بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

College of Information Technology and Computer Engineering
Palestine Polytechnic University
Hebron, Palestine

Mobile-Based GPS Land Area measurement System

By:
Cesar Zaitoun

Supervisor:
Dr. Radwan Tahboub

Submitted in partial fulfillment of the requirements of introduction to graduation project in Computer Systems Engineering

October, 2020

Acknowledgment

We would like to express our gratitude to everyone who helped us during the graduation project. To begin with, our supervisor Dr. Radwan Tahboub deserves our endless thanks since he did not save any effort in encouraging us to do a challenging project, providing our group with valuable information furthermore he guided us through the whole time. Thanks for the continuous support and kind communication which had a significant effect that in turn has led us to feel exciting about what we are working on.

Thanks are extended to the “IT & Computer Engineering Society” for the beneficial lectures which are the main resource of this work, besides it facilitated many things in our project.

During the college life, many friends have been a supportive model; we shared many precious and unforgettable moments I have to acknowledge all our colleagues in the research.

Last but not least important, we owe more than thanks to our families members, for their financial support and encouragement throughout our life. Without their support, it is impossible for us to finish our college and graduate.

ACRONYMS

Acronym	Full Version
DGPS	Differential Global Positioning System
Galileo	Global Navigation Satellite System
GLONASS	Global Navigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
mm	Millimeter
NAVSTAR	GPS Satellite System
PPM	Parts Per Million

ABSTRACT

People need new and easy methods to measure areas for different lands using methods that are available at hand rather than buying special equipment like Distomat. By using mobile GPS people can measure areas, but we need to study how we can correct the error that the GPS provide. GPS is used for measurement, but the error is high to be a reliable tool for measurement, our study will be about fixing the error using different ways of error correction such as DGPS then differentiate the error from both GPS and DGPS with Distomat. The error differentiation will give us feedback whether we can use mobile GPS as a measurement tool. We will measure a known area of a land with GPS and DGPS, they will give us a certain error which will lead us to correct it to a certain level, this level makes the decision if we can use mobile as a measurement tool using GPS.

Keywords: GPS, DGPS, Distomat, Area Calculation, Error Correction.

Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1

1 Introduction

This chapter presents an overview of the project included the motivations, objectives, project description and the list of the requirements in the project, expected results, significance, assumptions, and the literature review.

1.1 Overview

In the early days, we always look for the easiest way to do things, we might search on the internet or use any available tools that are available like phones that are used everywhere on daily bases.

The main concern in this project is making a study about different measurements ways using GPS and mobile, and how we can make these methods reliable to use in real life for people, by minimizing the error of each method using different methods like DGPS, post processing and Cellular mobile 3G, which will give us the best value that can be used on daily bases.

The different measurement methods will be compared with other useable methods in real life like Distomat and total station that is used to measure distances and area for a specific land, then the error will be compared with these different methods of measurements, it will identify for us which method will be the best to use.

1.2 Motivation

Our need to make people use available and cheap ways to do measurements for personal use, people prefer to use what is in hand than by a professional tool to measure like Distomat, that is expensive and not easy to carry, using phones will be a lot easier and available everywhere and easy to carry.

Mobile phones will give us a fast way to implement measurements that is needed on regular bases. People prefer to measure a distance or an area on their own, instead of paying for a professional to measure it, since mobile phones are available for everyone it is easy to measure with it.

1.3 Objectives

The project is assumed to achieve the following:

1) Study the different measurement methods that can be used with a mobile such as GPS, DGPS, and cellular. When we go through each method we will have to reduce the error that each method give us, and then compare the error with the different measurement tools.

2) Each method will have its` way in calculation, so we will Design and build a mobile application that will use these methods. This objective will be accomplished by knowing the method that Distomat uses to measure the landscape.

3) The mobile application will use GPS and DGPS to identify the borders of the landscape which give us the area of the land. The actions that the mobile applications will do is give the user the area of the land and its' borders.

1.4 Project Description

Our project will study the possibility to make a mobile application using GPS and minimize the error that the mobile application will give so it can be used for personal use, which will do the same as other measurement methods.

The error will determine whether we can use the mobile application as a measurement tool like Distomat and if the errors show that we cannot use the mobile application, and then we are going to specify the reason why we cannot use it.

The error will either be random error or systematic error which will define the future of the project. Systematic error is consistent, repeatable error associated with faulty equipment or a flawed experiment design, which will give us fixed or proportion if you repeat the experiment.

These errors are usually caused by measuring instruments that are incorrectly calibrated or are used incorrectly**.** Random error has no pattern, the result is unexpected, which make the results unexpected, the same result cannot be replicated by repeating the experiment. [1]

1.5 Literature Review

Most mobile phones uses NAVSTAR (GPS), GLONASS or Galileo for location pinpointing. Both of these systems give us an error when we use them, this error is caused by pseudorange. [2]

The pseudorange is an approximation of the distance between a satellite and a GNSS receiver. GNSS receiver will measure the ranges for at least four satellites, each satellite will give positional data that looks like a circle or orbital data ,thus each position can be calculated for the desired point.

Each satellite has pseudorange which is affected by the ionosphere and troposphere, then the signal is measured by multiplying time for each signal to reach the specified point by the gps device used.

GNSS receiver uses the satellites simultaneously, so all measured ranges has the same error, ranges with the same error are called pseudorange. This way we can calculate the error from all satellite used by the GNSS receiver.

The error can be corrected as certain studies showing methods for estimating it, such as DGPS, post processing GPS and Cellular mobiles 3G.

DGPS (Differential GPS) is essentially a system to provide positional corrections to GPS signals. DGPS uses a fixed, known position to adjust real time GPS signals to eliminate pseudo range errors.

GPS signals coming from satellites down to the ground have to travel through layers of the earth's atmosphere, so they are subjected to delays. [3] This affects the time taken for the signal to travel from any given satellite to a GPS receiver, which introduces slight error into the GPS engine, causing an error in the measured position.

Post processing GPS uses a known location or point on the map then uses it as a reference to calculate the difference between the signal and that known point.[4]

Cellular mobiles 3G uses multi signals that is send to receiver towers by calculating the time each data comes to the receiver and the difference between each time will know the point of the signal that sends that signal and the location.[5]

1.6 Significance

The Significance of this project is: people's need to use their mobile phones as a measurement tool like Distomat; because it is easier to use and available everywhere. People prefer to use what is at hand than buying a professional tool like Distomat, which cost more than using personal phone.

If the mobile application can be used as a measurement method like Distomat [6]. It will help people work through less budget because phones are available to each person and easy to carry and cheaper than buying a Distomat device and its' accessories.

If the error cannot be minimized then the mobile application will determine whether it is a systematic error which is repeatable error associated with faulty equipment or a flawed experiment design; or random error which you can't predict random error and these errors are usually unavoidable. when the error is random error it will give us a feedback for the future that the mobile application cannot be used in the future, but when the error is systematic error then it will prove to us that the technology cannot help us minimize which will fixed in the future of phones when the hardware becomes more accurate.

However, we can compensate if the error is systematic by adding the different methods for error reduction such as DGPS, cellular, or post processing. In addition, we are to identify the systematic error value that the system will provide us. For example if the value is fixed, then we can eliminate it so it can always be a certain value.

1.7 Methodology

Our study will be about using GPS as a measurement tool, but GPS gives us an error that cannot be used as credible tool for area measurement, in that case we will use GPS error correction method such as DGPS.

DGPS is one of error correction methods for GPS, which will correct the pseudo range error that GPS provide us. DGPS uses a hardware component as a reference point called Antenna. This antenna will correct the error of the pseudo range by using differential equation. After that, we will pinpoint the position of the mobile device; DGPS will give us the certain position of the device.

