processing chain in few aspects.

As soon as an observation is ready to be processed, the chain calculates the barometric altitude and the $\boldsymbol{GPS_{Fixed}}$ using the same approach. The difference is that, instead of submitting this output to kalman filters, $\overline{\boldsymbol{H_{t0}}}$ holds the average of the estimated real altitude value at $\boldsymbol{t_0}$.

$$\overline{\boldsymbol{H_{t0}}} = \frac{\sum_{t=0}^{n} \boldsymbol{GPS_{Fixed_t}}}{\boldsymbol{n}}$$

Then, the real altitude value may be calculated with such average as the reference altitude. Like last approach, to determine the final height, it is needed a $\boldsymbol{\Delta h}$ between current equipment height and the reference altitude height compared to mean sea level shown in 3.4.

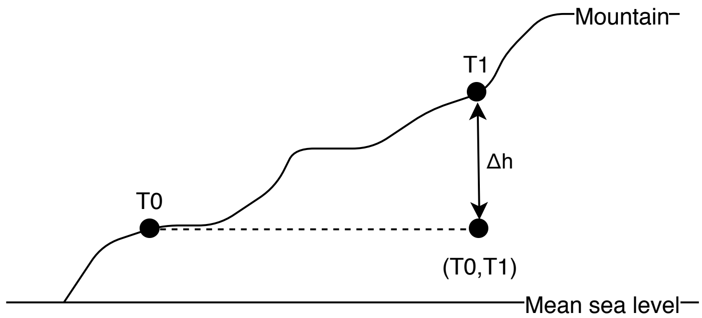


Figure 3.4: $\boldsymbol{\Delta h}$ representation

Using both values, the real altitude is obtained for each pressure event occurring in the system, evaluated by:

$$\boldsymbol{H} = \overline{\boldsymbol{H_{t0}}} + \boldsymbol{\Delta h}$$

This approach contains the same pros and cons observed in the same variation using kalman filters, but, tends to converge to real values very quickly. In the other hand, noise and outliers may be considered in the average, leading to a small and undesired drift.

3.4.3 Method 6 variants

Relying exclusively on forecasting data to determine the altitude between mean sea level and the equipment level may not be the best approach when there is limited, or even no access to internet. The purpose of the following variants is to continue tracking the altitude without the reference contained in forecast data. To achieve that, pressure sensor takes again the main role, but, on the other hand, the deterioration of the samples is yet an issue due to weather events. To deal with such decay these methods implement a window of processing, exposed in figure 3.5, where this window should be big enough to estimate the real altitude with quality, but short enough in order to stanch the pressure samples deterioration. The processing window size **PWS** is defined by the user, and the system then submits to process only the samples within that range:

$$\boldsymbol{ProcessingWindow} = [\mathbf{max(0,}\ \boldsymbol{sample_{n-PWS}})\boldsymbol{...}\ \boldsymbol{sample_n}]$$

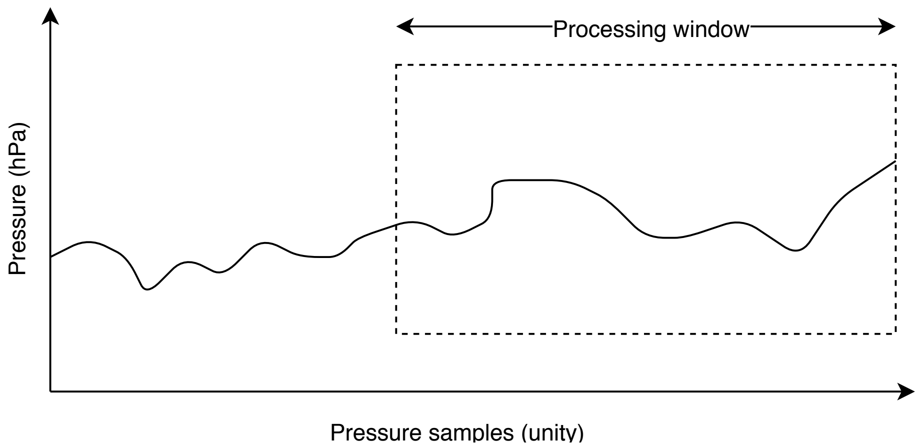


Figure 3.5: Variants processing window

The whole mechanics regarding the following variants starts in a similarly way than the $\mathbf{5}^{th}$ method. It is also determined the orthometric height using each value the localization sensor that arrives, subtracted the geoid height. An observation is also created, but the big difference is that instead using the forecast interpolated temperature, it is fetched the temperature **T** at the location, through a temperature API service. At this point, if no internet is detected, a standard value of temperature may be used.

The entry point of processing of this variation is to define the reference observation 3.7 (2), in other words the observation that in a determinate finite time series, is the first, on which the calculations should rely. Unlike the method 5.0, where the reference height to refine is fixed at the first measurement (1), this approach contains a variable reference, to avoid that a previous pressure samples do not get deteriorate over the time.

The reference observation is then defined using the **PWS** formula described above. After defining it the observation may be processed according the selected approach.

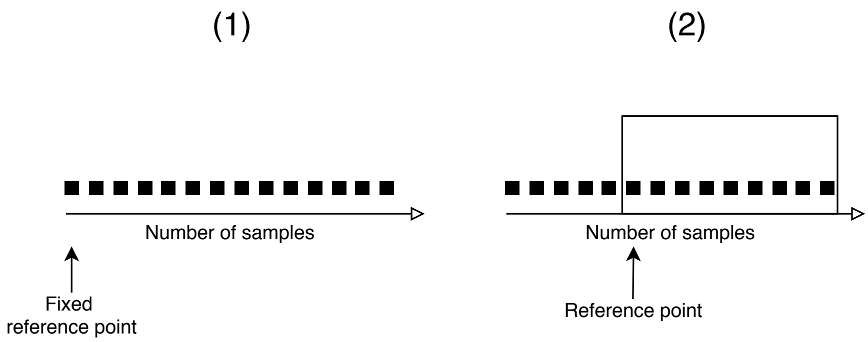


Figure 3.6: Method 5 vs Method 6

3.4.3.1 Method 6.0

This variant uses kalman filters, and likewise in previous method, the chain of execution is the same. It also have the same behaviour until the size of window configured by the user is reached.

The difference starts when the observations along the time series exceeds the window size, causing a re-evaluation of the kalman filter estimations, for every sample within the window size, which starts with the reference observation.

This action leads to a disregard of the pressure data collected that stands before the processing window, that already may be affected with the climatic events.

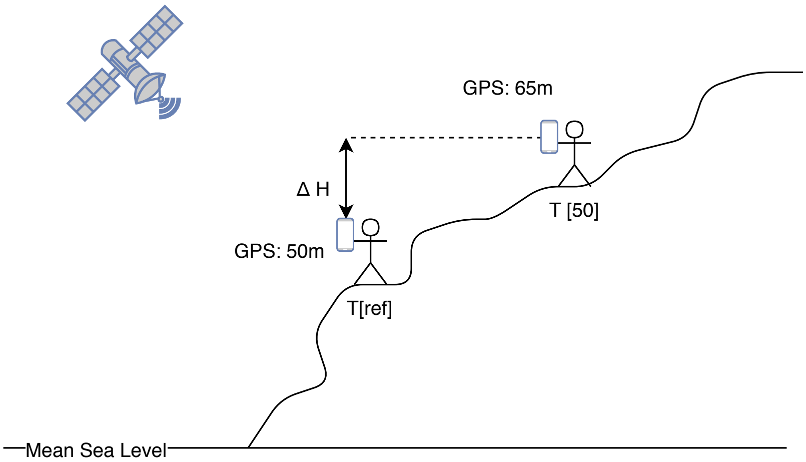


Figure 3.7: Method 6

Similarly to the previous approach, a delta for each observation against the reference observation is calculated using the barometric altitude between the reference observation $\boldsymbol{P_0}$ and the current pressure in the system

$$\boldsymbol{\Delta h = Hb(P_0, P, T)}$$

The previous calculation estimates the altitude difference between two observations, meaning that, it is possible to reveal the altitude, $\boldsymbol{GPS_{Fixed}}$, in the reference point.

Each of the $\boldsymbol{GPS_{Fixed}}$ value calculated within the observation series, is then submitted to a kalman filter calculation, to optimize the value.

The estimation of the kalman filter given a processing window, is the optimized altitude in the reference point.

The real altitude estimation is then, given by instant barometric altitude between the pressure sensor occurrence and the reference point $\boldsymbol{\Delta H}$.

$$\boldsymbol{H = \Delta H + KalmanEstimation}$$

The benefits of using this approach are huge, since, apart from the temperature gathered from the internet, no connection at all is needed to calculated this variant. Also, no forecasting or external sources are required to calculate the altitude with precision. The only con observed, is the same as method 5.0 where kalman filter tends to have a slow response when facing sudden changes regarding the GPS altitude.

3.4.3.2 Method 6.1

The mechanics regarding this variant are very similar than the described above, but instead using kalman filters, it uses the window average to determine the altitude at the reference point.

The processing of the observations are the same in method 5.1, while the window size is not exceeded. After that happens, likewise the previous approach, the delta of the altitude between every observation within the window and its reference, must be calculated.

$$\boldsymbol{\Delta h = Hb(P_0, P, T)}$$

After that calculation, each observation is now able to define the estimated altitude at reference point, using an average of the value that observation has in the GPS measurement, subtracted the calculated delta

$$\boldsymbol{\overline{H_{tref}} = \frac{\sum_{t=0}^{n} GPS_{Measurement}[n] - \Delta h[n]}{n}}$$

Also, an calculation of the barometric altitude between the equipment and the reference point, $\boldsymbol{\Delta h}$.

$$\boldsymbol{\Delta h = Hb(P_0, P, T)}$$

The real altitude estimation is the sum of the estimated altitude in the reference point, plus the occurring delta.

$$\boldsymbol{H = \Delta H + \overline{H_{tref}}}$$

Like the last approach, the same pros are present, and covers the sudden changes that a GPS may have, and yet converge to real values fast. As a con, this variant may include into its calculation some undesired noise related to the GPS output.

To summarize the explained methodologies, table 3.2 shows what each methods needs to operate.

Table 3.2: Methods Summary

		On boot				Operating		
	Sensor Calibration	**External Data**	**GPS**	**Barometer**	**Forecast Data**	**GPS**	**Barometer**	**Forecast Data**
GPS ⋆		Geoid Service				Altitude		
Method 1 ⊗		Elevation API Geoid Service	Lat, Lon	x	x	Lat, Lon	x	x
Method 2 ⊗	x		Lat, Lon				x	x
Method 3 ⊗		Geoid Service	Lat, Lon; Altitude	x		Altitude	x	
Method 4 ⊗		Geoid Service	Lat, Lon; Altitude	x		Altitude	x	x
Method 5 ▷		Geoid Service	Lat, Lon			Altitude	x	x
Method 5.1 ▷		Geoid Service	Lat, Lon			Altitude	x	x
Method 6 ▷		Geoid Service				Altitude	x	
Method 6.1 ▷		Geoid Service				Altitude	x	

⋆ Standalone sensor altitude measurement.

⊗ Non-Fused sensor altitude measurement.

▷ Fused sensor altitude measurement.

Chapter 4

Application Design and Implementation

This chapter addresses the system architecture and development. It starts by giving an overview of the Android framework that was used for the application design. Then it describes how the system operates with its external services, with its sensors, and explains further about the *Android* application. *AccuHeight*, has the goal to wrap up the methodologies studied in the last chapter, handling the sensors, API calls and the related calculations. Such application was designed to work on android devices that holds any version above *Android Oreo 8.0*, pressure sensor and localization sensors. This section presents the application and its features.

4.1 Android Framework

In order to understand the implementation section, it is important to remember some concepts regarding the Android environment.

4.1.1 Activity

Activities are unique modules that holds an user interface and a programmatic logic that allows the user interaction with the system.

Such modules are self-contained, and follows the Android framework patterns, known as a lifecycle described in the figure 4.1.

The lifecycle defines a sequence of steps that are triggered in an order defined by the *Android* framework, on which the developer interacts with the system and the user. It starts with the onCreate() method, where the user interface should be instantiated and inflated to the screen, followed by the onStart() method, that is triggered as soon as the view starts to be shown to the user. The other actions are activated depending on the user actions.

