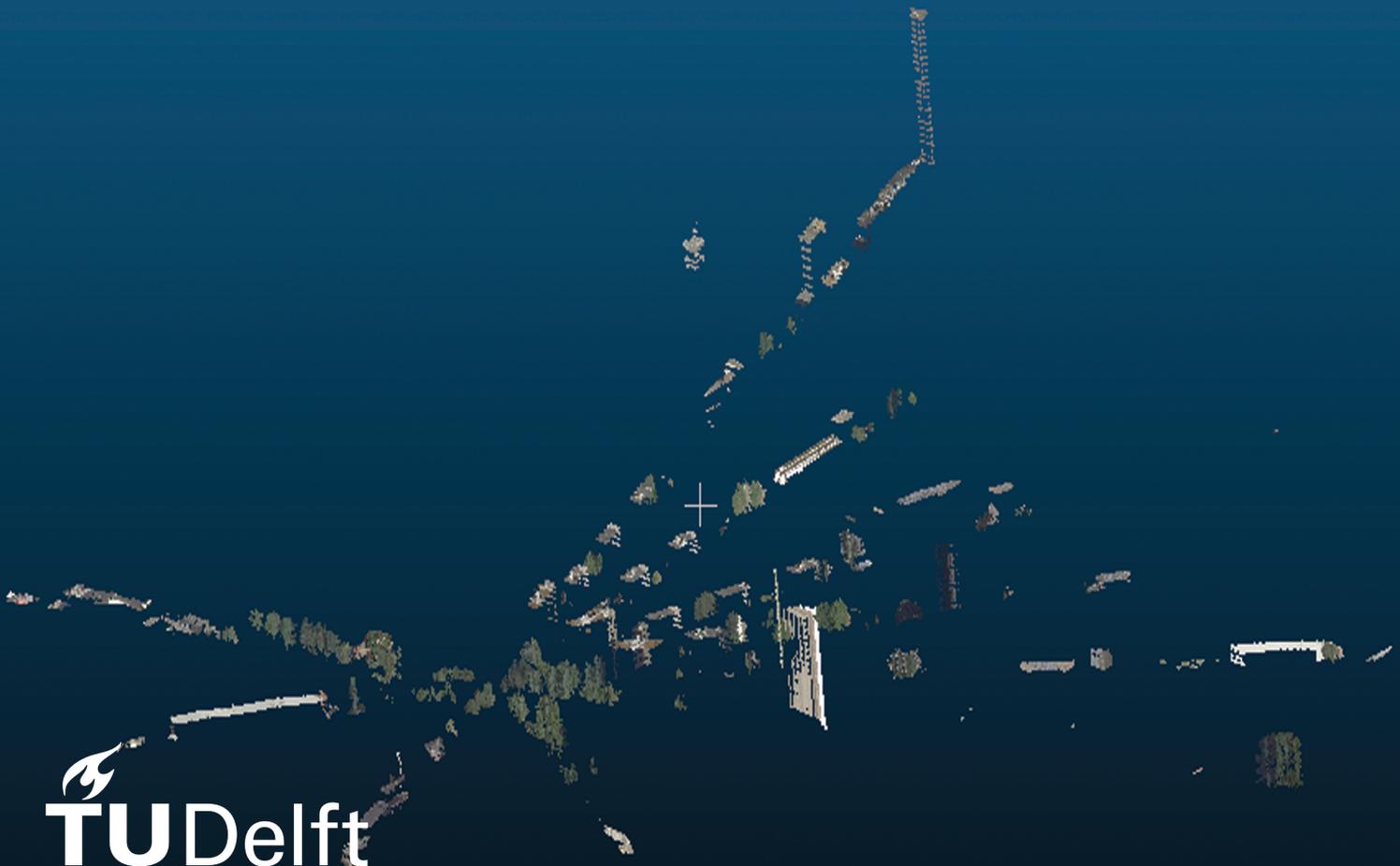


MSc thesis in Geomatics

# Analysis of the visibility of GPS satellites in the urban environment using point cloud representation

Irina Gheorghiu  
2024





MSc thesis in Geomatics

**Analysis of the visibility of GPS satellites  
in the urban environment using point  
cloud representation**

Irina Gheorghiu

January 2024

A thesis submitted to the Delft University of Technology in  
partial fulfillment of the requirements for the degree of Master  
of Science in Geomatics