We will use both GPS and DGPS to pinpoint location of the mobile device. Then, each method will give us a certain error, the error will be compared with other measurement tools such as Distomat.

After pinpointing all the points for a specific flat land, which is area is known to us, the mobile application will calculate the area of the land by using polygon area calculation formula.

The formulas for finding the area of polygons are as follows:[14]

$b = \text{base}$

$h = \text{height}$

$a = \text{length of a side}$

$w = \text{width}$

$\text{Triangle} = 1/2 \text{ x b x h}$

$\text{Square} = a2$

$\text{Rectangle} = \text{w x h}$

$\text{Parallelogram} = \text{b x h}$

By calculating the area of the land while using both GPS and DGPS, we will compare these values of the area with Distomat, then we will decide whether we can use GPS or DGPS as a measurement tool, by knowing the difference of error between GPS, DGPS and Distomat, then we will try to correct the error of both methods to a certain level. If this level of error is credible then we can use either GPS or DGPS as an area measurement tool, if not then we cannot use GPS as an area measurement tool.

1.8 Assumptions

- The point that the GPS gives us is not accurate.
- The mobile application needs access to internet connection that will be used to specify the point using GPS.
- It is difficult to use in deserted lands that does not have internet access or communication towers.
- The mobile application will be able to connect to third generation connection or cellular.
- The area that will be measured will be outside.
- The landscape will be flat level land.

Chapter 2

2 Problem statements

In this chapter we will go deeper in the system by talking about problem analysis and solution, list of requirements, and expected results.

2.1 Problem analysis and solution

There is a constant people's need to use their mobile phones to measure areas, than using professional tools like Distomat to measure it.

Our life become so busy and complex in these days, the issue in the project is the error reducing technic that we will be using for the GPS, because the error that the Distomat gives us is an accuracy range of (±2 mm + 1 ppm) to (±10 mm + 10 ppm) constant instrumental error e.g. 5mm represents an accuracy of 1/2000 at 10 m, 1/20000 at 100 m. [7]

Global Positioning system (GPS) is a satellite-based navigation system made up of at least 24 satellites. GPS works in any weather conditions, anywhere in the world, 24 hours a day, with no subscription fees or setup charges. [3] When we use GPS there will be at least four satellites available to pin point your location. Each satellite sends radio signals and their location status with time information, mobile that has GPS receive the signal noting their exact time of arrival and uses these to calculate its distance from each satellite it can see; as shown in figure.1.

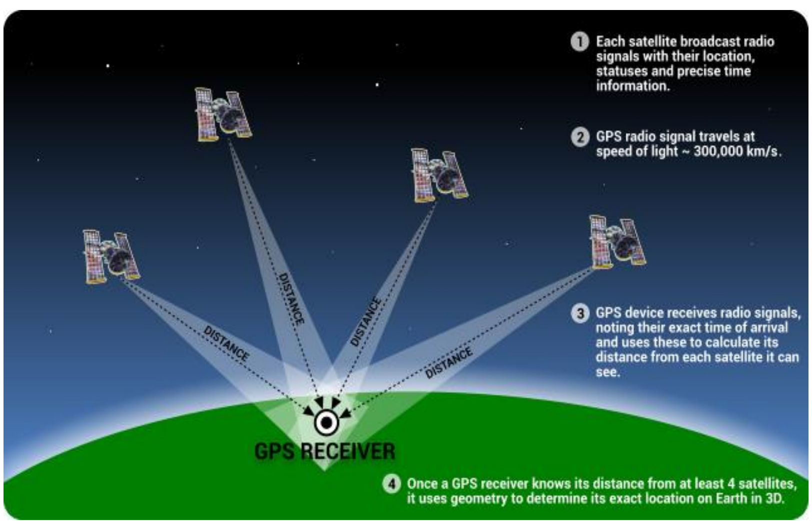

Figure 1 GPS satellite-based navigation system

Which gives us quite a challenge since the GPS gives us an error of about 3 meters [2], and that makes the GPS a un reliable tool for measurement, but there is some ways that will make the error less and the GPS will be useable as measurement tool. There are many ways like using DGPS (differential global positioning system), post processing GPS or using cellular. In our study we will use DGPS as a mechanism for reducing the error.

DGPS is essentially a system to provide positional corrections to GPS signals. DGPS uses a fixed, known position to adjust real time GPS signals to eliminate pseudo range errors.

Pseudo range is the pseudo distance between a satellite and a navigation satellite receiver, for instance Global Positioning System receivers.

The Pseudorange Equation [5]

$$p = \rho + d\&rho + c(dt - dT) + dion + dtrop + \varepsilon mp + \varepsilon p$$

$where$:

$pp\ =\ the\ pseudorange\ measurement$

$\rho\rho\ =\ the\ true\ range$

$d\rho d\rho\ =\ satellite\ orbital\ errors$

$cc\ =\ the\ speed\ of\ light$

$dtdt\ =\ satellite\ clock\ offset\ from\ GPS\ time$

$dTdT\ =\ receiver\ clock\ offset\ from\ GPS\ time$

$diondion\ =\ ionospheric\ delay$

$dtropdtrop\ =\ tropospheric\ delay$

$\varepsilon mp\varepsilon mp\ =\ multipath$

$\varepsilon p\varepsilon p\ =\ receiver\ noise$

Clock offsets are only one of the errors in pseudo ranges. Please note that the pseudo range, p, and the true range, ρ, cannot be made equivalent, without consideration of clock offsets, atmospheric effects and other biases that are inevitably present.

Some of those errors are shown in the previous equation. The pseudorange measurement equals rho, the true range, plus all of these other factors. If it were correct initially, if the pseudo range was good and complete by it, these other factors would not need to be considered. [14] The pseudo range, p, would be equal to the true range, rho.

But, in fact, there are satellite orbital errors because the orbits of the satellites are affected by many factors including the variations in gravity, drag, and tidal forces from the sun, the moon, etc. The speed of light is constant, but it has to be multiplied times the satellite clock offset from GPS time minus the receiver clock offset from GPS time, because of propagation delay and imperfect oscillators among other things.

While the satellite clock offset from GPS time is somewhat known from the Navigation Message, it's not perfectly known. The receiver clock offset from GPS time is not well known at all. Then, there is the ionosphere delay to be considered. There's also a bit of tropospheric delay. The troposphere includes the atmospheric layers below the ionosphere. [8]

Then, on top of that, there's something known as multipath. Multipath means that the signal from the GPS satellite bounces off of something before it gets into the receiver. Obviously, if you're measuring distances to determine your position, that bounce creates a difficulty. There is receiver noise as a bias in the system. [9]

In short, all these biases contaminate the pseudo range position. So, while the pseudo range is a good way to get started with a GPS position, it isn't the full answer.

DGPS will give us the exact spot of a certain device that is in the area of the antenna that we will use to take the signal from the satellite ,the error in DGPS is 3 cm[3] which is much less than the normal GPS, the DGPS will need an antenna that will receive the signal from the satellites and the antenna will determine the location of the device or mobile, differential global positioning system will use a differential equation to calculate the distance between the antenna and the mobile device as shown in figure 2.