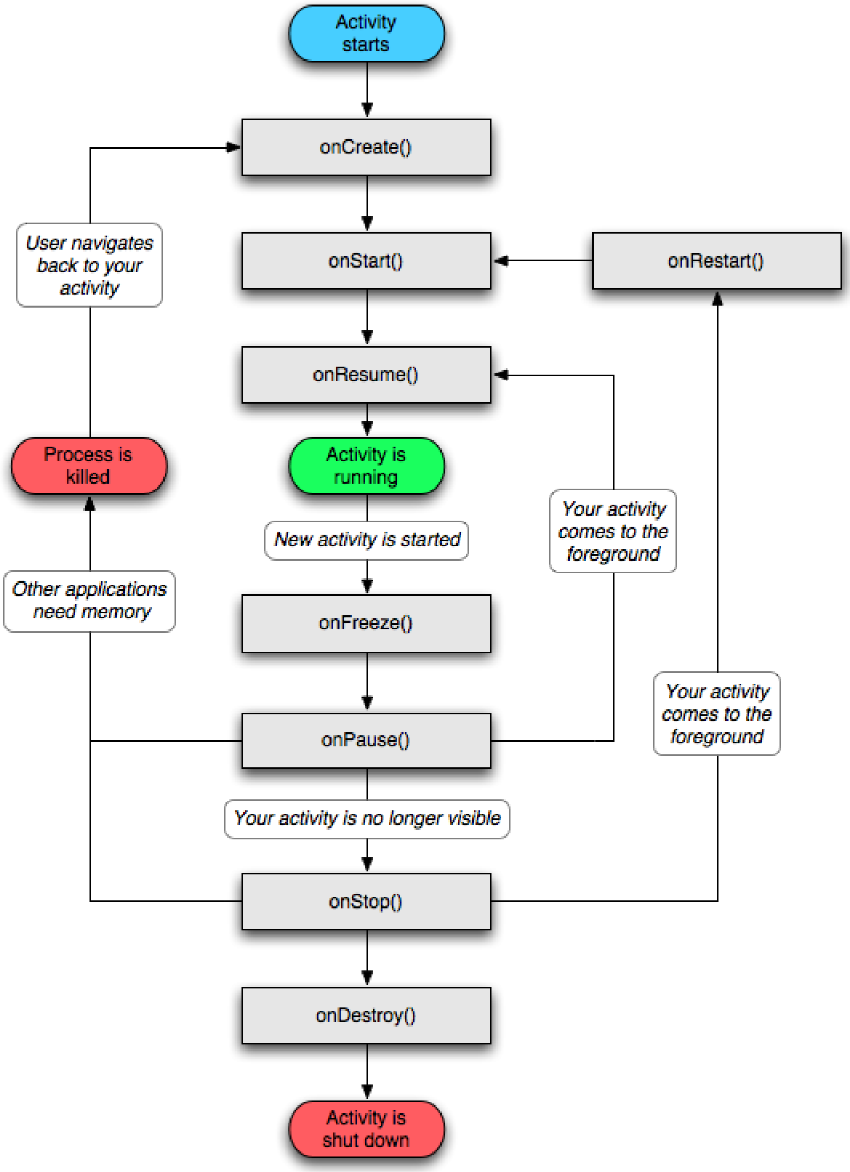


Figure 4.1: Android Lifecycle

4.1.2 Fragment

Fragments represents a reusable module that are contained within an activity. Like activities, fragments also have an interface associated with the code to make possible user interaction against the system. This approach is typically used to combine multiple interfaces in only one screen.

4.2 General behaviour

This software proposal relies on the pressure and location sensors along with forecast data gathered from National Oceanic and Atmospheric Administration (NOAA) repository. The suggested methodologies fuses these information depending on its purpose. In order to accomplish such fusion, figure 4.2 illustrates the system operation.

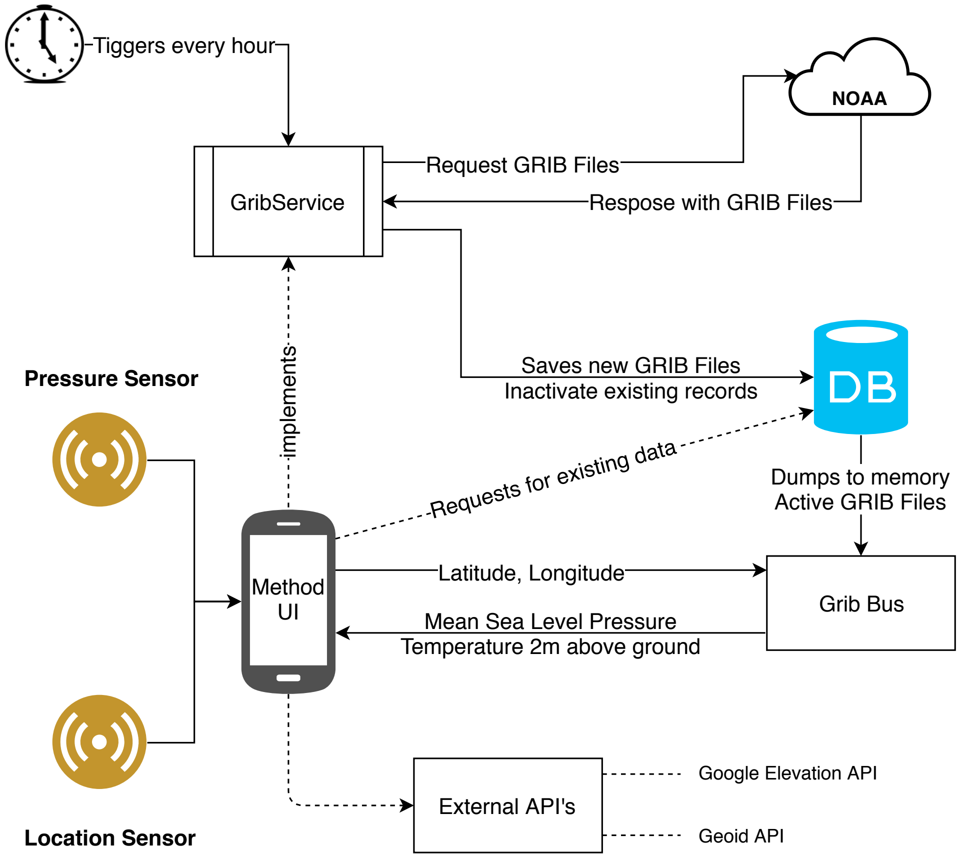


Figure 4.2: System Architecture

The system operation comprehends two different paths that works together, to bring a fluid experience to the user. In a cold start, the system triggers the GribService, that requests the source for new forecast data, given the location outputted by the phone sensor at that moment. As soon as the service responses with the requested files, these are saved in the database, to avoid new requests to the source while the files are not yet outdated. Grib Bus, is a static class, that acts as soon as new data reaches in the database, and loads into memory the files, to be possible to quickly request for its data given a latitude and longitude.

Along with this, we also implemented a timer, that wakes the GribService up, in order to seek for new and fresh forecasts data, and so, force the calculations to rely on a more accurate reference data.

In the other hand, in a hot or warm start, the system also triggers the GribService to do the same job cited above, but the probability of the database to have active files is high, and so, the system requests the database to dump into GribBus its forecast files content.

The system is then ready to use any of its implemented methods, where each of it calculates the altitude in different ways.

4.3 System Structure

Considering the presented system was built using design patterns with the goal to maximize the code reuse, *AccuHeight* was split in only one activity and fragments that represents the methodologies and the system options.

Entrypoint - the single system activity - was build to take care of the core responsibilities of the application. In it, it was created an side menu that is used throughout the entire system, and a fragment holder that is used to sit the fragments in the screen, that is represented in figure 4.3 in the view of a user click.

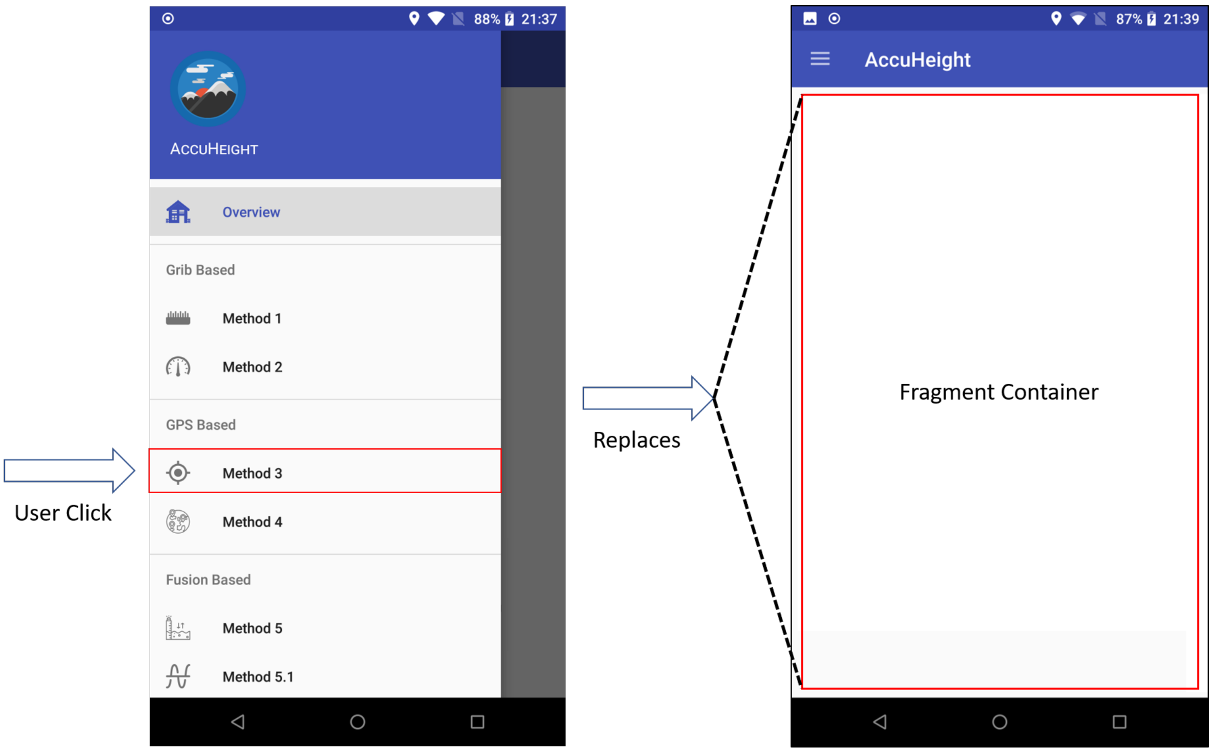


Figure 4.3: Entrypoint menu and fragment holder

The activity has as its goal, also the following responsibilities, and described below.

1. Handle user permissions
2. Handle and log uncaught exceptions thrown by the fragments
3. Manage system menu and related events

4. Manage and distribute sensor events

5. Finish and destroy system resources used by the fragments

Starting from the top, the system requires that the user allows the application to use the resources. The first permission prompt for the user is related to the access of photos, media and files of the device, and it is needed due to the database that this system uses to store relevant data. The second one, relies on the usage of the location sensor. These system must acquire such permissions as shown in figure 4.4, in order to guarantee the application works as expected, and will persist to have such permissions even if the user denies it acquisition.

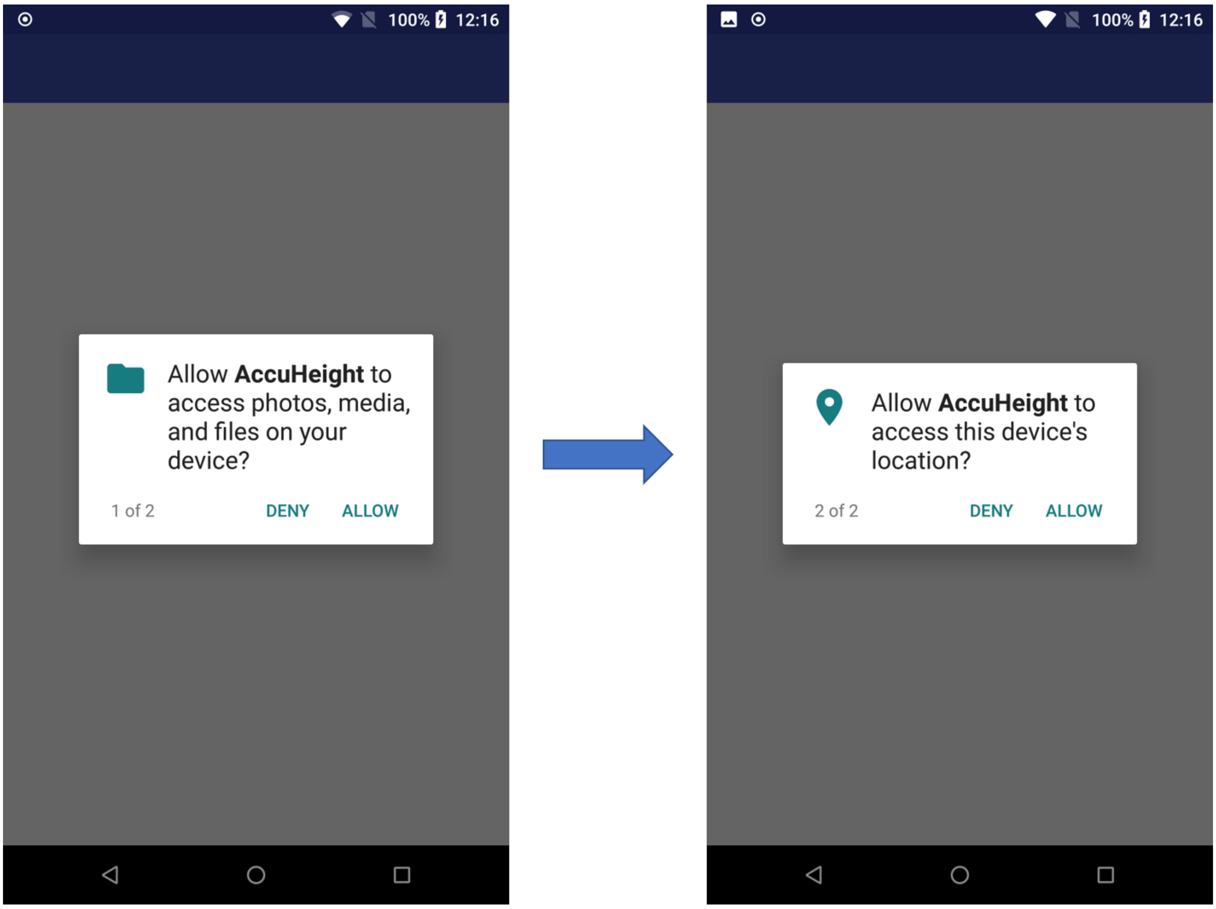


Figure 4.4: System Permissions

The uncaught errors feature is also implemented in the activity, as a core functionality. Not all implementations are wrapped inside a try/catch block, and in those cases, if an error occurs in the system - without such feature - it would not be possible to track the cause of the exception. With this feature, as soon as an unexpected failure occurs, the stack trace is saved in the database for posterior analysis.

The side menu was built inside the activity to be used along all the system, without the need of replicating the code on every user interface. In it it is possible to change the fragment within the fragment container and export logs and samples to email or cloud.