Irina Gheorghiu: *Analysis of the visibility of GPS satellites in the urban environment using point cloud representation* (2024)

© ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

The work in this thesis was carried out in the:



Geo-Database Management Center  
Delft University of Technology

Supervisors: Ir. Edward Verbree  
Dr. Ir. Martijn Meijers  
Co-reader: ir. Hans Hoogenboom

# Abstract

Global Navigation Satellite System is a spatial data acquisition technique, mostly used in navigation and positioning. One of the main components of this technique is the satellite visibility, which refers to the connection between the satellite and the ground receiver. It is known that the GNSS positioning systems are not as performing in urban areas due to the dense coverage of obstacles (buildings, trees, high terrain etc.). These obstacles can obstruct and reflect the lines of sight between the satellite and the ground receiver which can affect the quality of the performance of the GNSS service. The geometry configuration of the satellites above the receiver is another important aspect that has to be taken into consideration.

This research focuses on implementing a simulation similar to that of GNSS mission planning tools, but using point cloud data as the 3D representation of the surroundings of the receiver and using only the GPS constellation of satellites. Due to the large size of a point cloud sample, two visibility algorithms have been implemented to filter the necessary 3D data. The main output of the simulation are the dilution of precision values which give further information about the satellites' positions. The main purpose of this research is to understand the dilution of precision values, which are directly related to the geometry of the satellite configuration above the receiver. Understanding the behaviour and how the receiver's environment influences the DoP values can result in leading GNSS surveying missions with better results.

This output is then compared with the data acquired from a GNSS receiver in a real scenario. While the results are not favorable for the implemented simulation, it gives a better understanding of the surroundings of the receiver's location by using point cloud data than the already existing online GNSS tools.



# Acknowledgements

First of all, a huge thank you to my supervisors, Edward Verbree and Martijn Meijers who guided me through the process of the thesis. Without your help, both technical and theoretical, and your advice during our meetings this would not have been possible. I have learned a lot during our discussions and while writing this thesis about GNSS, land surveying and data management as both of you are of Geodesy background and explained everything patiently and easy to understand to me. I would also like to thank my co-reader, Hans Hoogenboom who gave me good advice about what needs to be further explained in the thesis so people who are not familiar with this domain can understand as well.

Secondly, I would like to thank my parents, who supported me during these years studying abroad at TU Delft and taught me from the very beginning that I need to be hard-working.

Finally, I would like to thank some of my master peers, especially the ones who learned together with me, helped me learn and explained to me, and who gave me inspiration during this master (you guys know who you are).

I am grateful for the experience and for everyone I have met during my studies abroad, both inside and outside of TU Delft and of the geospatial science field.



# Contents

<b>Acronyms</b>	<b>xvii</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Theoretical background . . . . .	2
1.1.1. Global Positioning System . . . . .	2
1.1.2. Dilution of Precision . . . . .	3
1.2. Problem statement . . . . .	4
1.3. Research question . . . . .	8
1.4. Research goal and scope . . . . .	8
1.5. Reading guide . . . . .	9
<b>2. Related work</b>	<b>11</b>
2.1. Researches and methods implemented in similar studies . . . . .	11
2.2. Research of this thesis . . . . .	16
<b>3. Methodology</b>	<b>17</b>
3.1. Overview of the methodology . . . . .	18
3.2. Reading the satellite data into ECEF coordinate system . . . . .	19
3.3. Position of the ground antenna . . . . .	21
3.4. Determining the receiver-satellite direction . . . . .	24
3.5. CRS conversion pipeline . . . . .	24
3.6. Visibility algorithm . . . . .	25
3.7. Conversion from ECEF to ENU . . . . .	31
3.8. DoP calculation algorithm . . . . .	32
3.9. Azimuth and Elevation Angles . . . . .	34
3.10. GNSS data measurements . . . . .	35
<b>4. Implementation and Results</b>	<b>39</b>
4.1. Resources . . . . .	39
4.1.1. AHN4 Point Clouds . . . . .	39
4.1.2. NMEA Tools . . . . .	40
4.1.3. CloudCompare . . . . .	41
4.2. Implementation . . . . .	44
4.2.1. CRS conversion implementation . . . . .	44
4.2.2. Azimuth and Elevation computation . . . . .	46
4.2.3. Visibility algorithm . . . . .	46
4.3. Results . . . . .	54
4.3.1. First Test Site . . . . .	55
4.3.2. Second Test Site . . . . .	61
4.3.3. Third Test Site . . . . .	67
4.3.4. Fourth Test Site . . . . .	73
4.3.5. Analysis of results . . . . .	79