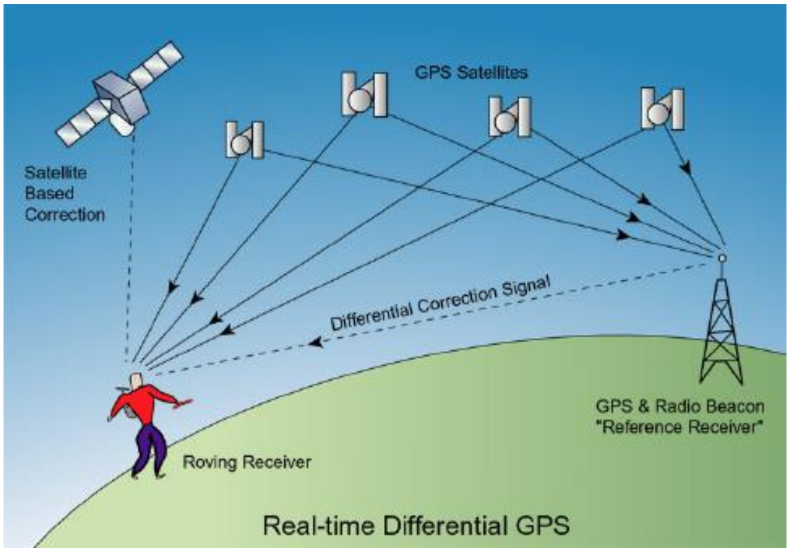

Figure 2: Real time differential GPS. This figure was taken from https://quizlet.com/462370819/inertial-sense-flash-cards/

But, there is still a problem that will encounter for measuring the area of an unstable level land like mountains or roof tops.

DGPS will give us the angle of the mobile device which will determine the altitude of that mobile device as the differential equation will be demonstrated.

2.2 List of Requirements

We can summarize our project requirements as:

- Mobile application which reads GPS data.
- Compute the area of flat polygon land.
- Compute area using standard area measurement devices.
- Make correction factor to the mobile area.
- The mobile application will use Google maps to find the coordinates of each point.

2.3 Expected Results

We expect to accomplish the following at the end of the project:

1. A mobile application that can calculate the area of a specific land with the least error we can achieve.

2. Minimizing the error of GPS using DGPS that will give us a reliable point that will help us measure the area and the altitude.

2.4 Methodology

In this section we will explain how the requirements will be filled. At first we will build the mobile application using android studio software, the mobile will be of high end types which uses the gps sensor receiver to connect with the gps data.

The mobile must have gps sensor receiver to give us the data we need from the gps, the data that will be taken is location data to give us the coordinate in latitude and the longitude for each point the user specifies ,then after choosing this points the application will have a function that draws the shape of the land after taking all the wanted points.

After that the mobile application will also has a function that computes the area in square meters which will be shown on the main screen of the application, all the location information's will be taken from Google maps by connecting the application with it.

Then we will use the stonex DGPS tool to differentiate the difference in area calculations, and also to differentiate the error in coordinates of latitude and longitude points taken by the mobile application.

After differentiating the error we will try to find the best way to correct the error in the mobile application, if the error is systematic we can use the application as a measurement tool.

Chapter 3

3 Background

3.1 Overview

This chapter introduces the theoretical background of the project; description of Distomat mobile application system component, and some description of hardware and software components used in the system. Finally, some descriptions of design specifications and constraints.

3.2 Theoretical Background

There is always an idea behind things that will take us to the next level, the idea comes to fulfill a specific need or open a new idea that will give us a better development for things in our life that will make it easier to do.

The idea is that can we use smartphones to measure areas, that can be done in many ways, but smartphones don't have that hardware or ability to do so, instead we can use an external hardware that will make things easier and more reliable, so we chose to use GPS which is available in every smartphone, but GPS has high error rates that cannot be used in measurements of distances. [10]

Fortunately, there is a way to reduce that error that GPS give us and make the measurements more accurate by using DGPS.

DGPS ,which stands for Differential Global Positioning System, it is a system that uses receivers on the ground that receives the signals from satellites and then determine the signal from the smartphone ,then it will use a differential equation that will calculate the place of that signal ,then pin point it on the map to show it exact place .[3]

With DGPS we can use smartphones to calculate areas by pin pointing the position of the phone then calculate the area and the difference between each point ,which will give us a map for the land and its borders .

Even so, we cannot take smartphones that uses DGPS as a reliable tool, because the Distomat is more reliable, but smartphones can be used for personal usage. After all, the idea opens other ideas that might let us use the smartphone as reliable measurement tool for areas that can replace Distomat in the future.

3.3 Hardware Component of the System

3.3.1 Stonex Antenna (receiver station)

Each DGPS uses a network of fixed ground-based reference stations (as shown in figure 3) to broadcast the difference between the positions indicated by the GPS satellite systems and known fixed positions. These stations broadcast the difference between the measured satellite pseudo ranges and actual (internally computed) pseudo ranges, and receiver stations may correct their pseudo ranges by the same amount.[15]

The digital correction signal is typically broadcast locally over ground-based transmitters of shorter range as shown in figure 4.

Figure 3: Antenna for DGPS

Smartphone

A smartphone is a mobile phone with highly advanced features. A typical smartphone has a high-resolution touch screen display, Wi-Fi connectivity, Web browsing capabilities, and the ability to accept sophisticated applications. The majority of these devices run on any of these popular mobile operating systems: Android, iOS, OS and Windows Mobile. [13]

A smartphone is expected to have a more powerful CPU, more storage space, more RAM, greater connectivity options and larger screen than a regular cell phone.

In addition to the features mentioned earlier, smartphones are also equipped with innovative sensors like accelerometers or even gyroscopes. Accelerometers are responsible for displaying screens in portrait and landscape mode, while gyroscopes make it possible for games to support motion-based navigation.

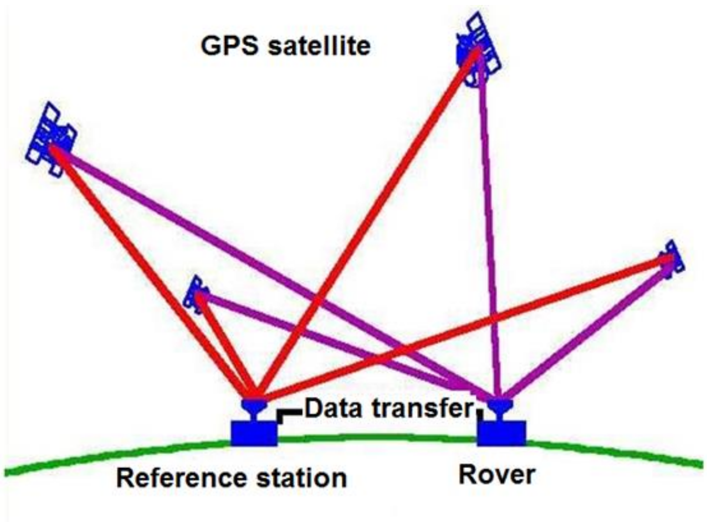

Figure 4: GPS Data Transfer

3.3.2 GPS receiver sensor

GPS sensor which is found in every high end phones, will be used to take the location data to specify the coordinates of each point taken, which is connected with GNSS satellites, that will search for the location of the phone to find where the point is located.[12]

3.4 Software Component of the System

3.4.1 Android Programming Language

Android is a mobile operating based on Linux operating system for mobile touchscreen devices such as tablets or smartphones. Android software source code is an open source. [13] Applications are usually built in the java programming language using an Android software development kit (SDK). Android gives you a rich application framework which allows you to be creative in building applications and games, so it's the best choice to build a mobile application related to the system.

3.5 DGPS (Differential Global Positioning System)

DGPS (Differential GPS) is essentially a system to provide positional corrections to GPS signals. DGPS uses a fixed, known position to adjust real time GPS signals to eliminate pseudo range errors.

GPS signals coming from satellites down to the ground have to travel through layers of the earth's atmosphere, so they are subjected to delays. This affects the time taken for the signal to travel from any given satellite to a GPS receiver, which introduces slight error into the GPS engine, causing an error in the measured position.