Sensors handling are also a core functionality, implemented in the activity. EntryPoint instantiates the sensors, and requests for its updates. The key goal is to send the result of these updates to a broker, to be shared along with the fragments.

Before killing the application process, the activity is also responsible to terminate the fragments events, services updates and requests, in order to avoid memory leaks on the device.

Some of these events and actions, are aided by third party frameworks, that is presented in the next session.

4.4 External services and Frameworks

As the focus of this application is to fuse information using sensors and forecast data, this application is supported by usage of external services and frameworks, to high-level acquire data and help the fusion processing of the developed methodologies.

4.4.1 GRIB Java Decoder

As seen before, this solution makes use of forecast data at mean sea level, and such information is contained in General Regularly-distributed Information in Binary form (GRIB) files distributed by NOAA. As a complex binary file, retrieving its information is not trivial, and University Corporation for Atmospheric Research (UCAR) managed to build this framework, that consists of a set of utilities that high-level handles the file contents.

With the raw GRIB file downloaded in the mobile, the first step to use such framework is to create a GridDataSet object, using the following line of code represented in listing 4.1:

```
GridDataset gridDataset = GridDataset.open(gribFilePath);
```

Listing 4.1: Convert binary GRIB file to GridDataset

The created object provides a set of utilities to read the binary file, and for this study, the interest relies on retrieving the pressure at mean sea level, and the temperature at the same point, 2 meters above ground. Since there may have multiple values to extract, an helper method was created, aiming to reuse code. **getVariableValue**, receives as parameters the GridDataset, coordinates and the variable name, and is responsible to return the value for that variable at the requested point, shown in 4.2.

With this, the system is capable of parsing the binary file into a domain object, that holds the forecast values, and uses it in any necessary situation.

```
private static Double getVariableValue(GridDataset dataset, Double latitude,
    Double longitude, String variableName) {
      GridDatatype grid = dataset.findGridDatatype(variable);
      GridCoordSystem gcs = grid.getCoordinateSystem();
      int[] xy = gcs.findXYindexFromLatLon(latitude, longitude, null);
      return grid.readDataSlice(0, 0, xy[1], xy[0]).getDouble(0);
  }
```

Listing 4.2: Private method to retrieve the variable value

4.4.2 Google Elevation API

Google Elevation API is a service that provides the elevation for a location on the earth surface. Such elevation values are related to the local mean sea level, and expressed in meters.

This service works through an HTTP interface, using the geographic coordinates, along with a developer key, to construct an HTTP *GET* call. The result of this request is an JavaScript Object Notation (JSON). Linsting 4.3 shows an example of the output related to the request.

```
{
   "results" : [
      {
         "elevation" : 68.65777587890625,
         "location" : {
            "lat" : 41.15181389,
            "lng" : -8.636144440000001
         },
         "resolution" : 9.543951988220215
      }
   ],
   "status" : "OK"
}
```

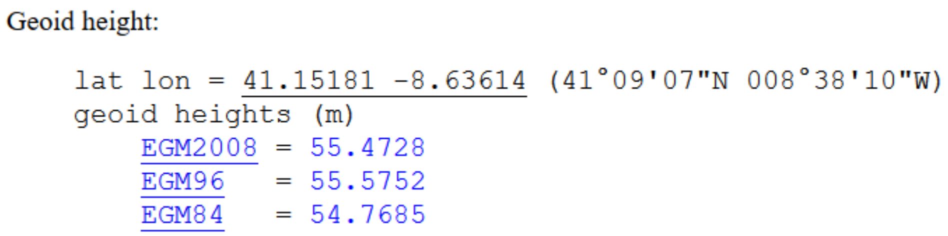
Listing 4.3: Google Elevation API response

Within *AccuHeight*, this API works as a background asynchronous service, where it is used passing required arguments, the latitude and longitude. The URL is then created, and the request is made, resulting in an output similar to listing 4.3. The JSON response is then parsed, and the service returns the *elevation* node value.

4.4.3 Geoid Height Calculator API

Similarly to the previous API, this service is responsible to estimate the geoid height at a determined geographic point. The request is constructed as a URL string, using coordinates to the calculation.

Since the response is an HTML page 4.5, a parser was used to retrieve the geoid value, from the Earth Gravitational Model (EGM) 2008 model.


Geoid height:

```
lat lon = 41.15181 -8.63614 (41°09'07"N 008°38'10"W)
geoid heights (m)
    EGM2008 = 55.4728
    EGM96   = 55.5752
    EGM84   = 54.7685
```

Figure 4.5: Geoid Service output as an HTML page

Throughout the system, this service is used along with GPS measurements, in order to estimate the orthometric height.

4.4.4 EventBus Framework

Aiming to send to every fragment the changes occurring in the sensors, *EventBus* was used to be a data bus between the Publishers - pressure and location sensors - and the subscribers - the fragments - that implements a listener to a specific event. The framework operation is described in figure 4.6.

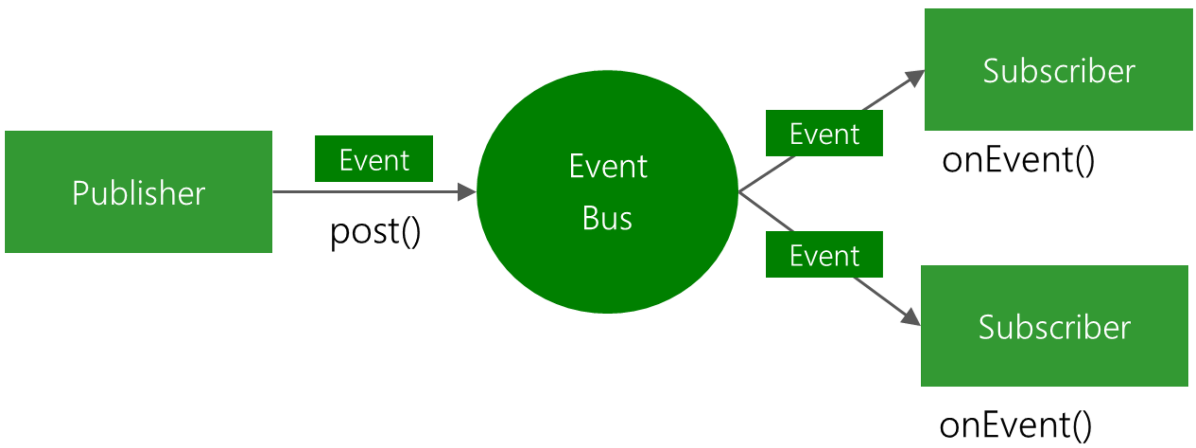


Figure 4.6: Eventbus

AccuHeight contains two main publishers, the pressure publisher, and the location publisher. Each of them, posts a new message to the event bus, as soon as there is a new data obtained in the sensors. The eventbus, distributes the occurrences to the registered fragments, that reacts to the data according the selected approach.

4.4.5 GreenDAO Framework

GreenDAO is an Object-relational mapping (ORM) that uses the SQLite engine to develop for databases. As any other ORM, this option relieves the usage of raw database queries and automatically maps an result set to an business object.

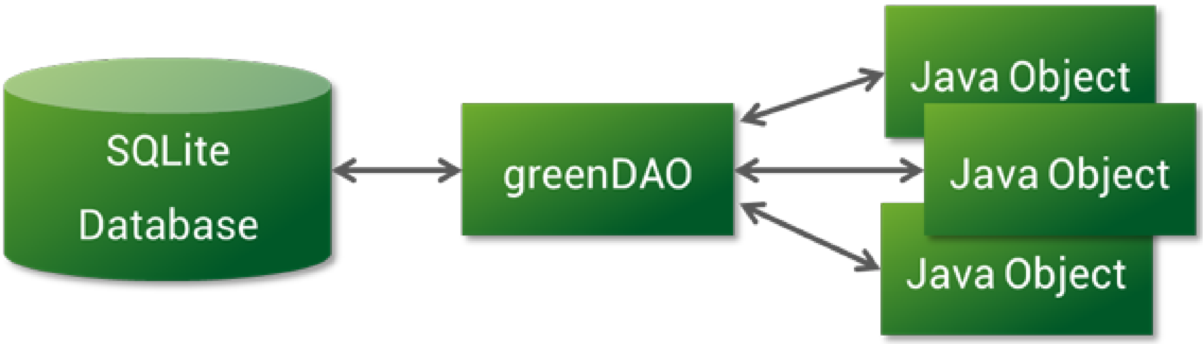


Figure 4.7: greenDAO

This ORM, acts as an interface for the data storage, that is presented in the next section.

4.5 Data Storage

AccuHeight uses a database to storage items to be used along the different user sessions, persisting data to make the system fluid and personalized. Figure 4.8 shows the schema applied in the system.

The above schema demonstrates the system repository tables, and each of the table is a domain object on behalf of greenDAO.

The GribFiles table, is the storage responsible to hold the files that are downloaded to the device through the GribService.

The table named Logs, has as its responsibility, to store system events. It saves both, exception messages to further analysis, and simple event logs to keep track system health.

Parameters is a table where user preferences are found.

And the last table is the Samples, where all the samples that the system retrieves go, to be used to post-processing.

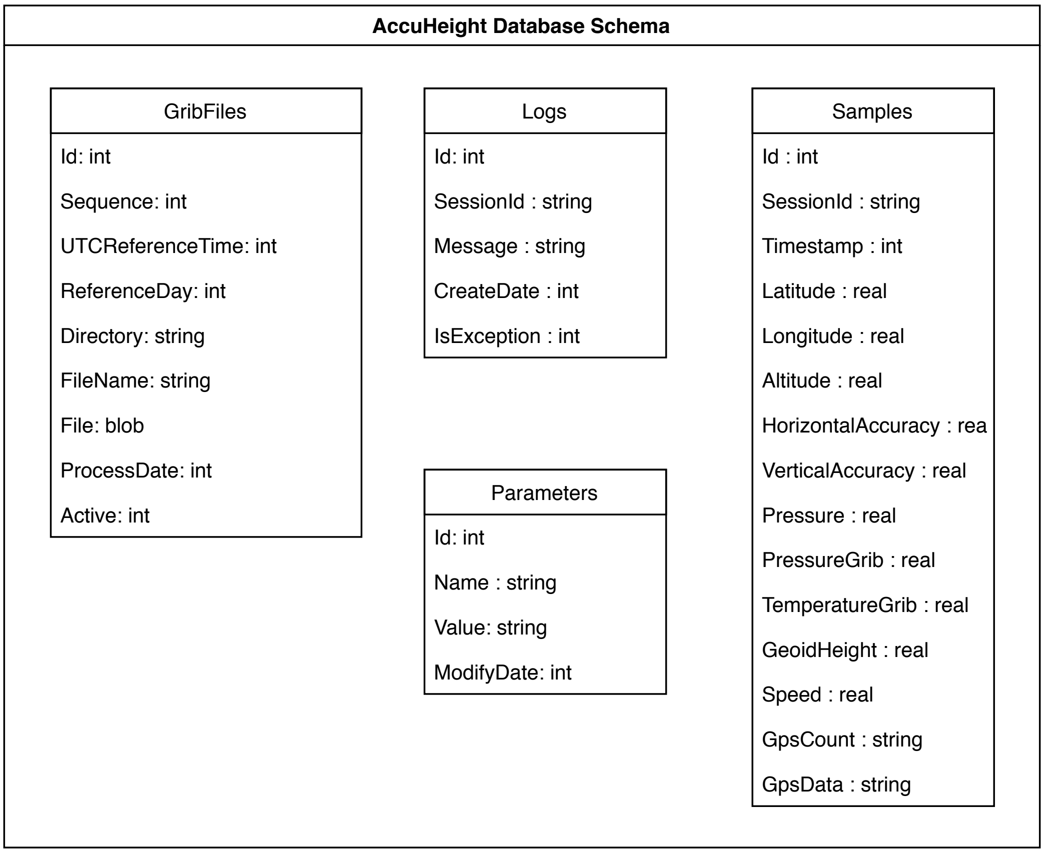


Figure 4.8: AccuHeight Database Schema

GribFiles	
Id	Autoincrement / Unique identifier
Sequence	The temporal order of GRIB files
UTCReferenceTime	UTC Time of the GRIB file
ReferenceDay	Day of the GRIB file
Directory	NOAA structure folder name
FileName	File name of the GRIB file
File	File content
ProcessDate	Date of download
Active	Defines weather the file is active or not

Table 4.1: GribFiles Table Structure

4.6 Features

AccuHeight was designed to estimate the height under several scenarios, by using different methodologies and parameterization. It contains an overview page, the implementation of the 8 methods studied, parameters to tune the default values of the application, logs to follow up the system health and data export to post-processing values in another environment.

Logs	
Id	Autoincrement / Unique identifier
SessionId	The unique session identifier
Message	The message associated to the log
CreateDate	The date of the log
IsException	Splits logs that represent an exception to a simple log

Table 4.2: Logs Table Structure

Parameters	
Id	Autoincrement / Unique identifier
Name	The name of the parameter
Value	The value of the parameter
ModifyDate	The last date that parameter was modified

Table 4.3: Parameters Table Structure

4.6.1 Overview

The overview is the landing page. Since the fragment sits on an activity that distributes the sensor values, it is possible to follow up the sensor variations. For the pressure, it was used a label at the left top corner, and for the GPS, it was used not only a label system but also a graph of the GPS height variations. 4.9 is an example of visualization when using the application for the first time.