*Contents*

<b>5. Conclusion and Discussion</b>	<b>85</b>
5.1. Answering the secondary research questions . . . . .	85
5.2. Answer to the main research question . . . . .	87
5.3. Sources of errors and limitations . . . . .	90
5.4. Recommendations and future work . . . . .	90
<b>A. GNSS Measurement Data</b>	<b>93</b>

# List of Figures

1.1.	Signal obstruction from satellite to receiver (Lu and Han)	1
1.2.	The three segments (Azorsensors)	3
1.3.	Good GDOP (Dana, 1994)	4
1.4.	Poor GDOP (Dana, 1994)	4
1.5.	Methodology for the Sky DEM application (Gandolfi and La Via, 2011)	6
1.6.	Menu to add the obstruction from the Navmatix tool	7
1.7.	Simple three-dimensional blocks as obstructions	7
2.1.	Complex 3D environment (Dandurand et al., 2019)	11
2.2.	Occlusion algorithm for vegetation points (Zhang et al., 2021)	12
2.3.	Shadow matching concept (Groves, 2011)	13
2.4.	Availability of GPS during a one day period (Tiberius and Verbree, 2003)	14
2.5.	DEM nodes investigated to define the obstacle mask (Gandolfi and La Via, 2011)	14
2.6.	Sky View Factor product (Bouzas et al., 2018)	15
2.7.	Schematic of identifying visible satellites (Yuen, 2009)	15
2.8.	Visible Galileo+GPS satellites from test point (van Capel et al., 2022)	16
3.1.	Main methodology for the research	17
3.2.	Relation between ellipsoid and geoid (van der Marel, 2020)	22
3.3.	Geoid Height Map for the Netherlands (Bruijne et al., 2005 - in van der Marel, 2020)	23
3.4.	NTv2 transformation version for RDNAPTRANS2018 (van der Marel, 2020)	25
3.5.	Computing the orientation of the boxes using the differences on the XY axis	26
3.6.	Rectangular bounding boxes along the ray between the satellite and the receiver (X coordinate on the horizontal axis and Y coordinate on the vertical axis)	26
3.7.	Implementing the square boxes (green) compared to the rectangle boxes (purple)	27
3.8.	Square bounding boxes on the perpendiculars of the satellite-receiver line (X coordinate on the horizontal axis and Y coordinate on the vertical axis)	27
3.9.	Cylinder between P1 (receiver) and P2 (satellite), with Q being the point to be tested - obtaining vectors P1Q and P1P2	28
3.10.	ENU vs ECEF ( <a href="https://en.wikipedia.org/wiki/Local_tangent_plane_coordinates">https://en.wikipedia.org/wiki/Local_tangent_plane_coordinates</a> )	31
3.11.	Azimuth and elevation angles computed from point A to point B (van der Marel, 2020)	34
3.12.	NMEA strings to be analysed	35
4.1.	LAS Point Data Record format table (Ledoux et al., 2021)	40
4.2.	The first 10 classes of points from AHN (Ledoux et al., 2021)	40
4.3.	Log screen	41
4.4.	Settings for the strings	41