The signals have to travel through the ionosphere, which is the outer edge of the atmosphere. This bit of the atmosphere is hit by solar radiation, which causes particles to split up and become positively charged. This layer of the atmosphere has the largest impact on electromagnetic signals passing through, which of course includes the radio signals coming down from satellites .[3]

The ionosphere will add a relatively large delay; the actual number depends on receiver location, satellite location, time of day, solar flare activity, etc. A lot of those factors are estimated and allowed for, which then corresponds to a delay of up to 16 ns for the signal passing through, which is changing all the time. This can introduce up to a 5 m error to the captured position.

The second layer the GPS signals travel through is the troposphere. This is the 'weather' section of the atmosphere, so includes conditions such as clouds, rain and lightning. This adds a much smaller delay to the signal of up to 1.5 ns, which can introduce up to a 0.5 m positional error.

These delays are random delays which fluctuate. As such, there is no way to precisely measure what they will be at any given time. Each delay is also specific to every individual satellite, as they are positioned at different areas around the world, so their signals will be subjected to different atmospheric conditions.

A static base station can be used to provide correction messages to signal delays. This is done by setting the base station up in a set point on the ground, and then working out its exact position on the earth's surface. This is done by leaving it recording GPS data for as long as possible. Over the time that the base station is capturing data, the ionosphere and troposphere change, causing the delays in the signals to change, randomly. Because the delays are subject to random changes, they can be averaged out.

3.6 Design Specification and constrains

The mobile application will receive the signal from the antenna that provides the signal from the GPS that will give us the location of the smartphone or the person, that will pin a point on the application so that it can connect all points together to calculate the area of the land, then draw the shape that has those points connected to draw the area that we took the coordinates from.

Some constraints in the shipping of the equipment's if we buy them online, or in the cost if we buy them locally since it's very expensive.

Chapter 4.

4 System Design

4.1 Overview

This chapter discusses the design option of hardware and software, and conceptual design of the system such as block diagram of the hardware component of the system.

4.2 Brief Description of the System

Mobile GPS is an application that measures land areas that are flat. By pin pointing each point of the land and join them together to give us the land architecture. Then it will give us the area of the land and its` parameter.

To study this system as a reliable one we will calculate the error that it gives us, and then compare that error with other calculation systems and tools like Distomat and cellular.

4.3 Design Options

4.3.1 Hardware Design

Our hardware consists of the stonex antenna that is used as a receiver station that will use DGPS to pin point the location of the person.

Also the gps sensor receiver which can measure atmospheric pressure, data measured by it is used to determine how high the device is which turn results to location data.

4.3.2 Software Design

We use android software to design the mobile application that will calculate the area of the land and draw the architecture of it.

Android Software: We use the Android Studio program; it is an open source and provides the fastest tools for building applications on every type of Android device. Similar to smartphones, Android applications are usually developed using java through the Android Software Development Kit. In this project, we will use it to build an application related to our system to communicate and serve it.

4.4 Detailed Design

4.4.1 System design

The system is designed using android studio, the mobile application will consists of three main functions to fulfill our requirements to measure any area that is given by the user.

The three functions are add marker, draw lines and calculate area.

- Add marker

Add marker is the function that will pin point each point the user chooses by standing on that phone then pressing the button of add marker the on MapReady (GoogleMap googleMap)the application will use the gps sensor receiver to take the point coordinates and pin point it on the map.

- Draw lines

This function will draw the shape of the land after pin pointing the points that the user choose, the shape will be drawn based on the order the user chose to pin point the points from the first step to the last one, by using void draw Lines (),this will be achieved.

- Calculate area

Using the method computeSignedArea (List<LatLng> path, double radius) will take all the point coordinates and put it in a list of latitude and longitude to add it to the main function for calculating.

SphericalUtil.computeArea (latLngs)[11]this method will take the coordinates as parameters then calculate the area that those points cover.

4.4.2 System Diagrams

Figure 5 shows the block diagram of the system, and how hardware components communicate with each other in Figure 6.

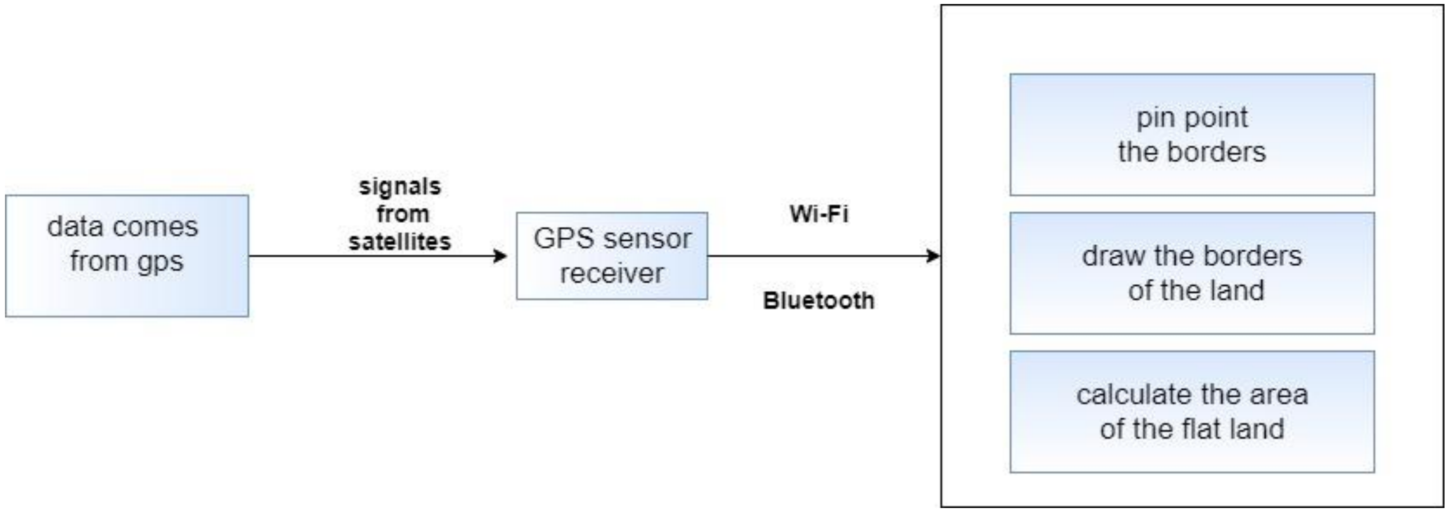

Figure 5: System Block Diagram

The block diagram in Figure 5 shows how the data that is taken from the GPS can be used throw suitable way (Wired or Bluetooth), after processing the data and depending on the result will do the three actions which are pin pointing, then draw borders, finally calculate the area of the flat land.