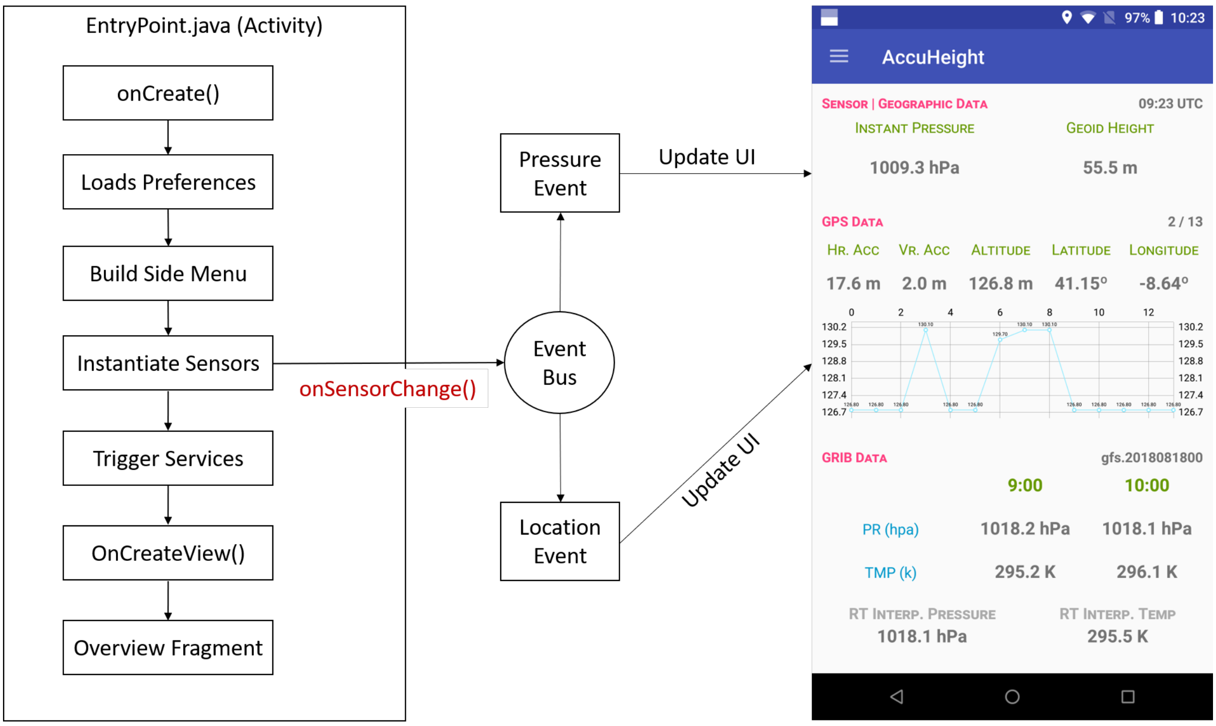


Figure 4.9: AccuHeight Overview

Samples	
Id	Autoincrement / Unique identifier
SessionId	The unique session identifier
Timestamp	The timestamp of the sample
Latitude	Latitude from location sensor
Longitude	Longitude from location sensor
Altitude	Altitude from location sensor
HorizontalAccuracy	Horizontal accuracy from location sensor
VerticalAccuracy	Vertical accuracy from location sensor
Pressure	Pressure from the pressure sensor
PressureGrib	Interpolated pressure at mean sea level from current Grib File
TemperatureGrib	Interpolated temperature at mean sea level from current Grib File
GeoidHeight	The last geoid fetched from the geoid service
Speed	Speed from location sensor
GpsCount	Satellites in view concatenated with number of satellites
GpsData	Detailed information of each satellite that constructed the last location

Table 4.4: Samples Table Structure

From top-down, it is possible to follow the variations of the pressure sensor, in hectopascal at left, and the geoid height to the current location at right. on top of it, the current UTC time, that turns to be useful when handling GRIB files.

In the middle, there are data regarding the GPS. The presented table shows the Horizontal Accuracy, Vertical Accuracy, Altitude, Latitude and Longitude fetch from the last location update.

In the bottom, it is shown the forecast data regarding the current UTC time, and the next one, in order to be possible to interpolate values between time. The table presented contains the fetched pressure at mean sea level, and temperature 2 meters above ground.

Bellow that information, it is found the real time interpolation of the pressure, along with the temperature, in kelvin.

The next section is addressed to the presentation of the user interface regarding the methods.

4.6.2 Methods Screens

Every implemented method have a similar layout, like shown in figure 4.10. The user interface is split in two parts, where the top part is destined to real time data from sensors, variables, and calibrations. There, resides the most important data from which, each of the methodology studied estimates the altitude.

Such information is useful for the user, to understand if the system is responding correctly to the commands.

At the bottom of the same figure, there are two possible views depending on the selected method. The main goal of such view, is to present the real time estimated altitude. The variation of it, also shows a debug window, where it is possible to visualize the current results of an interaction.

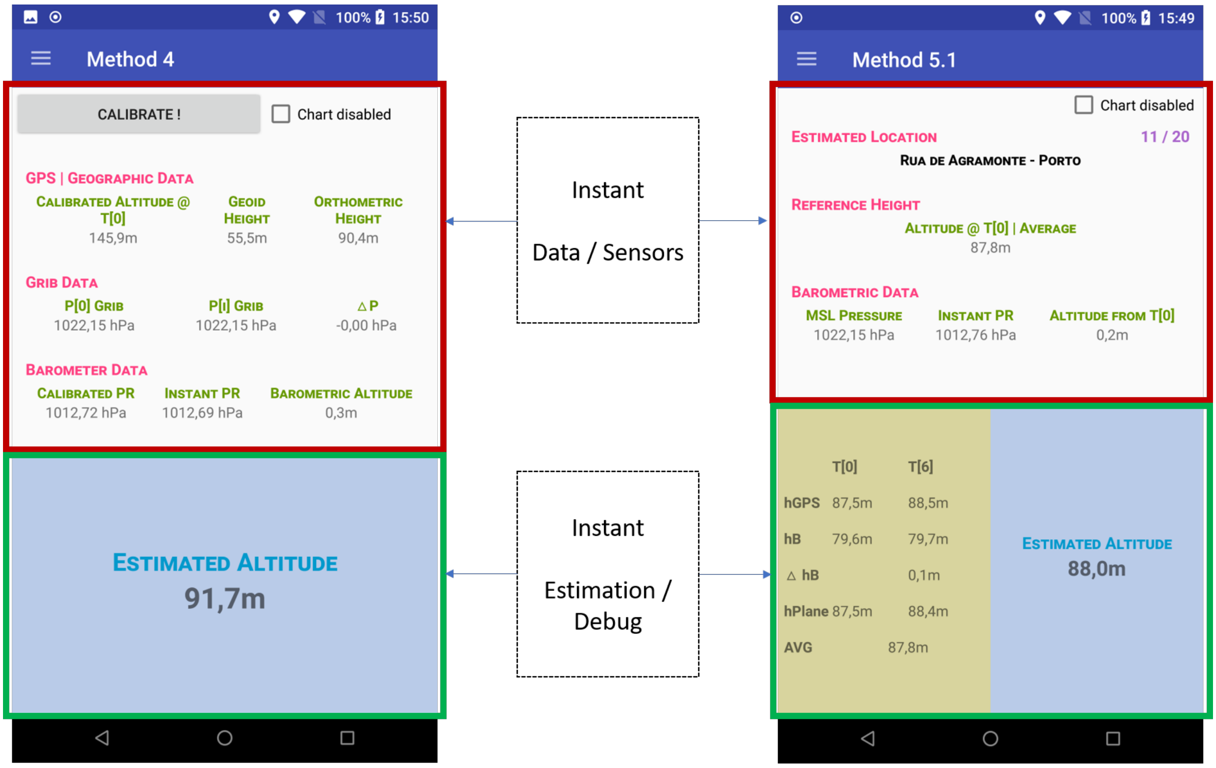


Figure 4.10: Methods Layout

As an alternate view of the system, it was implemented a graphical visualization, where a line chart is plotted in the screen in order to follow the altitude variations as a time series. To switch to such screen, the user must toggle the chart option at the top right corner, like shown in the 4.11 figure.

4.6.3 Options

In addition to the implemented methodologies, there are options that allows the user to customize the experience, and follow up the health of the system.

4.6.3.1 Parameters

Parameters were implemented in order to affect the calculations given user input, and turn the application flexible due to a determined condition.

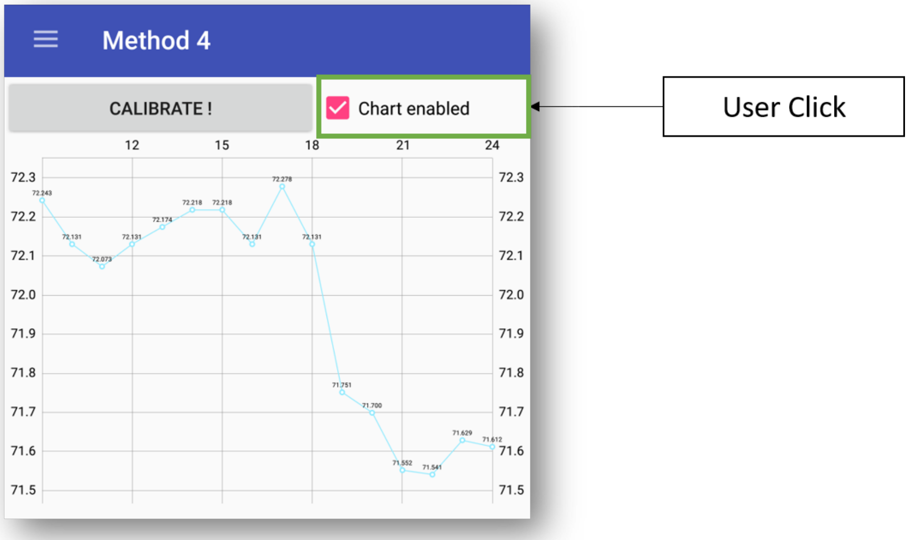


Figure 4.11: Graph Layout

All user preferences are saved in the shared preferences, provided by the android framework.

In figure 4.12, it is possible to view some of the categories that are contained in the preference windows, such as: 1) Calibration, 2) GPS, 3) Kalman Filter, 4) Methods

In the calibration section, it is possible to set two configurations. The first one, is the barometer calibration, where it is set an offset, aiming to affect the pressure provided by the sensor. Typically, this offset is the difference between the phone measurement and a weather station measurement when both are located at the same point. It is also possible to set an calibration timeout, in seconds, regarding the methods that execute a calibration before outputting the height estimation. Such value leads the system to consume samples proportionally to the timeout amount.

The GPS section handles options regarding the accuracy of the satellites outputs. In it, the accuracy of the vertical and horizontal data is set, leading the calculations to discard values received that are above the user input.

Kalman Filter group deals with the GPS error in measurement. As discussed in previous chapters, this kind of filter needs an default value to the apparatus error, and such value is determined in the section. By default, it is set an error of 25 meters.

In the last category - Methods - it is configured data related to the calculations defaults. Height above ground is the first option, that allows the user to set an default height to be considered, to avoid the user to put the smartphone in the ground to calibrate. Window size, is the system last option, regarding the chunk size that should be considered in the method 6 and 6.1

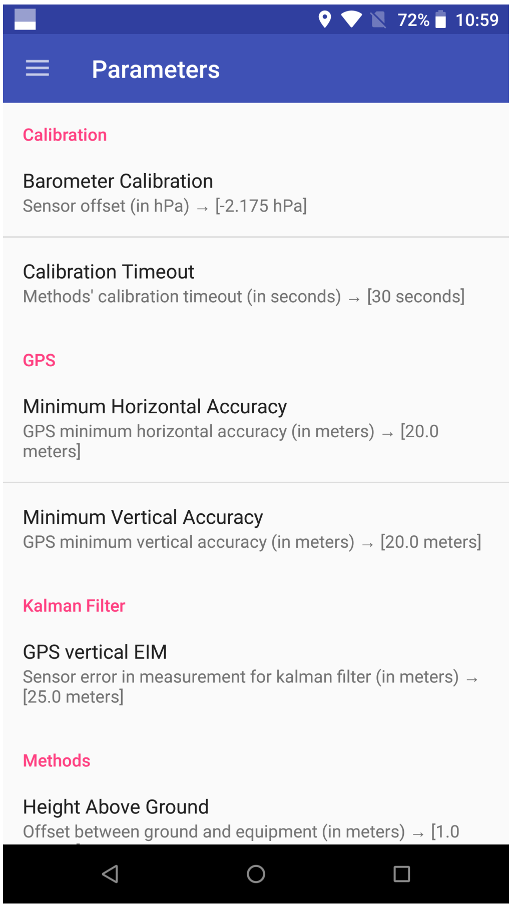


Figure 4.12: Parameters

4.6.3.2 Logs

Aiming to understand further the system actions and health, a log system was developed, registering all the milestones and errors caused in the environment. Every important action or error is saved along with the timestamp, in a table in the database.

The actions were created so that, the used may have a deep knowledge of the steps system are following. In the other hand, the errors - highlighted with a red background - are meant for developer analysis and debugging.

It is possible to view an example of logs on figure 4.13, when the service is started.

4.6.3.3 Export Data

Along with the system calculations, sensor values are stored in the database asynchronously, allowing the user to export the content of the table. Export Data is the third resource in the options menu, and when clicked, the user is prompt options to upload a zip file, which contains the GRIB files used in the data collection, and an file in Comma-Separated Values (CSV) format that holds all the logs in a structured form.