List of Figures

4.5. Trace polyline tool (middle pictogram) and the Point Picking tool (left pictogram) . . . . .	42
4.6. Tracing a line between two points . . . . .	42
4.7. Line that serves as the cylinder ax . . . . .	43
4.8. Using the Point Picking tool of CloudCompare . . . . .	44
4.9. PROJ Commands, (Lesparre et al., 2018) . . . . .	44
4.10. Schema of transformations (Lesparre, 2023) . . . . .	45
4.11. Visibility methodology . . . . .	46
4.12. Choosing the radius of the cylinder - the diameter (twice the radius - green line) of the cylinder in 2D (black parallelogram) is twice the diagonal of a square bounding box . . . . .	48
4.13. 20 meters limit for the diagonal . . . . .	48
4.14. 50 meters limit for the diagonal . . . . .	49
4.15. 100 meters limit for the diagonal . . . . .	49
4.16. Evaluation of the time used by the visibility algorithm based on the size of the diagonal threshold . . . . .	50
4.17. Whole point cloud used as representation of the environment . . . . .	51
4.18. Point cloud after LASTools height filter . . . . .	51
4.19. Point cloud after the 2D algorithm filter . . . . .	52
4.20. The 2D algorithm and the 3D algorithm . . . . .	52
4.21. Point cloud after the 3D algorithm filter . . . . .	53
4.22. Skyplot for the 9:00:00 UTC time - 21st of November . . . . .	55
4.23. Skyplot for the 9:20:00 UTC time - 21st of November . . . . .	56
4.24. First test site used for data acquisition . . . . .	57
4.25. Statistics for the visible satellites for the chosen threshold - Test Site 1 . . . . .	59
4.26. DoP values calculated by the smartphone application - Test site 1 . . . . .	60
4.27. DoP values calculated by the simulation (each 5 minutes) - Test site 1 . . . . .	60
4.28. Skyplot for the 9:30:00 UTC time - 21st of November . . . . .	61
4.29. Skyplot for the 9:50:00 UTC time - 21st of November . . . . .	62
4.30. Second test site used for data acquisition . . . . .	63
4.31. Statistics for the visible satellites for the chosen threshold - Test Site 2 . . . . .	65
4.32. DoP values calculated by the smartphone application - Test site 2 . . . . .	66
4.33. DoP values calculated by the simulation - Test site 2 . . . . .	66
4.34. Skyplot for 15:40:00 - UTC Time (5th of December) . . . . .	67
4.35. Skyplot for 15:50:00 - UTC Time (5th of December) . . . . .	68
4.36. Third location of data acquisition . . . . .	69
4.37. Statistics for the visible satellites for the chosen threshold - Test Site 3 . . . . .	71
4.38. DoP values calculated by the smartphone application - Test site 3 . . . . .	72
4.39. DoP values calculated by the simulation - Test site 3 . . . . .	72
4.40. Skyplot for 15:00:00 - UTC Time (18th of December) . . . . .	73
4.41. Skyplot for 15:10:00 - UTC Time (18th of December) . . . . .	74
4.42. Forth location of data acquisition . . . . .	75
4.43. Statistics for the visible satellites for the chosen threshold - Test Site 4 . . . . .	77
4.44. DoP values calculated by the smartphone application - Test site 4 . . . . .	78
4.45. DoP values calculated by the simulation - Test site 4 . . . . .	78
4.46. DoP values presented by the Navmatix mission planning tool - 21st of November	80
4.47. DoP values presented by the Navmatix mission planning tool - 5th of December	80
4.48. DoP values presented by the Navmatix mission planning tool - 18th of December	81
4.49. Point distribution for cylinder ray towards Satellite 03 - Test Site 3, MoT 15:53:57 UTC, 5th of December, blocked . . . . .	82

4.50. Point distribution for cylinder ray towards Satellite 06 - Test Site 3, MoT 15:53:57 UTC, 5th of December, visible . . . . .	82
4.51. Point distribution for cylinder ray towards Satellite 19 - Test Site 3 MoT 15:53:57 UTC, blocked . . . . .	83
5.1. Visible satellites (green dots) and the LoS (pink lines) towards the receiver . . .	89