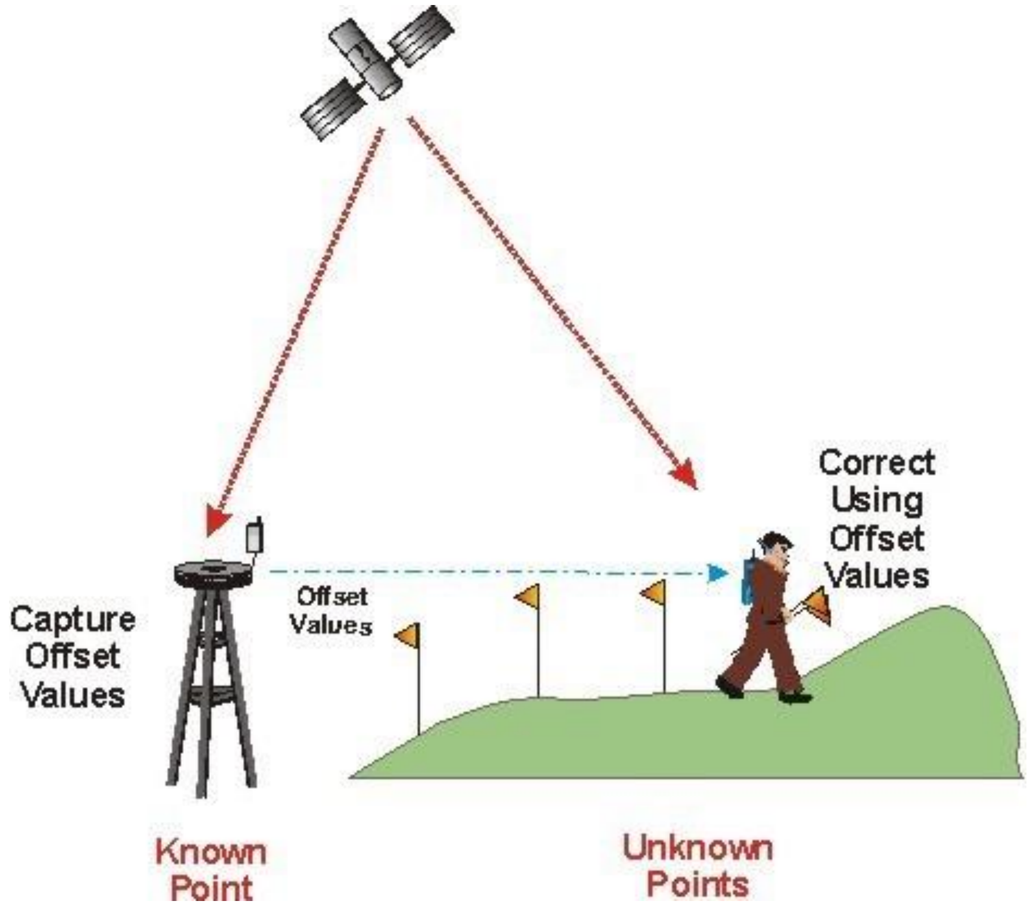

Figure 6: DGPS System Diagram

The diagram in Figure 6 shows how hardware component is connected with each other in the system; to locate the points.

4.4.3 System Flowchart

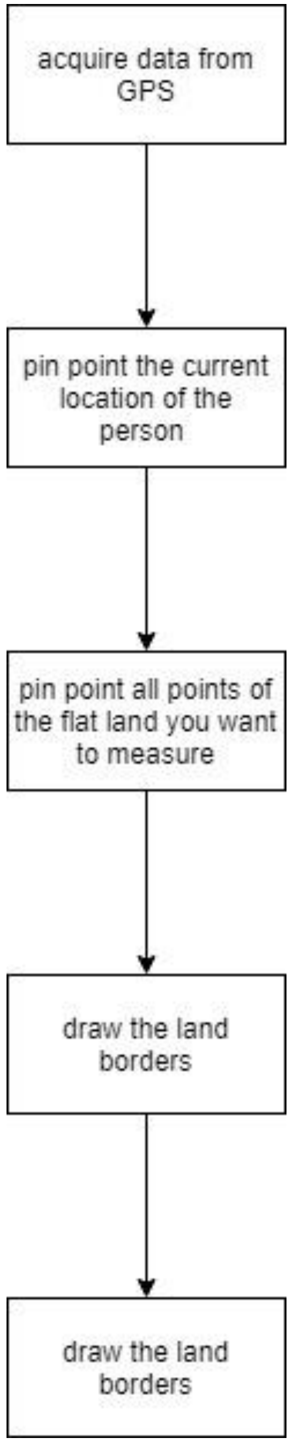

Figure 7: System Flowchart

5 implementation

In this chapter, an explanation about how our hardware and software works together and how they are connected.

5.1 Software implementation

The implementation will include what our mobile application will use to calculate any flat area, we will import the location services in android studio to take the points coordinates each point will have the step number and latitude longitude numbers.

By calling Google maps the main activity will be the Google maps interface to see how the user react with it by adding markers where he stands.

After pin pointing all points the draw lines function will connect the markers together as the user ordered the steps, by calling Polyline Options so the application can draw any shape that is polygon.

In the end, the user will calculate the area of the points he chose using calculate Area to show the user the area in meter square which will be showed as toast.

5.2 Hardware implementation

Most basic hardware that we used in this project is the gps sensor receiver, this sensor made it possible for us to make the application works. by understanding how the sensor works we made the error less than expected, the sensor depends at how the phone stands and in which direction knowing this fact when we used the application the mobile device while taking each point was facing north or the same direction for the first point. this is done to make the shape of the land as right as possible.

Also the sensor takes time to verify the point of the user so the application refresh the location services info each second to make it more accurate, that is done by standing more than two seconds at the point then pin pointing that point, however the user must let the mobile device above the point he wants to take, because the sensor is connected to the phone, if the user stands above the point that will cause an increase in the error.

We used also the stonex antenna to estimate the actual area of each land we tested; to give us the longitude and latitude of each point; so we can compare it with the mobile readings. The antenna is connected with gnss which is the same satellites used by the gps sensor receiver.

6 Testing and correction

6.1 Introduction

This chapter will describe how we used the application to show us the calculation of area to three specific lands that has been measured with DGPS then with the mobile application.

We will show what kind of error that the mobile application give us, whether it is systematic or random error; to see the possible correction methods based on the error that is taken in consideration.

The outcomes of the tests will decide if the mobile application is credible to be used as a measurement tool for areas, like other famous measurement tools.

6.2 Testing

This section will show the lands that we choose to measure by DGPS and mobile application, then compare it with each other to see how much error the mobile application give us, if we can correct it or not.

The three lands that will be tested are at the Palestine polytechnic university inner land; the three lands were chosen for the flat structure.

6.2.1 DGPS testing

The DGPS that we will use is stonex software and stonex antenna ,to measure the area of the specified lands and to pin point the coordinates for each point taken to each land, the coordinates will be taken for its' latitude and longitude.

Each land will have and area view in the stonex device, that will show us the borders and each point location, also the first point taken and the others in order.

Table 1 DGPS first land

point	latitude	longitude	area
1	31.3022.64	35.0525.25	
2	31.3022.81	35.0524.66	
3	31.3026.64	35.0552.48	
4	31.3023.38	35.0525.56	389.4

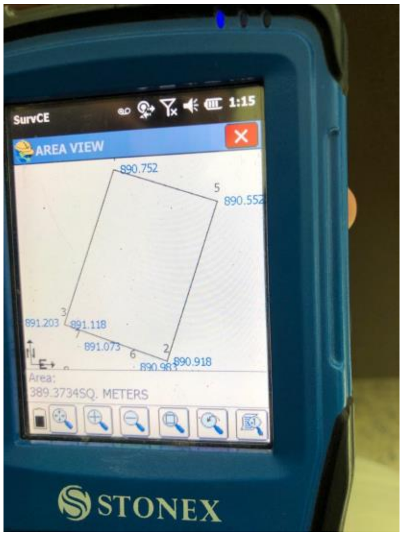

Figure 8 DGPS first land drawing

Table 2 DGPS second land

point	latitude	longitude	area
1	31.3022.61	35.0525.06	
2	31.3022.71	35.0524.74	
3	31.3022.53	35.0524.66	
4	31.3022.43	35.0524.99	54.3

Figure9 DGPS second land drawing

Table 3 DGPS third land

point	latitude	longitude	area
1	31.3022.82	35.0524.38	
2	31.3022.51	35.0525.41	
3	31.3021.36	35.0524.93	540.21

Figure10 DGPS third land drawing

6.2.2 Mobile application testing

Now we will use the mobile application that has been programed by android studio to calculate the area of a specified land.

We developed two applications to measure the area of lands, each mobile application has the same purpose, which is to measure the area of specific land, but each application uses different method for pin pointing the points that are taken.

The first application uses the eastern and northern coordinates to pinpoint each point, which is used in the known measurement tools; un fortunately after testing the first application, the application calculation of area did not give understandable measurements; the calculation was wrong in measurement, which cannot be taken as a number of area calculation.