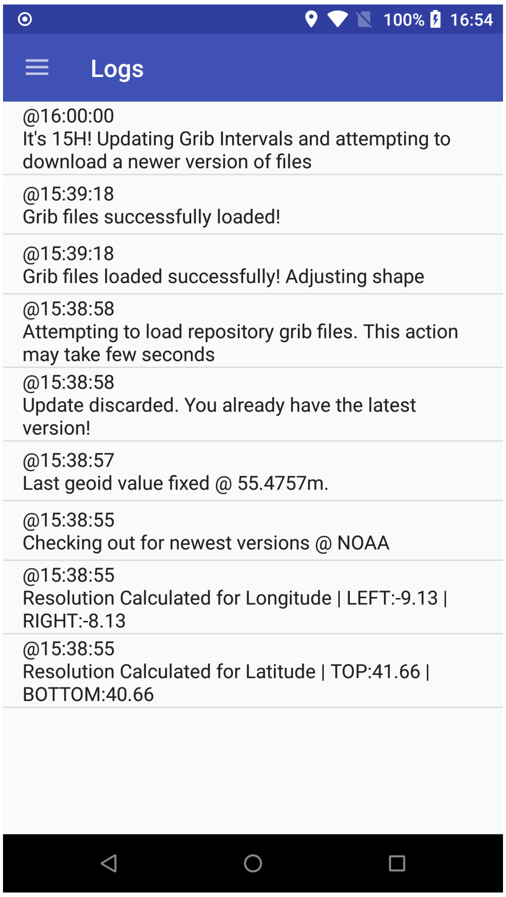


Figure 4.13: Logs

Such option opens the possibility to post process the data.

A	B	C	D	E	F	G	H	I	J	K
Day	Hour	Latitude (Degree)	Longitude (Degree)	Altitude (Meters)	H. Accuracy (Meters)	V. Accuracy (Meters)	Pressure (hPa)	Pressure Interpolated @ Grib (hPa)	Temperature Interpolated @ Grib (hPa)	Geoid (Meters)
26/08/2018	11:05:09	41.1584867	-8.6331426	160.0	13.630000114440918	46.0	1004.0450439453125	1016.2356214192708	293.90662541707354	55.4757
26/08/2018	11:05:09	41.1584867	-8.6331426	160.0	13.630000114440918	46.0	1004.0450439453125	1016.2356214192708	293.90662541707354	55.4757
26/08/2018	11:05:10	41.1584544	-8.6331723	159.0	13.222999572753906	40.0	1004.0258178710938	1016.2356130642362	293.9068361070421	55.4757
26/08/2018	11:05:10	41.1584544	-8.6331723	159.0	13.222999572753906	40.0	1004.0258178710938	1016.2356130642362	293.9068361070421	55.4757
26/08/2018	11:05:11	41.1584262	-8.6331984	159.0	11.967000007629395	32.0	1004.0099487304688	1016.2356130642362	293.9068361070421	55.4757
26/08/2018	11:05:11	41.1584262	-8.6331984	159.0	11.967000007629395	32.0	1004.0099487304688	1016.2356130642362	293.9068361070421	55.4757
26/08/2018	11:05:12	41.158407	-8.6332156	159.0	10.336999893188477	26.0	1004.0213012695312	1016.2356047092014	293.90704679701065	55.4757
26/08/2018	11:05:12	41.158407	-8.6332156	159.0	10.336999893188477	26.0	1004.0213012695312	1016.2356047092014	293.90704679701065	55.4757

Figure 4.14: Export Data (A-K)

<table>
<tr><th>L</th><th>M</th><th>N</th></tr>
<tr><td>Speed (Meters/Second)</td><td>GNSS [Used | InView]</td><td>GNSS (Foreach Used) Collection [Constellation (Name) | AzimuthDegrees (Degree) | CarrierFrequency (Hz) | CarrierToNoise (Hz) | ElevationDegrees (Degree)]</td></tr>
<tr><td>0.0</td><td>6 | 19</td><td>[GPS | 314,00 | 0,00 | 22,30 | 13,00] , [GPS | 288,00 | 0,00 | 22,60 | 33,00] , [GPS | 88,00 | 0,00 | 19,70 | 40,00] , [GPS | 50,00 | 0,00 | 19,00 | 31,00] , [GPS | 147,00 | 0,00 | 22,20 | 45,00] , [GPS | 322,00 | 0,00 | 18,10 | 63,00]</td></tr>
<tr><td>0.0</td><td>6 | 19</td><td>[GPS | 314,00 | 0,00 | 22,30 | 13,00] , [GPS | 288,00 | 0,00 | 22,60 | 33,00] , [GPS | 88,00 | 0,00 | 19,70 | 40,00] , [GPS | 50,00 | 0,00 | 19,00 | 31,00] , [GPS | 147,00 | 0,00 | 22,20 | 45,00] , [GPS | 322,00 | 0,00 | 18,10 | 63,00]</td></tr>
<tr><td>0.0</td><td>6 | 19</td><td>[GPS | 314,00 | 0,00 | 21,40 | 13,00] , [GPS | 288,00 | 0,00 | 24,20 | 33,00] , [GPS | 88,00 | 0,00 | 22,10 | 40,00] , [GPS | 50,00 | 0,00 | 19,20 | 31,00] , [GPS | 147,00 | 0,00 | 22,70 | 45,00] , [GPS | 322,00 | 0,00 | 19,90 | 63,00]</td></tr>
<tr><td>0.0</td><td>6 | 19</td><td>[GPS | 314,00 | 0,00 | 21,40 | 13,00] , [GPS | 288,00 | 0,00 | 24,20 | 33,00] , [GPS | 88,00 | 0,00 | 22,10 | 40,00] , [GPS | 50,00 | 0,00 | 19,20 | 31,00] , [GPS | 147,00 | 0,00 | 22,70 | 45,00] , [GPS | 322,00 | 0,00 | 19,90 | 63,00]</td></tr>
<tr><td>0.0</td><td>6 | 19</td><td>[GPS | 314,00 | 0,00 | 21,00 | 13,00] , [GPS | 288,00 | 0,00 | 24,60 | 33,00] , [GPS | 88,00 | 0,00 | 19,50 | 40,00] , [GPS | 50,00 | 0,00 | 19,30 | 31,00] , [GPS | 147,00 | 0,00 | 23,30 | 45,00] , [GPS | 322,00 | 0,00 | 18,60 | 63,00]</td></tr>
<tr><td>0.0</td><td>6 | 19</td><td>[GPS | 314,00 | 0,00 | 21,00 | 13,00] , [GPS | 288,00 | 0,00 | 24,60 | 33,00] , [GPS | 88,00 | 0,00 | 19,50 | 40,00] , [GPS | 50,00 | 0,00 | 19,30 | 31,00] , [GPS | 147,00 | 0,00 | 23,30 | 45,00] , [GPS | 322,00 | 0,00 | 18,60 | 63,00]</td></tr>
<tr><td>0.0</td><td>6 | 19</td><td>[GPS | 314,00 | 0,00 | 20,20 | 13,00] , [GPS | 288,00 | 0,00 | 22,30 | 33,00] , [GPS | 88,00 | 0,00 | 19,50 | 40,00] , [GPS | 50,00 | 0,00 | 18,80 | 31,00] , [GPS | 147,00 | 0,00 | 21,00 | 45,00] , [GPS | 322,00 | 0,00 | 20,30 | 63,00]</td></tr>
<tr><td>0.0</td><td>6 | 19</td><td>[GPS | 314,00 | 0,00 | 20,20 | 13,00] , [GPS | 288,00 | 0,00 | 22,30 | 33,00] , [GPS | 88,00 | 0,00 | 19,50 | 40,00] , [GPS | 50,00 | 0,00 | 18,80 | 31,00] , [GPS | 147,00 | 0,00 | 21,00 | 45,00] , [GPS | 322,00 | 0,00 | 20,30 | 63,00]</td></tr>
</table>

Figure 4.15: Export Data (L-N)

Chapter 5

Results and Analysis

This chapter was built aiming to compare the results of the studied methodologies, and quantify its reliability under different scenarios.

5.1 Test Environment

To perform an offline experiment, *AccuHeight* was installed in a Google Nexus 6 device, aiming to gather data to be post-processed, while working in online mode. For this purpose, the application is prepared to store the data, and export it when requested.

With the data exported, a few scenarios were explored to understand the behaviour of the studied methodologies, simulating different contexts in which the user may be. The first test case was a mixed environment scenario, where it explores the ability of the application to react to different conditions such as entering and exiting an indoor space where the reception of the GPS signal is obstructed, thus leading to a very inaccurate altitude estimations, and test if the GPS values are confident enough to be used as a reference altitude when the signal arrives in good conditions, in an outdoor condition. A full-indoor testing scenario has also been taken into consideration, aiming at an environment in which the GPS signal is non-existent, and therefore the application will only use the the barometric sensor. Full-outdoor experiences also were taken into place to simulate a scenario where the GPS reception is potentially good.

The data gathered in the online experiments, was used to plot the graphs shown in this chapter, post-processed using a *Java* application developed for the purpose. The behaviour of the offline experiment using the mentioned application, produces the same output that the online would in real-time. All the plots shown in this chapter were built using Microsoft Excel.

5.1.1 Metrics

To measure the performance of the predictions, a comparison between the Ground Truth (GT) Altitude and the altitude of each studied methodology was performed, using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). All errors were calculated based on outdoor samples, since in indoor locations it was not possible to obtain the real altitude.

The GT was fetched for each sample of the exported data, using the Google Elevation API external service.

In some test cases, the Ground Truth Adjusted (GTA) was considered, aiming to adjust the GT, in an hypothetical context where the GPS altitude reference has the smallest error compared to it. This adjustment causes a vertical shift, that is calculated using outdoor samples only, as the result of the difference between the average of GPS altitude and the average of GT values. The resulting bias, is added in each GT sample, originating the GTA.

In the case of windowed-methodologies, the size of the window in the experiment was of 500 samples.

We now start presenting the first scenario.

5.2 Mixed Scenario

The mixed scenario aims to test the behavior of the studied methodologies, in a track that passes through indoor locations, with little or no reception of GPS signal, and by outdoor locations, which will possibly bring more accurate estimates regarding the GPS sensor. With this scenario, it would be possible to observe the behavior of the application in environments where the GPS did not have reception but based on previously read measurements where the GPS reception was good, it was possible to continue to estimate the altitude based on the previous estimated reference.

To start the tests, a calibration period of 2 minutes on the balcony of a 5-floor building, was made so that the sensor data had less error values to be used as a reference. After this period, a track towards the nearest metro station was started on foot, that lasted about 5 minutes. A metro trip was carried out, and for a period of 4 minutes the system was exposed to a very weak or absent GPS signal. When leaving the destination metro station, in an open environment, we waited for a bus that would take one more trip to the destination building. The bus ride lasted at least 15 minutes to reach the final destination. Already inside the building, that contains 2 floors, few meters were walked, towards the elevator. Through it, a trip to the $\boldsymbol{4^{th}}$ floor was held until arriving at the terrace, which is open, the device rested above a table for about 10 minutes, and with this the experiment finished.

The figure 5.1 expresses the altitude results on this itinerary for the methodologies without fusing sensors. The dotted line in red, represents the altitude read by the GPS signal, and the

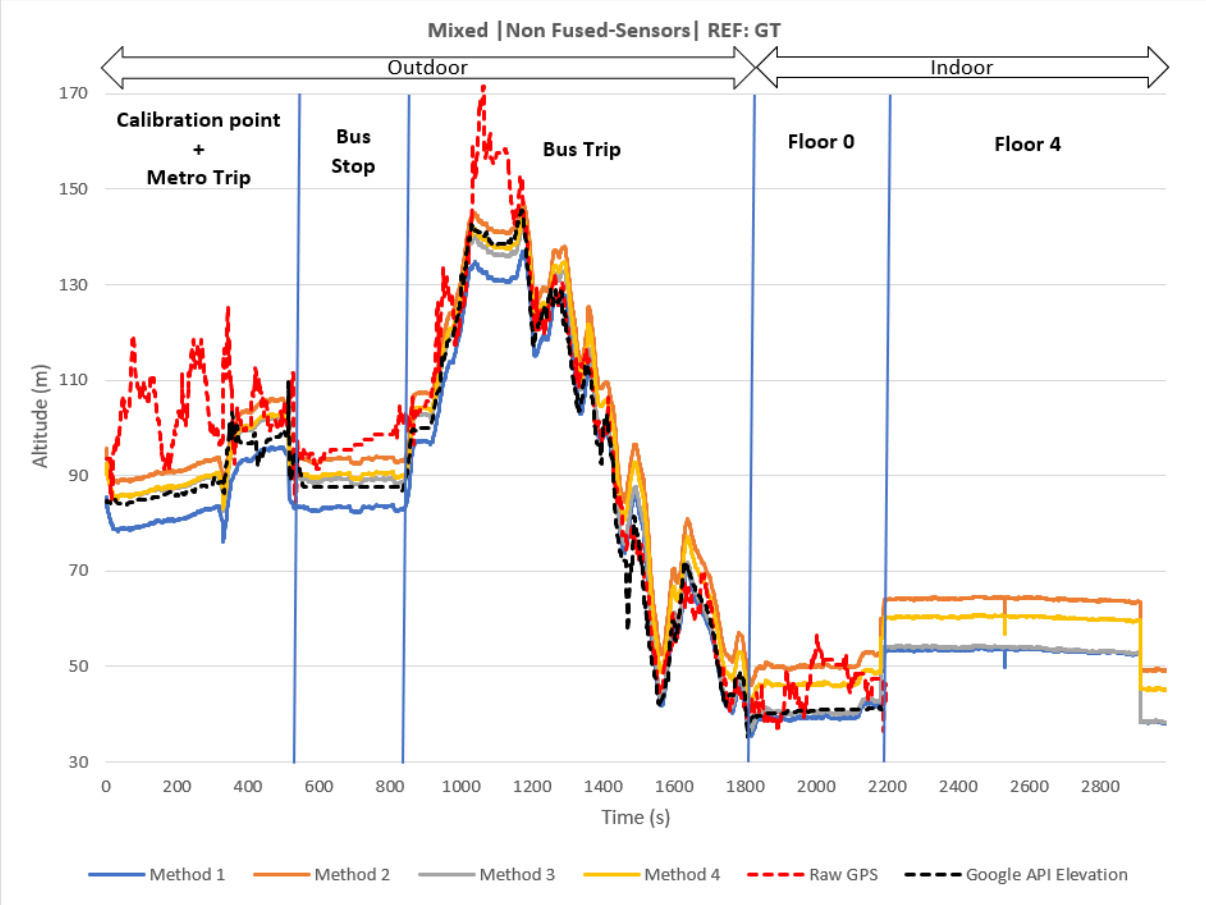


Figure 5.1: Non Fused Sensors approaches applied to mixed environment | REF: GT

black dotted line, shows the GT, until the destination building arrival. Table 5.1 shows the errors calculated for this context.