# List of Tables

3.1. GNCGA string explanation . . . . .	36
3.2. GNCSA string explanation . . . . .	37
3.3. GPGSV string explanation . . . . .	37
4.1. Visibility analysis for test site 1 - 09:00:00 . . . . .	57
4.2. Visibility analysis for test site 1 - 09:05:00 . . . . .	57
4.3. Visibility analysis for test site 1 - 09:10:00 . . . . .	58
4.4. Visibility analysis for test site 1 - 09:15:00 . . . . .	58
4.5. Visibility analysis for test site 1 - 09:20:00 . . . . .	58
4.6. Visibility analysis for test site 2 - 09:33:41 . . . . .	63
4.7. Visibility analysis for test site 2 - 09:38:41 . . . . .	63
4.8. Visibility analysis for test site 2 - 09:43:41 . . . . .	64
4.9. Visibility analysis for test site 2 - 09:48:41 . . . . .	64
4.10. Visibility analysis for test site 2 - 09:53:41 . . . . .	64
4.11. Visibility analysis for test site 3 - 15:41:57 . . . . .	69
4.12. Visibility analysis for test site 3 - 15:44:57 . . . . .	69
4.13. Visibility analysis for test site 3 - 15:47:57 . . . . .	70
4.14. Visibility analysis for test site 3 - 15:50:57 . . . . .	70
4.15. Visibility analysis for test site 3 - 15:53:57 . . . . .	70
4.16. Visibility analysis for test site 4 - 15:02:00 . . . . .	75
4.17. Visibility analysis for test site 4 - 15:05:00 . . . . .	75
4.18. Visibility analysis for test site 4 - 15:08:00 . . . . .	76
4.19. Visibility analysis for test site 4 - 15:11:00 . . . . .	76
4.20. Visibility analysis for test site 4 - 15:14:00 . . . . .	76
4.21. DoP Values . . . . .	79
4.22. Percentage of AHN classes which serve as obstructions inside the cylinder ray	83
A.1. GNSS measurements for test site 1 - 09:00:00 . . . . .	93
A.2. GNSS measurements for test site 1 - 09:05:00 . . . . .	93
A.3. GNSS measurements for test site 1 - 09:10:00 . . . . .	94
A.4. GNSS measurements for test site 1 - 09:15:00 . . . . .	94
A.5. GNSS measurements for test site 1 - 09:20:00 . . . . .	95
A.6. GNSS measurements for test site 2 - 09:33:41 . . . . .	95
A.7. GNSS measurements for test site 2 - 09:38:41 . . . . .	95
A.8. GNSS measurements for test site 2 - 09:43:41 . . . . .	96
A.9. GNSS measurements for test site 2 - 09:48:41 . . . . .	96
A.10. GNSS measurements for test site 2 - 09:53:41 . . . . .	96
A.11. GNSS measurements for test site 3 - 15:41:57 . . . . .	97
A.12. GNSS measurements for test site 3 - 15:44:57 . . . . .	97
A.13. GNSS measurements for test site 3 - 15:47:57 . . . . .	98
A.14. GNSS measurements for test site 3 - 15:50:57 . . . . .	98
A.15. GNSS measurements for test site 3 - 15:53:57 . . . . .	98

*List of Tables*

A.16.GNSS measurements for test site 4 - 15:02:00 . . . . .	99
A.17.GNSS measurements for test site 4 - 15:05:00 . . . . .	99
A.18.GNSS measurements for test site 4 - 15:08:00 . . . . .	99
A.19.GNSS measurements for test site 4 - 15:11:00 . . . . .	100
A.20.GNSS measurements for test site 4 - 15:14:00 . . . . .	100

# Acronyms

GNSS - Global Navigation Satellite Systems

GPS - Global Positioning System

PRN - Pseudo-Random Noise

DoP - Dilution of Precision

LoS - Line of Sight

DEM - Digital Elevation Model

DSM - Digital Surface Model

ECEF - Earth Centered Earth Fixed

RDNAP - Rijksdriehoeksmeting and Normaal Amsterdams Peil

LIDAR - Light Detection and Ranging

3D - Three Dimensional

MoT - Moment of time



# 1. Introduction

According to [Lu and Han\(2020\)](#), the performance of the Global Navigation Satellite System is strongly correlated to the position of the ground receiver. However, the quality of the results given by the GNSS assembly (the constellation of satellites and the ground receiver) is influenced by the environment in which the receiver is positioned. The urban areas, known as urban canyons, which contain tall buildings, narrow streets and trees can have thus a negative impact regarding satellite visibility. Satellite visibility implies the connecting line between the ground receiver and the satellites in the sky which will further be referred to in this study as the line of sight (LoS).

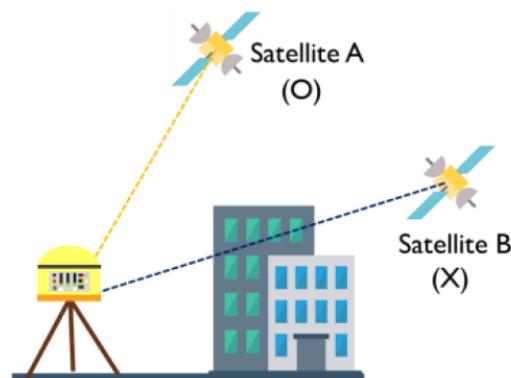


Figure 1.1.: Signal obstruction from satellite to receiver ([Lu and Han](#))

Combining different GNSS constellations will result in a larger number of LoS. It has to be mentioned that accurate positioning results cannot be obtained without the receiver connecting to four or more satellites. Most of these studies are carried using three dimensional models of the urban areas, in combination with other two-dimensional data, such as cadastral maps or rasters, in order to predict which satellites are visible from different locations and in what moment of time.