The second application which will be used in the comparing measurement with the DGPS testing, uses the latitude and the longitude coordinates to pin point the points taken for each land.

Each land were measured according to the order of the DGPS coordinates, every point was taken by putting the mobile device above the desired point not the body of the user, to minimize the effects of error that is caused by the signals of the users body, also the mobile device while taking each point was facing the same direction, in our testing the mobile was facing the north direction while taking each point, to minimize the contradiction of direction.

Every land will be measured ten times for measuring the average of error that will occur from the GPS, so we can find the best way for error correction.

After pinpointing all the points, the mobile applications draw the borders of the land that has been taken, then we calculate the area using the app function “calculate area” that will show us the area of the drawn land.

Table 4 mobile app first land

point	latitude	longitude	area
1	31.3064.95	35.0504.35	
2	31.3065.20	35.0502.98	
3	31.3062.71	35.0502.14	
4	31.3060.64	35.0503.24	329.4
1	31.3065.02	35.0504.116	
2	31.3005.08	35.0502.75	
3	31.3063.19	35.0501.90	
4	31.3062.87	35.0503.66	350.8
1	31.3064.89	35.0504.60	
2	31.3065.33	35.0503.24	
3	31.3063.61	35.0502.37	
4	31.3063.36	35.0503.43	245.9
1	31.3064.74	35.0505.23	
2	31.3065.29	35.0503.55	
3	31.3063.40	35.0502.30	
4	31.3063.11	35.0503.85	358.1
1	31.3064.56	35.0505.38	
2	31.3065.57	35.0503.90	
3	31.3063.60	35.0502.50	
4	31.3062.89	35.0503.97	414.8
1	31.3064.64	35.0505.65	
2	31.3065.36	35.0503.55	
3	31.3063.76	35.0502.56	
4	31.3063.00	35.0503.93	402.8
1	31.3065.00	35.0504.73	
2	31.3065.30	35.0503.05	
3	31.3063.40	35.0502.10	
4	31.3062.98	35.0503.60	367.9
1	31.3065.25	35.0504.35	
2	31.3065.56	35.0502.94	
3	31.3063.95	35.0502.30	
4	31.3063.34	35.0502.50	279.34

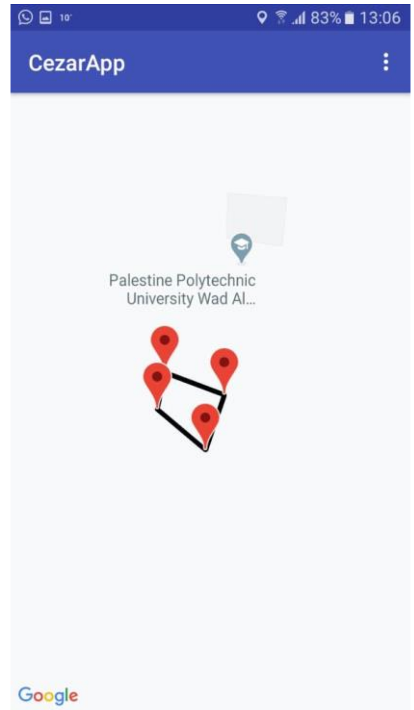

Figure 11 mobile app drawing first land

1	31.3064.96	35.0505.15	
2	31.3065.42	35.0503.80	
3	31.3063.67	35.0502.75	
4	31.3062.85	35.0503.43	300.08
1	31.3064.64	35.0505.65	
2	31.3065.36	35.0503.55	
3	31.3063.76	35.0502.56	
4	31.3063.00	35.0503.93	310.14

Table 5 mobile app second land

point	**latitude**	**longitude**	**area**
1	31.3062.96	35.0503.17	
2	31.3063.53	35.0503.43	
3	31.3063.57	35.0502.37	
4	31.3063.25	35.0501.90	61.1
1	31.3062.98	35.0503.10	
2	31.3063.52	35.0503.40	
3	31.3063.77	35.0503.40	
4	31.3062.30	35.0501.90	57.8
1	31.3062.83	35.0502.60	
2	31.3063.32	35.0501.57	
3	31.3062.98	35.0501.68	
4	31.3062.56	35.0502.63	28.8
1	31.3061.72	35.0502.50	
2	31.3062.27	35.0500.88	
3	31.3061.59	35.0501.83	
4	31.3062.89	35.0503.05	5.5
1	31.3063.10	35.0502.52	
2	31.3006.24	35.0501.45	
3	31.3006.35	35.0501.10	
4	31.3062.54	35.0503.24	135.5
1	31.3062.60	35.0502.56	
2	31.3062.56	35.0501.76	

3	31.3062.43	35.0501.57	
4	31.3062.50	35.0502.98	17.4
1	31.3062.45	35.0502.56	
2	31.3062.90	35.0501.34	
3	31.3062.50	35.0501.34	
4	31.3062.50	35.0502.56	22.03
1	31.3062.03	35.0501.90	
2	31.3062.31	35.0501.87	
3	31.3062.16	35.0502.20	
4	31.3061.97	35.0502.63	10.8
1	31.3062.89	35.0502.90	
2	31.3062.98	35.0501.76	
3	31.3062.94	35.0501.68	
4	31.3062.79	35.0502.18	13.1
1	31.3062.89	35.0502.20	
2	31.3062.68	35.0501.45	
3	31.3062.70	35.0501.18	
4	31.3062.62	35.0502.86	15.45

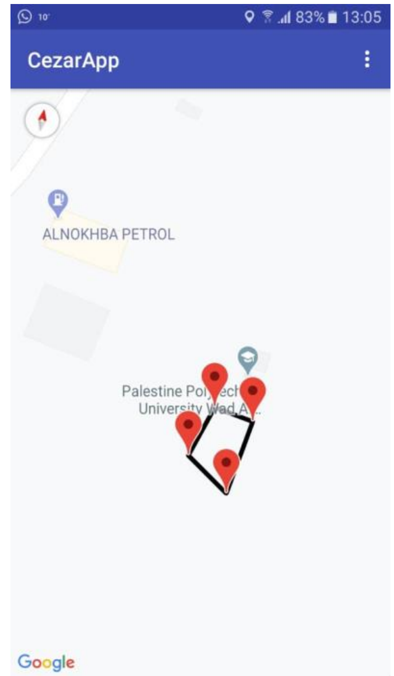

Figure 12 mobile app drawing second land

Table 6 mobile app third land

point	**latitude**	**longitude**	**area**
1	31.3062.77	35.0502.67	
2	31.3061.19	35.0504.50	
3	31.3058.92	35.0502.67	527.39
1	31.3059.03	35.0502.37	
2	31.3062.94	35.0503.74	
3	31.3063.94	35.0500.90	620.4
1	31.3059.99	35.0502.60	
2	31.3062.20	35.0503.43	
3	31.3063.17	35.0500.57	526.2
1	31.3059.72	35.0502.06	
2	31.3062.62	35.0503.63	
3	31.3063.74	35.0501.00	494.9

1	31.3060.60	35.0502.02	
2	31.3062.12	35.0503.66	
3	31.3063.11	35.0500.76	318.9
1	31.3061.15	35.0502.56	
2	31.3062.45	35.0503.78	
3	31.3063.38	35.0501.22	234.8
1	31.0359.05	35.0502.52	
2	31.0362.43	35.0504.00	
3	31.0363.36	35.0561.00	609.5
1	31.0359.76	35.0502.86	
2	31.0362.79	35.0503.74	
3	31.0363.30	35.0500.80	493.3
1	31.0393.60	35.0502.60	
2	31.0362.24	35.0504.35	
3	31.0363.32	35.0500.57	673
1	31.0359.19	35.0502.30	
2	31.0362.79	35.0503.47	
3	31.0363.38	35.0501.55	508.9