	GPS	Method 1	Method 2	Method 3	Method 4
MAE	13,193	4,179	7,665	2,446	4,556
RMSE	15,716	5,017	8,761	3,657	5,907

Table 5.1: Non-Fused Sensors approaches errors applied to mixed environment | REF: GT

The experiment shows an advantage of the method 3, possibly due to the good altitude reference calculated while in the calibration time. It can be observed that the method 1 had a drop after the calibration, which is explained by the point in which it occurred, that should be at ground level. The GT calculated on the timeout of the calibration for method 1, does not correspond to the GT at the balcony, causing a bias on its results.

During the subway and bus trips, we can observe that the methods approximately follow the GT. In the last part of the experiment, already on the destination building, the barometer can clearly identify a rise of approximately 20 meters, explaining the visit to the $\mathbf{4}^{th}$ floor.

We now present the behaviour of the methods based on sensor fusion, represented in figure

5.2.

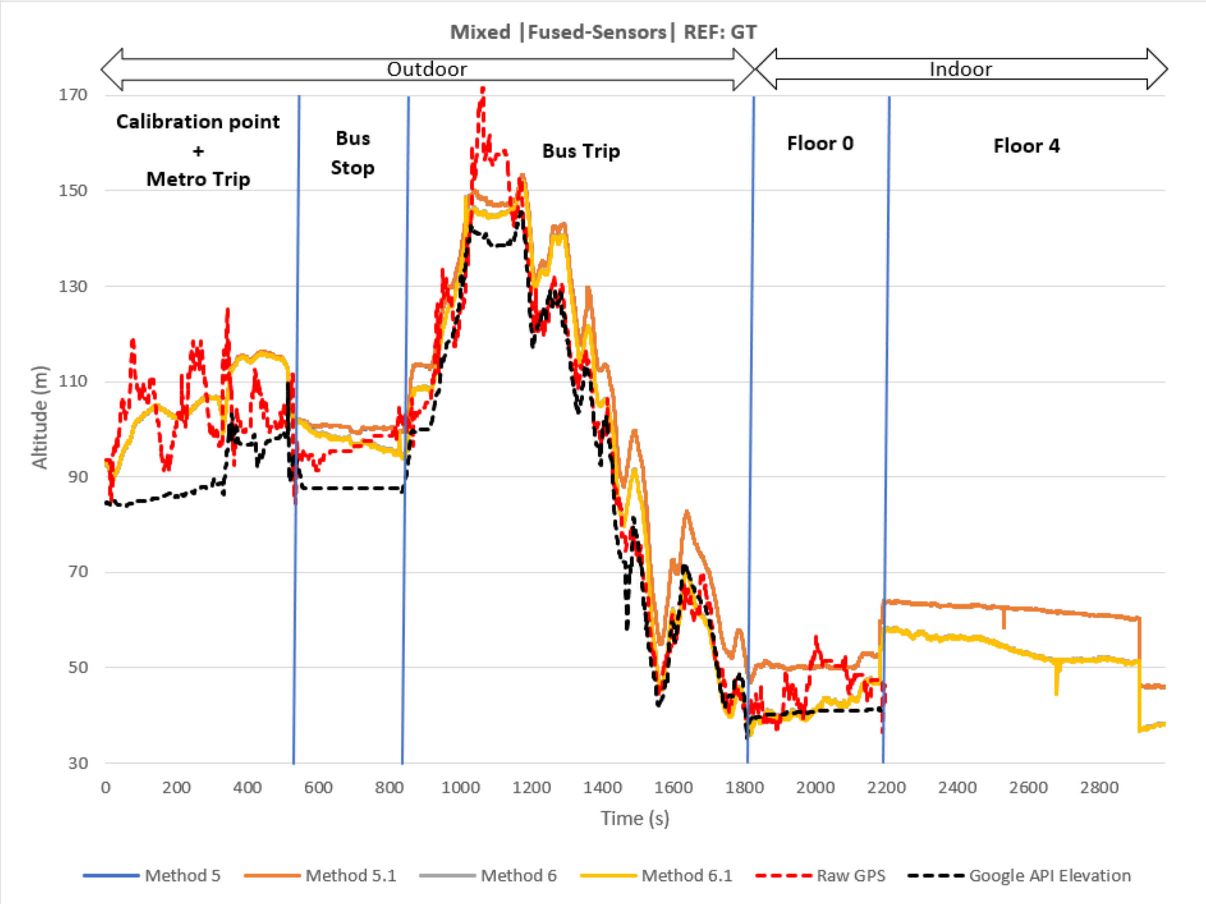


Figure 5.2: Fused Sensors approaches applied to mixed environment | REF: GT

In figure 5.2 we can observe a stable behaviour after instant 800. Previously, it was verified the existence of values with a higher error, possibly due to the existence of many buildings and trees on the way to the subway station, which degraded the estimate obtained by GPS, where it can be observed variations of up to 20 meters compared to the GT. The RMSE obtained with the fused methodologies, described in table 5.3, shows an error reduction up to 33% with regard to the errors of the raw GPS measurements.

	Errors (REF: GT)				
	GPS	Method 5	Method 5.1	Method 6	Method 6.1
MAE	13,193	13,092	13,115	8,902	8,831
RMSE	15,716	13,754	13,778	10,691	10,623

Table 5.2: Fused Sensors approaches errors applied to mixed environment | REF: GT

Although the results presented have shown some efficiency compared to standalone GPS measurements, it can be observed a bias associated with GPS measurements compared to GT, which causes a vertical shift in the estimated measurements by the fusion methods. Calculating the bias that originates the GTA, makes it possible to understand the behaviour of the methodologies if the GPS altitude reference had the minimum error compared to GT. Figure 5.3 shows the results applying this adjustment.

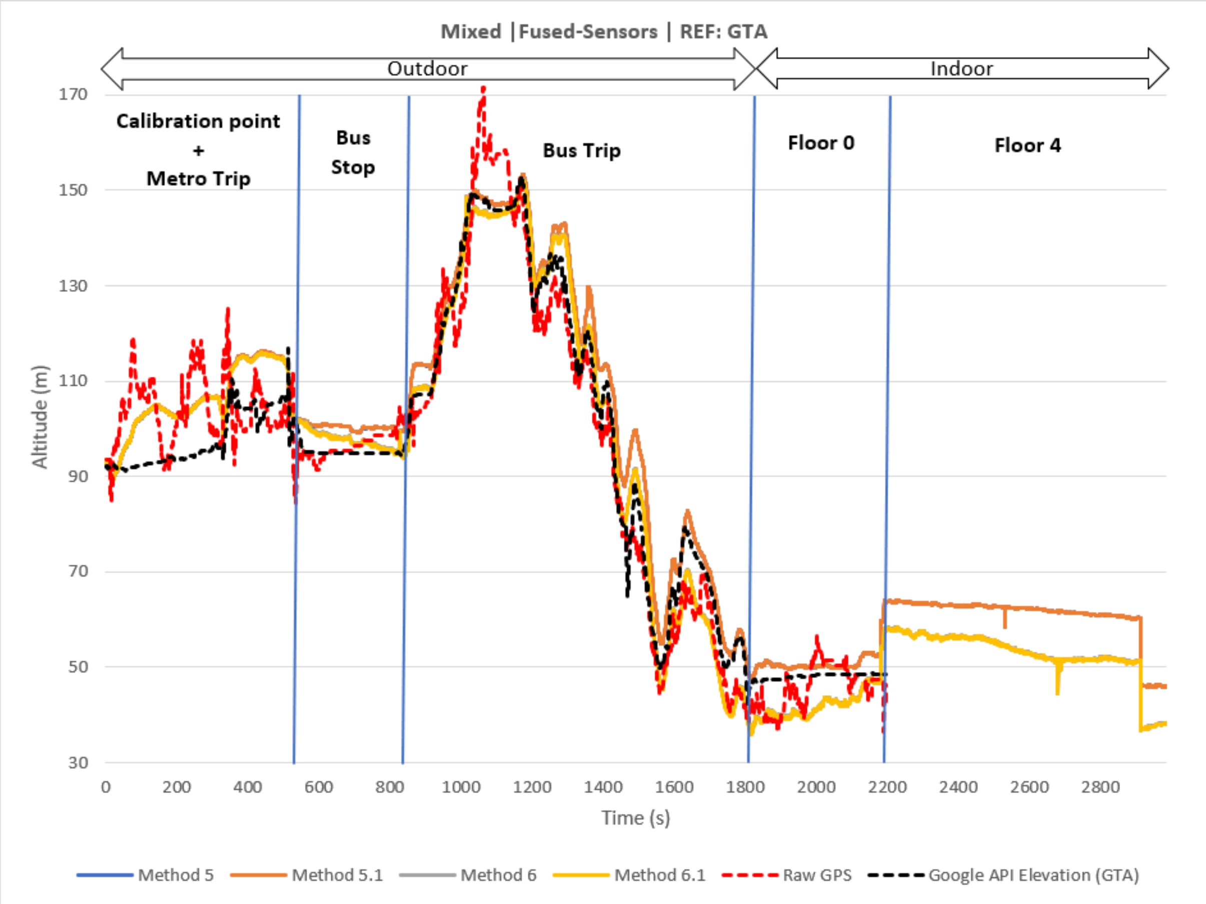


Figure 5.3: Fused Sensors approaches applied to mixed environment | REF: GTA

Table 5.3 shows the errors in this contexts

	Errors (REF: GTA)				
	GPS	Method 5	Method 5.1	Method 6	Method 6.1
MAE	10,938	5,748	5,767	5,521	5,555
RMSE	12,742	7,076	7,098	6,663	6,718

Table 5.3: Fused Sensors approaches errors applied to mixed environment | REF: GTA

We can observe that, when the GPS reference altitude provided is good, the studied approaches shows that the altitude estimation can be at least 1,8x better than GPS standalone.

We now present the full-indoor context.

5.3 Indoor Scenario

To perform the indoor tests, we went to a small shopping in the city, which contains 6 floors, of which 4 are below the ground level. We collected data for 2 minutes at the main door of the shopping in order to serve as a base reference for the calculation of methods. The first goal was to reach the highest floor of the building, and it was necessary to cross 2 sets of escalators. When reaching the top, we walked through the perimeter of the floor, until a set of stairs, that gave access to the floor -1. To test the behavior of the methodologies using an elevator, it was necessary to climb again to the top of the shopping, and the same route described previously was accomplished, only adding a set of stairs that connected the floor -1 to 0. On the top floor and in the elevator, the destination was the lowest floor possible that gave access to the garages. From there, the elevator had as its destination the floor 0, where the experiment was finished.

The analysis of this environment will not be given through an MAE and RMSE table, since it was not possible to obtain the actual altitudes of each floor of the shopping. The analysis was performed by observational events.

Using non-fusion methods, the results obtained are expressed in figure 5.4.

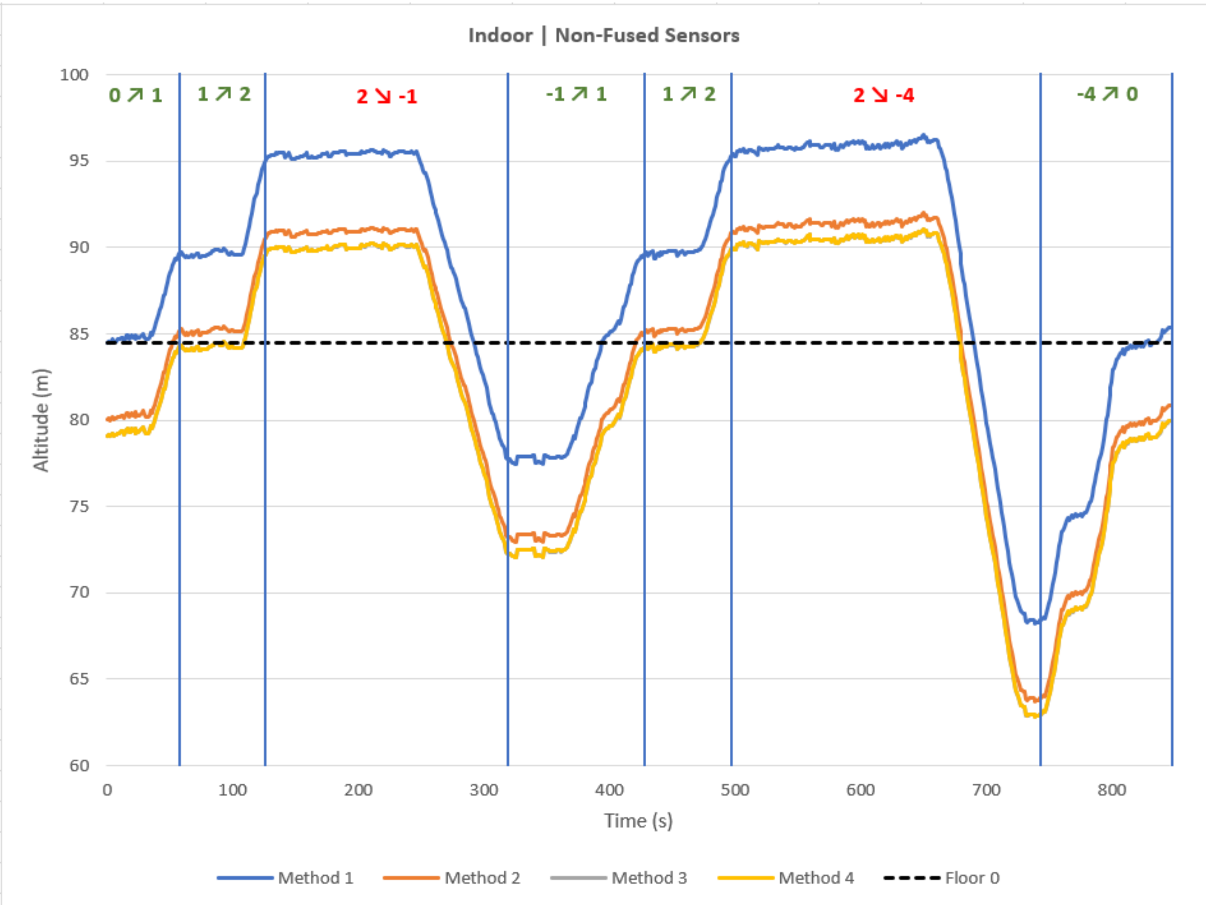


Figure 5.4: Non Fused Sensors approaches applied to mixed environment

In the experiment that used the non-fused sensor methods, it was possible to observe with great clarity the moments when there was a rise or a decline in altitude, allowing even a deduction of the number of floors that the shopping has. The first method presented a high degree of agreement, because it was able to identify the correct altitude at the entrance of the shopping and keep it at the end of the experiment, when leaving the same door in which the experiment began, suggesting that along this experiment, the barometric values captured were not influenced by meteorological variations. The other methods show similar behavior, apart from a bias that was observed, which are due to different factors. In method 2, there was possibly a poor forecast data provided in that moment, that affected the barometer calculations, even calibrated. For methods 3 and 4, which presented identical results between them, the observed shift was due to the initial altitude calibration obtained via GPS measurement, in which the altitude estimated by the sensor did not present the expected vertical accuracy, which is possibly due to the building environment in which the shopping is located, causing the multipath effect described earlier in this work.

We now present the results in an fusion sensor approach perspective, represented in figure 5.5.

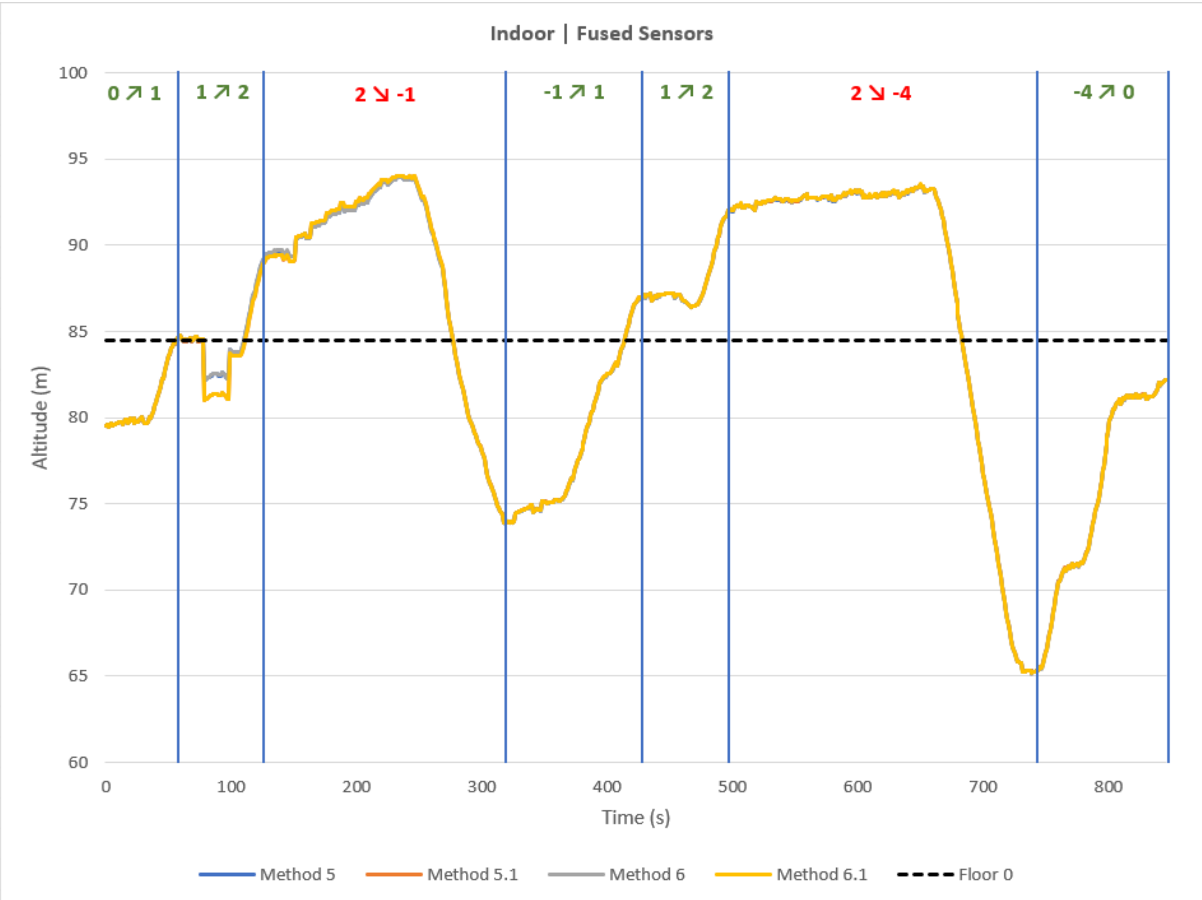


Figure 5.5: Fused Sensors approaches applied to indoor environment

In this scenario, unlike the previous one, it was not possible to see with the same clarity, at the beginning of the experiment the floors traveled. We observed that the graphic resulting from the indoor walk, has as beginning a starting point with a bias, due to the fact of the initial GPS height not being very accurate. In the time interval from 120 seconds to 210 seconds, it was observed an attempt of the system to fix some altitude using new GPS signals that have reached the system in that moment. This can also be observed at the end of the experiment, where there is a difference of approximately 3 meters between the starting point and the point of arrival, which are points that represent the same GT. From the moment the application seems to have ceased to receive GPS signal, at the instant after 350 seconds, the altitude estimate appears to have a very similar form to the methods previously demonstrated, because in this scenario the barometer acts alone, using as reference the latest valid GPS altitude measurements. It can also be observed in this experiment that all fusion methods behave similarly, in an full-indoor environment.

5.4 Outdoor Scenario

This scenario was built with a walk through a path with an extension of 3km having significant altitude variations.

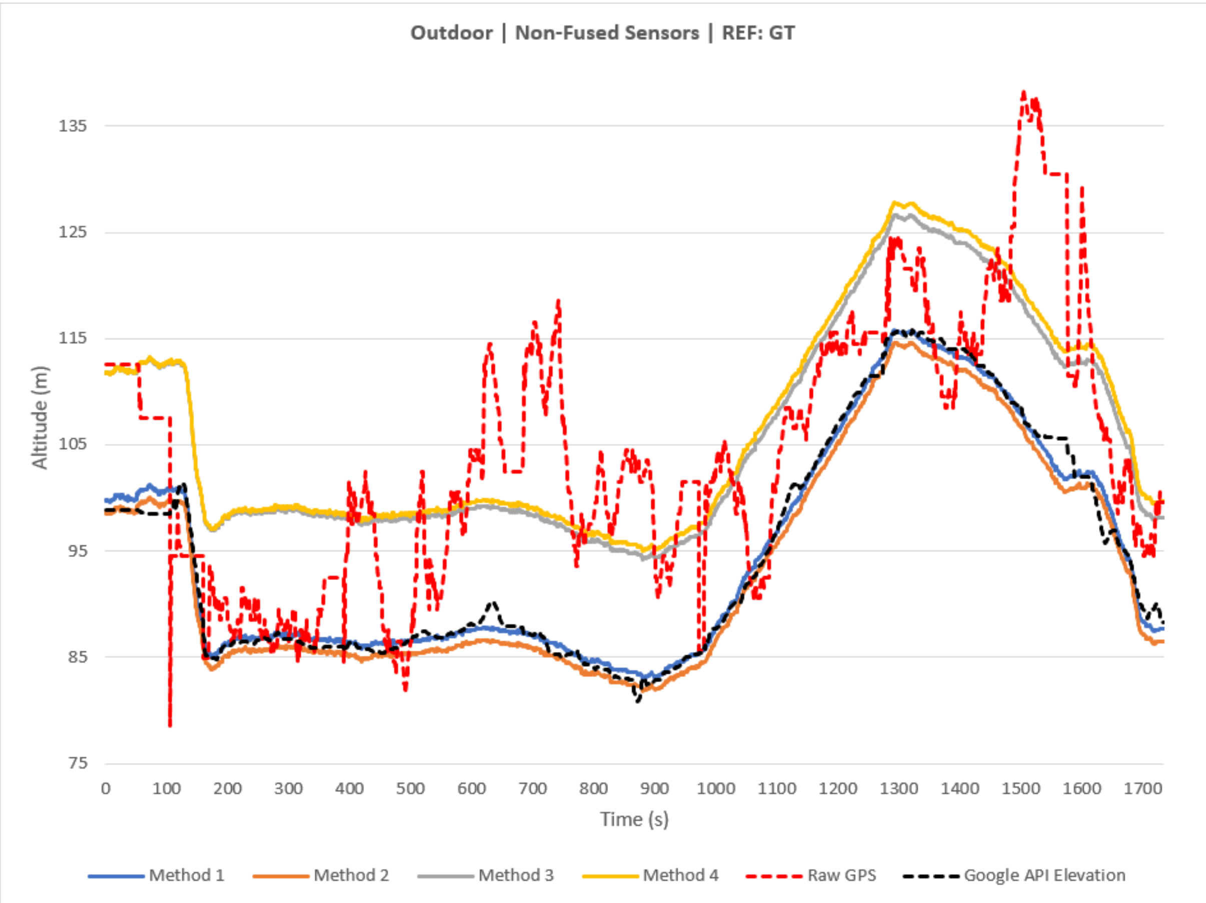


Figure 5.6: Non-Fused Sensors approaches applied to outdoor environment | REF: GT

The figure 5.6 shows a fairly precise behavior of methods 1 and 2. On the other hand, methods 3 and 4 present a relatively higher error. This was due to the fact that initial calibration phase of those methods used GPS samples with poor accuracy. Table 5.4 shows the magnitude of the errors. It is possible to note that method 1 had the best performance, possibly due to the great horizontal accuracy obtained at the time of method start-up. It is also noted that the barometer was able to follow quite precisely the altitude of the traveled path. Regarding method 2, success is due to the well-adjusted calibration of the sensor at the time of the test, and good forecast data.

	GPS	Method 1	Method 2	Method 3	Method 4
MAE	10,072	0,727	1,254	11,310	12,084
RMSE	12,650	1,023	1,498	11,377	12,127

Table 5.4: Non-Fused Sensors approaches errors applied to outdoor environment | REF: GT

Regarding fusion methods, which are represented in the figure 5.7, it can be observed the the results are quite close to the GT. The good performance, is possibly explained due to the good reception of the GPS samples, which continually contributed to the definitions of the plane to be updated with values ever closer to the real ones. The graph shows an advantage of the windowless methods between the time interval of 700 to 1300 seconds, being very close to the estimated actual value. In the final stage of the test, between 1300 and 1500 seconds, after the peak observed in the graph, it is possible to see a faster convergence of the window-based methods.

The 5.5 table shows in detail the performance of the methodologies.

	Errors (REF: GT)				
	GPS	Method 5	Method 5.1	Method 6	Method 6.1
MAE	10,072	7,915	7,905	8,756	8,609
RMSE	12,650	8,356	8,348	9,690	9,551

Table 5.5: Fused Sensors approaches errors applied to outdoor environment | REF: GT

Similarly to what occurred in the mixed scenario, there is a bias between the associated values of GPS and GT. Figure 5.8 shows this correction applied, and the errors associated with the methodologies compared to the actual altitude adjusted can be observed in table 5.6.