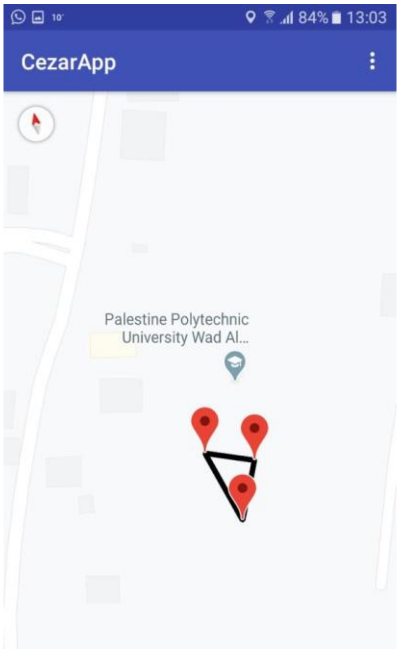

Figure 13 mobile app drawing third land

After finding the coordinates we can see that the area measurement is not specified; because each reading gives us different coordinates and area measurement.

6.3 Correction

We will make different ways of correction to find whether we can find the error type if it is either systematic or random; if the result where systematic then it can be fixed and we can use the application to measure area.

However, if the error is random we cannot use the application to measure area .

By calculating the average and standard deviation of the taken areas we can determine what kind of error we have which will show us the next step.

Table 7 mobile app first land average

	area
	389.4
	329.4
	350.8
	245.9
	358.1
	414.8
	402.8
	367.9
	279.34
	300.08
	310.14
AVERAGE	340.7873
standard deviation	50.74635

Table 8 mobile app second land average

	AREA
	61.1
	57.8
	28.8
	5.5
	135.5
	17.4
	22.03
	10.8
	13.1
	15.45
AVERAGE	36.748
STANDARD DEVIATION	37.55645

Table 9 mobile app third land average

	AREA
	527.39
	620.4
	526.2
	494.9
	318.9
	234.8
	609.5
	493.3
	673
	508.9
AVERAGE	500.729
STANDARD DEVIATION	126.9351

As we can see from the average of each land, the average does not calculate the same area for the actual calculation for the first land the actual area is 389.4 square meter and the average is 340.78 square meter.

As for the second land which the area is small the actual area is 54.03 square meter and average of the mobile application is 36.7 square meter. The third land actual area is 540.21 square meter as for the average of the app is 500.729 square meter.

As we can notice the difference in average for the actual area is about 40 square meters in the first and the third land where both have larger areas, but for the second land which has much smaller area than the other two the difference between the actual and the application average is approximately 4 square meters.

This difference in averages will make it difficult to correct because it give us a random measurements for the same area.

As for the standard deviation which is a quantity expressing by how much the members of a group differ from the mean value for the group, in the second the area the standard deviation is higher than the average which means that the data is more spread out of the average, however in the first and third land which have larger areas the standard deviation is less than the average which indicates more of the data is clustered about the average.

6.4 Correction methods

We will try two methods for correction the error based on our results of average and standard deviation. The first method will be point's average, the second method will line points.

6.4.1 Point's average:

Figure 14 ten samples for the same point

In this method we will take many samples for the same point and try to find the average of the points, by using the app each point will have ten samples to find whether we can find the approximate point to the actual point if possible.

To minimize the error we will try to take many points for the same pin point and find the average of the points circle around.

Using the app we will take 10 points for the same location to know the behavior of the gps receiver sensor if the Average is near the Actual point we can use it to make a more credible app reading .

After testing the results it is not possible to find the Average of the points; because if we take 10 samples for the same point the area will cover 80 square meters, when we took more than ten then the area will increase, due to the behavior of the location services.

The gps take each point as a circle and each point must be in that circle however ,The other points has another circle and each circle has a perimeter different from the original or first circle which make it nearly unexpected , as shown in figure 14, behavior to take the average of each circle. If the points Are around the perimeters of the first circle that would be easier to take the average and fix the error.

6.4.2 Line points:

In this method, we will take many points for the same line the connects each point together, after taking the points for all the lines we will calculate the average of the land.

The user will add marker after each step he walks toward the other point approximately about three points in each line, then calculate the average for the land, for more specific distance calculation, as shown in figure 15.

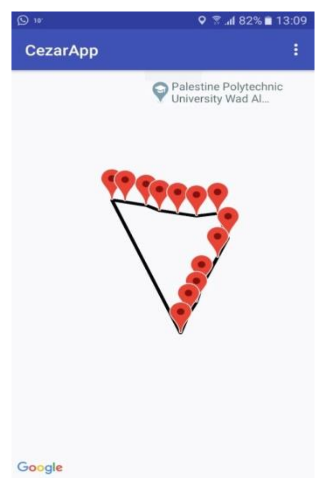

Figure 15 app drawing of points for the same line

After testing line points calculation, we can notice that the points does not give us a straight line, because the each point takes the shape of a circle, and the marker is in that circle. In addition, the area is different from the first tests because it gives us different area calculations, but the area calculations are near the other test which is the same error before.

6.5 Results:

As we can conclude the error is random and cannot be fixed, which makes it difficult to use the mobile as a credible measurement tool.

Even after trying to minimize the error with different methods it was not possible to use for area measurements, however we can conclude that there is some error that can vary between measurements which can be taken to give a slightly approximate result for the area.

We can use the mobile application to give us an example for the area of land approximately; not specifically right area measurement.

7 Summary

In this chapter we will discuss the report summary, conclusion and next semester work for the project.

7.1 Summary

The need for area measurement tools that are available and easy to use for the average user, makes mobile is a needed method for measurement. We can use mobile for measurement by using Global Positioning System, but GPS give us and error which cannot be used as a reliable tool.

If we can correct that error using GPS error correction methods, we can use GPS as a reliable measurement tool, error correction methods such as DGPS can give us certain level of error, which is less than the actual GPS.

Our study will be about error correction for GPS using GPS and DGPS, then differentiate the error with other area measurement tool such as Distomat, which will specify whether we can use GPS as measurement tool.

Then we can decide whether we can use the mobile application as a measurement tool for area, but based on our giving results we can use it to measure an approximate area for large lands, lands that are larger than 200 meter square.

.

7.2 Conclusion

GPS cannot be used as a measurement tool by correcting the error. But, by using GPS error correction methods, we can make GPS using mobile as approximate tool for measurement.

If the hardware of the gps sensor developed in the future their might be a chance to use the mobile as credible tool for area measurement.