	Errors (REF: GTA)				
	GPS	Method 5	Method 5.1	Method 6	Method 6.1
MAE	6,595	2,453	2,461	3,784	3,793
RMSE	8,322	3,125	3,137	4,222	4,237

Table 5.6: Fused Sensors approaches errors applied to outdoor environment | REF: GTA

The table 5.6 shows that the methodology that obtained the smallest error was the methodology 5, and to observe its behavior along the route. A graph was created that is represented in

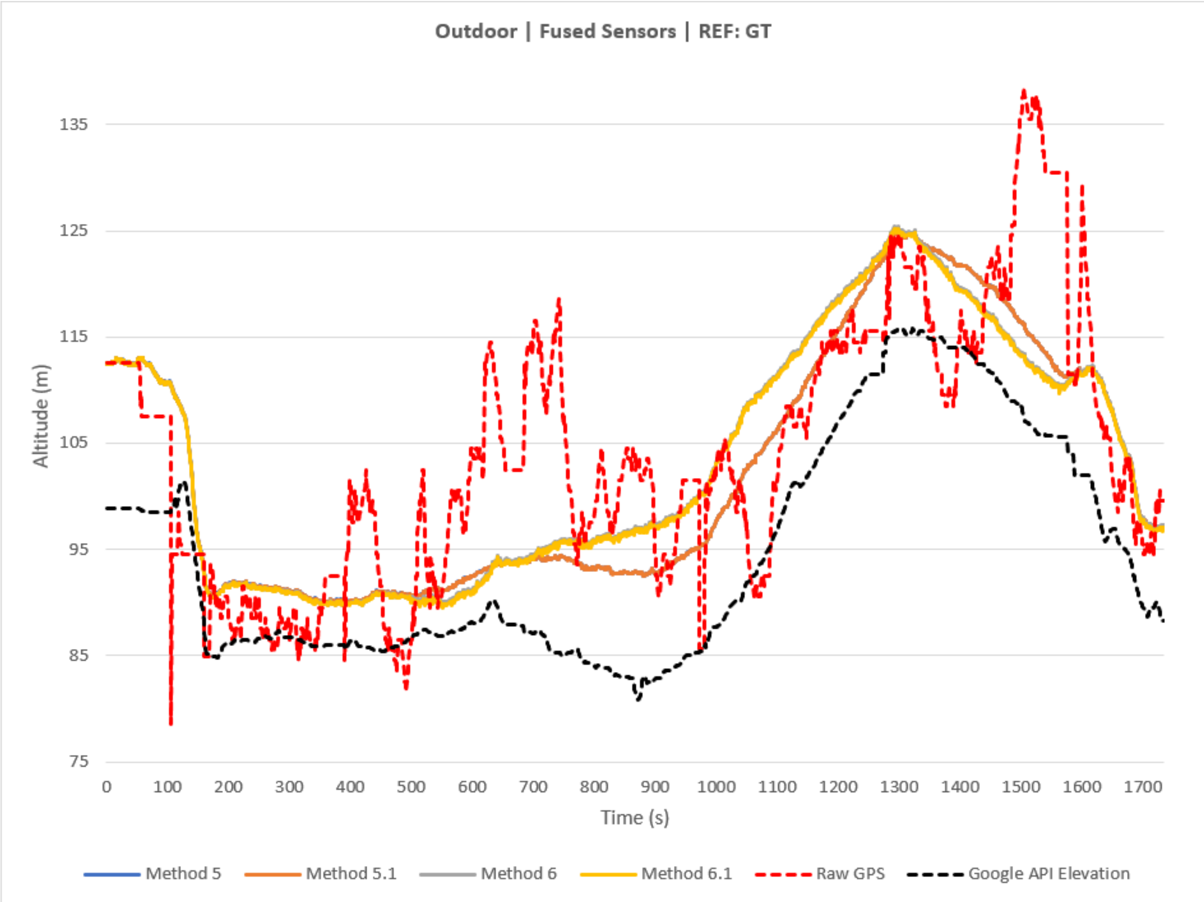


Figure 5.7: Fused Sensors approaches applied to outdoor environment | REF: GT

Figure 5.9, where it is possible to observe the values obtained by the barometer, and the plane from which the method iteratively calculated, as being the base reference altitude.

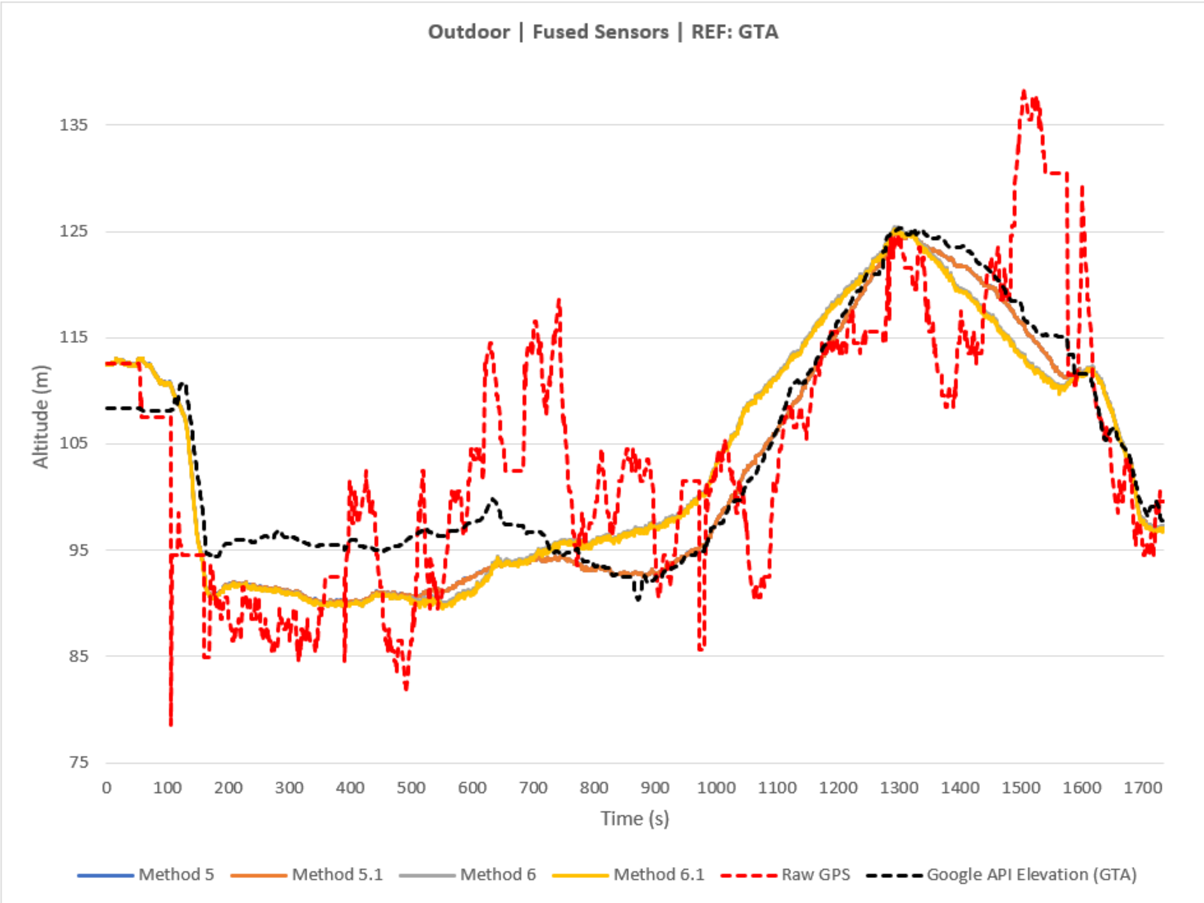


Figure 5.8: Fused Sensors approaches applied to outdoor environment | REF: GTA

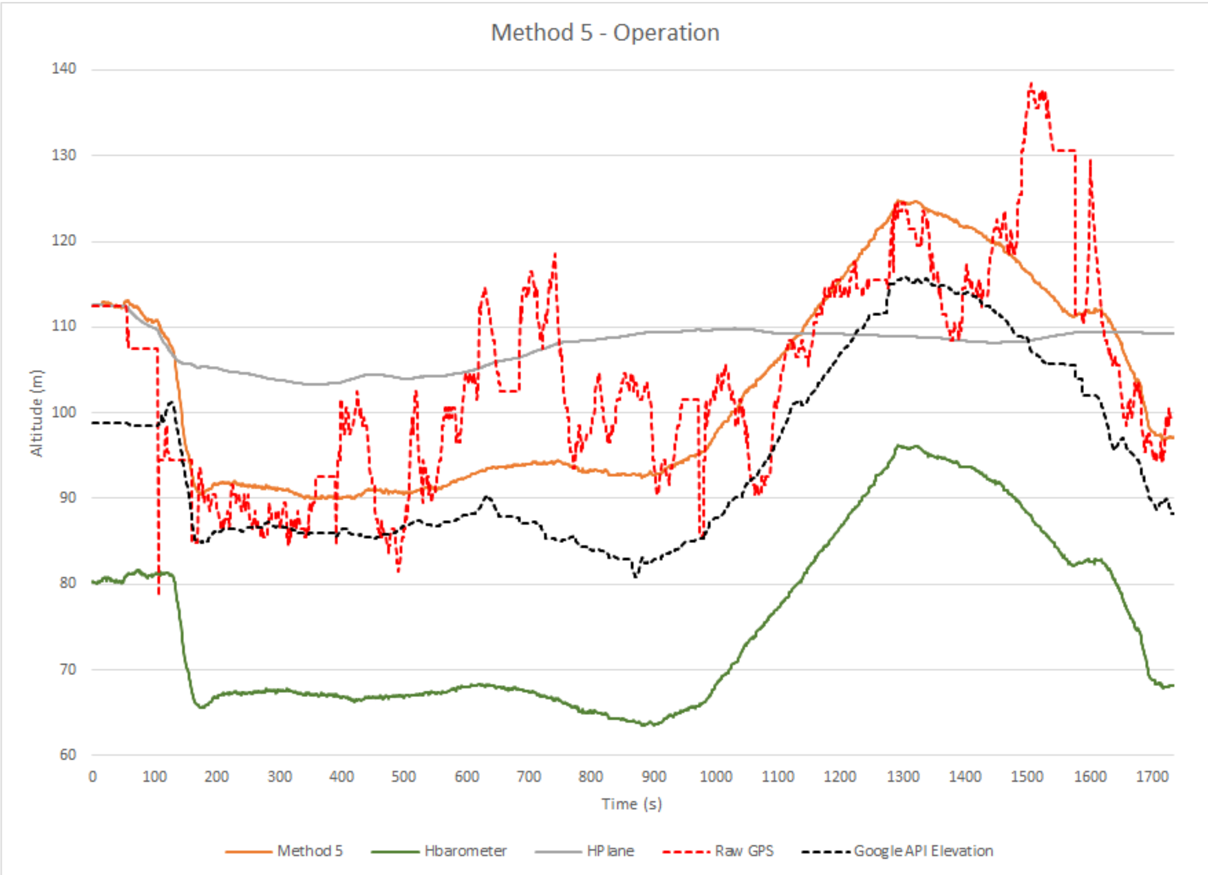


Figure 5.9: Method Operation

Chapter 6

Conclusions and Future work

The main purpose of this work was the study of fusion techniques between the barometric and GPS sensors, and the realization of an Android application that embarks the development of these methodologies. Different approaches have been implemented, designed for different scenarios, which use some weather forecasting data as aid. Techniques have also been explored that reduce the noise of the sensors so that it is possible to obtain better results.

6.1 Achievements

The proposed methodologies, independently of their category, presented better results when compared to standalone results (without any fusion or post-processing) from the GPS measurements. The methodologies that were not based on fusion showed solid results (i.e., small MAE and RMSE when compared to GPS alone) both in outdoor and indoor environments. However, they have a drawback of needing external data (e.g., weather forecast data, Google elevation API) and they need the barometer to be calibrated. Fusion methodologies present similar performance results without having these external dependencies. The created Android application is robust and user-friendly, opening a range of configuration options so that the users can get the altitude in various environments.

6.2 Future Work

At the end of this project, we identified some aspects that need to be improved. In the fusion techniques based on the use of Kalman filters, we observe that the altitude jumps that may occur on a GPS sensor leads the filter to converge slowly when a portion of the initial samples does not have a good accuracy. It would be interesting to study an approach that could make a weighted average of the GPS samples according to the quality of the GPS signal. Moreover, it would be convenient to study what would be the optimal initial measurement error parameter to be considered in the Kalman filter.

Regarding the sliding window-based fusion methods, we need to perform further research to derive what would be the best window size to use and/or to identify and evaluate the trade-offs that need to be taken into account on its choice.

Moreover, a comprehensive and vast set of tests should be performed in order to access with greater confidence performance results, and limitations of the proposed solutions, under distinct well defined test scenarios. In that line, it would be essential to have a trustful altitude ground-truth, both in outdoor in indoor environments.

Bibliography

[1] J. Azevedo and S. Crisóstomo. Weather stations-assisted barometric altimeter for android: Interpolation techniques for improved accuracy. In *2016 IEEE Sensors Applications Symposium (SAS)*, pages 1–6, April 2016. doi:10.1109/SAS.2016.7479890.

[2] Zachary Davies Boren. There are officially more mobile devices than people in the world. Online, October 2014.

[3] Climate Prediction Center. Reading grib files. Online.

[4] Safaa Dawoud. Gnss principles and comparison.

[5] Bureau Internations des Poids et Mesures. Si brochure: The international system of units (si). Online, November 2014.

[6] USGS Science for a changing world. Geoid illustration. Online, September 2018.

[7] gisgeography. Trilateration vs triangulation – how gps receivers work. Online, February 2018.

[8] gpsworld. Almanac | orbit data and resources on active gnss satellites. Online, August 2018.

[9] RG Grant. *Flight: The Complete History of Aviation*. Dorling Kindersley Ltd, 2017.

[10] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, and Elmar Wasle. *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media, 2007.

[11] AR Labs. Accurate altimeter. Online, July 2018.

[12] Leica. Introduction to gps (global positioning system). Technical report.

[13] GSA European Global navigation Satellite Systems Agency. What is gnss? Online, October 2017.

[14] NOAA National Oceanic and Atmospheric Administration. What is the geoid? Online, June 2014.

[15] University of Illinois. Atmospheric pressure. Online, 1999.

[16] Ian Poole. Gps accuracy, errors and precision. Online, September 2018.

[17] Philip J. Robinson. Evangelista torricelli. *The Mathematical Gazette*, 78(481):37–47, 1994. doi:10.2307/3619429.

[18] Matias Tailanian, Santiago Paternain, Rodrigo Rosa, and Rafael Canetti. Design and implementation of sensor data fusion for an autonomous quadrotor. In *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International*, pages 1431–1436. IEEE, 2014.

[19] EXA Tools. Altimeter. Online, February 2018.

[20] Frank van Diggelen and Per Enge. The world's first gps mooc and worldwide laboratory using smartphones. Online, September 2015.

[21] Hao Xia, Xiaogang Wang, Yanyou Qiao, Jun Jian, and Yuanfei Chang. Using multiple barometers to detect the floor location of smart phones with built-in barometric sensors for indoor positioning. *Sensors*, 15(4):7857–7877, 2015. ISSN: 1424-8220. doi:10.3390/s150407857.

[22] Paul A. Zandbergen and Sean J. Barbeau. Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones. *Journal of Navigation*, 64(3):381–399, 2011. doi:10.1017/S0373463311000051.