8 Bibliography

[1] https://www.thoughtco.com/random-vs-systematic-error-4175358

[2] https://books.google.ps/books?id=PqpkAwAAQBAJ&printsec=frontcover&hl=ar#v=onepage&q&f=false

[3] https://racelogic.support/01VBOX_Automotive/01General_Information/Knowledge_Base/How_Does_DGPS_(Differential_GPS)_Work%3F

[4] https://novatel.com/an-introduction-to-gnss/chapter-5-resolving-errors/gnss-data-post-processing

[5] https://www.signalbooster.com/pages/do-cell-phone-signal-boosters-work

[6] https://careertrend.com/list-7168513-electronic-surveying-instruments.html

[7] https://cdn.ymaws.com/www.nysapls.org/resource/resmgr/2015_Conference_Handouts/012115-06_Errors_Paiva.pdf

[8] https://niwa.co.nz/education-and-training/schools/students/layers

[9] http://gpsinformation.net/multipath.htm

[10] https://www.tandfonline.com/doi/full/10.1080/13658816.2015.1086924

[11] https://googlemaps.github.io/android-maps-utils/javadoc/com/google/maps/android/SphericalUtil.html

[12] https://www.campbellsci.com/gps

[13] https://searchmobilecomputing.techtarget.com/definition/smartphone

[14] https://study.com/academy/lesson/how-to-find-the-area-of-irregular-polygons.html

[15] https://www.stonex.it

Appendix

Main activity:

```
package wecsee.caddx.cezarapp;

import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.CameraPosition;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.PolylineOptions;

import java.util.ArrayList;
import java.util.List;

import com.google.maps.android.SphericalUtil;

import static com.google.maps.android.SphericalUtil.computeSignedArea;
import static java.lang.Math.abs;

public class MainActivity extends AppCompatActivity implements OnMapReadyCallback {

    LocationManager locationManager;
    LocationListener ls;
    private GoogleMap mMap;

    float globalLat = 0, globalLng = 0;
    ArrayList<LatLng> points = new ArrayList<>();

    @RequiresApi(api = Build.VERSION_CODES.M)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        locationManager = (LocationManager)
                getSystemService(Context.LOCATION_SERVICE);
```

```
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);

        ls = new LocationListener() {
            @Override
            public void onLocationChanged(Location location) {
                Log.d("Location", location.toString());
                globalLat = (float) location.getLatitude();
                globalLng = (float) location.getLongitude();
            }

            @Override
            public void onStatusChanged(String provider, int status, Bundle extras) {

            }

            @Override
            public void onProviderEnabled(String provider) {
                Toast.makeText(MainActivity.this, "Enableb", Toast.LENGTH_LONG).show();
            }

            @Override
            public void onProviderDisabled(String provider) {
                Toast.makeText(MainActivity.this, "Disabled", Toast.LENGTH_LONG).show();
            }
        };

        if (checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
checkSelfPermission(Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(MainActivity.this,
                    new String[]{Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.ACCESS_COARSE_LOCATION},
                    1);
            return;
        }
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 0, ls);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        LatLng palestine = new LatLng(31.332570, 35.059827);
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(palestine, 17));
        mMap.animateCamera(CameraUpdateFactory.zoomIn());
        mMap.animateCamera(CameraUpdateFactory.zoomTo(17), 2000, null);

//        points.add(new LatLng(0, 0));
//        mMap.addMarker(new MarkerOptions().position(new LatLng(0, 0)).title("Tessst" + points.size()));
//        points.add(new LatLng(5, 10));
//        mMap.addMarker(new MarkerOptions().position(new LatLng(5, 10)).title("Tessst" + points.size()));
//        points.add(new LatLng(30, 14));
//        mMap.addMarker(new MarkerOptions().position(new LatLng(30, 14)).title("Tessst" + points.size()));
    }

    void drawLines() {
        if (points.size() == 0) {
            return;
        }
        PolylineOptions polylineOptions = new PolylineOptions();
        for (int i = 0; i < points.size(); i++) {
            polylineOptions.add(points.get(i));
        }
        polylineOptions.add(points.get(0));
        mMap.addPolyline(polylineOptions);
    }
```

```
    void calculateArea() {
        if (points.size() == 0) {
            return;
        }
        List<LatLng> latLngs = new ArrayList<>();
        for (int i = 0; i < points.size(); ++i)
            latLngs.add(points.get(i));
        latLngs.add(points.get(0));

        double result = SphericalUtil.computeArea(latLngs) ;

        Toast.makeText(getApplicationContext(), "area=" + result, Toast.LENGTH_LONG).show();
//        Toast.makeText(getApplicationContext(), "area=" + computeSignedArea(latLngs), Toast.LENGTH_LONG).show();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main_menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.btnAddMarker:
                if (globalLat == 0 || globalLng == 0) {
                    Toast.makeText(getApplicationContext(), "Doesn't have new readings yet", Toast.LENGTH_SHORT).show();
                    break;
                }
                points.add(new LatLng(globalLat, globalLng));
                mMap.addMarker(new MarkerOptions().position(new LatLng(globalLat, globalLng)).title(
                        "Step number " + points.size() + " > " +
                                globalLat + " " + globalLng
                ));
//                mMap.addMarker(new MarkerOptions().position(new LatLng(globalLat, globalLng)).title(globalLat + " " + globalLng));
                globalLat = 0;
                globalLng = 0;
                break;
            case R.id.btnDrawLines:
                drawLines();
                break;
            case R.id.btnCalculateArea:
                calculateArea();
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

maps activity:

```
package wecsee.caddx.cezarapp

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle

import com.google.android.gms.maps.CameraUpdateFactory
import com.google.android.gms.maps.GoogleMap
import com.google.android.gms.maps.OnMapReadyCallback
import com.google.android.gms.maps.SupportMapFragment
import com.google.android.gms.maps.model.LatLng
import com.google.android.gms.maps.model.MarkerOptions

class MapsActivityXX : AppCompatActivity(), OnMapReadyCallback {
```

```
    private lateinit var mMap: GoogleMap

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_maps_xx)
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        val mapFragment = supportFragmentManager
            .findFragmentById(R.id.map) as SupportMapFragment
        mapFragment.getMapAsync(this)
    }

    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the camera. In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will be prompted to install
     * it inside the SupportMapFragment. This method will only be triggered once the user has
     * installed Google Play services and returned to the app.
     */
    override fun onMapReady(googleMap: GoogleMap) {
        mMap = googleMap

        // Add a marker in Sydney and move the camera
        val sydney = LatLng(-34.0, 151.0)
        mMap.addMarker(MarkerOptions().position(sydney).title("Marker in Sydney"))
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney))
    }
}
```

polugonutil:

```
package wecsee.caddx.cezarapp;

import com.google.android.gms.maps.model.LatLng;

import java.util.List;

import static java.lang.Math.*;

public class PolygonUtils {

    /**
     * The earth's radius, in meters.
     * Mean radius as defined by IUGG.
     */
    static final double EARTH_RADIUS = 6371009;

    /**
     * Returns the area of a closed path on Earth.
     * @param path A closed path.
     * @return The path's area in square meters.
     */
    public static double computeArea(List<LatLng> path) {
        return abs(computeSignedArea(path,EARTH_RADIUS));
    }

    /**
     * Returns the signed area of a closed path on a sphere of given radius.
     * The computed area uses the same units as the radius squared.
     * Used by SphericalUtilTest.
     */
    static double computeSignedArea(List<LatLng> path, double radius) {
        int size =  path.size();
        if (size < 3) { return 0; }
        double total = 0;
```

```
        LatLng prev = path.get(size - 1);
        double prevTanLat = tan((PI / 2 - toRadians(prev.latitude)) / 2);
        double prevLng = toRadians(prev.longitude);
        // For each edge, accumulate the signed area of the triangle formed by the North Pole
        // and that edge ("polar triangle").
        for (LatLng point : path) {
            double tanLat = tan((PI / 2 - toRadians(point.latitude)) / 2);
            double lng = toRadians(point.longitude);
            total += polarTriangleArea(tanLat, lng, prevTanLat, prevLng);
            prevTanLat = tanLat;
            prevLng = lng;
        }
        return total * (radius * radius);
    }

    /**
     * Returns the signed area of a triangle which has North Pole as a vertex.
     * Formula derived from "Area of a spherical triangle given two edges and the included angle"
     * as per "Spherical Trigonometry" by Todhunter, page 71, section 103, point 2.
     * See http://books.google.com/books?id=3uBHAAAAIAAJ&pg=PA71
     * The arguments named "tan" are tan((pi/2 - latitude)/2).
     */
    private static double polarTriangleArea(double tan1, double lng1, double tan2, double lng2) {
        double deltaLng = lng1 - lng2;
        double t = tan1 * tan2;
        return 2 * atan2(t * sin(deltaLng), 1 + t * cos(deltaLng));
    }
}
```