Aerial LIDAR point clouds were used for previous studies, due to their reliability when it comes to offering three-dimensional highly detailed information and representation of topographic elements which can influence the LoS. Based on [Zhang et al.\(2017\)](#), point clouds can be used without the need of generating surfaces. However, it has to be mentioned that point cloud analysis means a lot of resources and thus methods were developed to reduce the computation costs. The aerial borne point clouds were also mostly combined with other ways of representation. Zhang et al. also present a certain disadvantage of the aerial point clouds, as they do not capture the façade information correctly, but they can still be used as the position of the building is easy to determine from the position of the rooftop which is mostly accurately captured.

## 1. Introduction

Dandurand et al., (2019) referred to GNSS as two parts: the ground receiver antennas and the constellation of satellites. Their method implies calculating the distance from satellite to the GNSS receiver - pseudorange - using the time difference between its internal clock and the clock from the signal's satellite. It states that one of the main factors of the quality of these measurements is the number of visible satellites. The paper further describes the beneficial usage of point clouds as the environment in which the ground receiver is placed due to their ability to represent tall vegetation, as trees behave in a different way regarding satellite signals than buildings do.

### 1.1. Theoretical background

#### 1.1.1. Global Positioning System

The GPS satellite constellation consists of a number of 24-31 working satellites. Developed in the United States of America originally for military purposes, it now works anywhere and anytime around the world. The positioning system consists of three segments:

- Space segment: Satellites - the space vehicles; each satellite has its own PRN code which helps in distinguishing it
- Control segment: Sixteen monitor stations around the Earth which track the satellites and send correction for the satellite orbits and clocks.
- User segment: Ground receiver equipment; receives the signals from the satellites and based on the information from the satellite computes its position on Earth

A receiver on any part of the planet is said to have access to 6 satellites from the GPS constellation in an open-sky environment (Novatel, 2015).

The range measurement between the satellite and the receiver is determined by the difference of the transmission time compared to the time it was received by the ground antenna. Knowing that the signal travels with the speed of light, the range can be easily computed by multiplying the time difference with the speed of light.

The range measurements help in determining the three coordinates of the receiver, so three satellites are needed for that. However, the clocks of the receiver is not as performing as that of the satellite and this difference between the clocks creates errors. Because of this, the clock of the receiver either advances or delays its time based on that of the satellite, until the other three satellites converge at one point. This requires a fourth satellite to determine the position of the ground receiver. (Novatel, 2015)

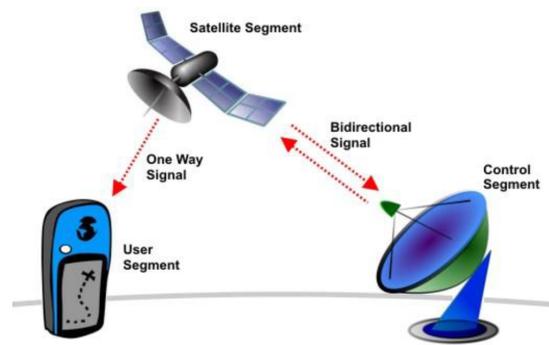


Figure 1.2.: The three segments (Azorsensors)

### 1.1.2. Dilution of Precision

Dilution of precision are described as numerical representations of the geometry of the satellites and are dependent on the positions of satellites that are visible to the receiver. Small DoP results in a better result of position and time calculation. The opposite happens for very high DoP values (Novatel, 2015).

From the mathematical aspect, the DoP could be explained as the ratio between the standard deviation of a specified parameter of the position and the pseudorange. From the physical aspect, the DoP values represent the strength of the visible satellites' signals based on their geometry related to the ground receiver. Such a situation is described by wide angles relative to each other (Yuen, 2009)

The Geometrical DoP has four components:

- PDOP - Position Dilution of Precision (3D)
- HDOP - Horizontal Dilution of Precision (Latitude, Longitude)
- VDOP - Vertical Dilution of Precision (Height)
- TDOP - Time Dilution of Precision (Time)

(Dana, 1994) <sup>1</sup>

---

<sup>1</sup>[https://foote.geography.uconn.edu/gcraft/notes/gps/gps\\_f.html](https://foote.geography.uconn.edu/gcraft/notes/gps/gps_f.html)

## 1. Introduction

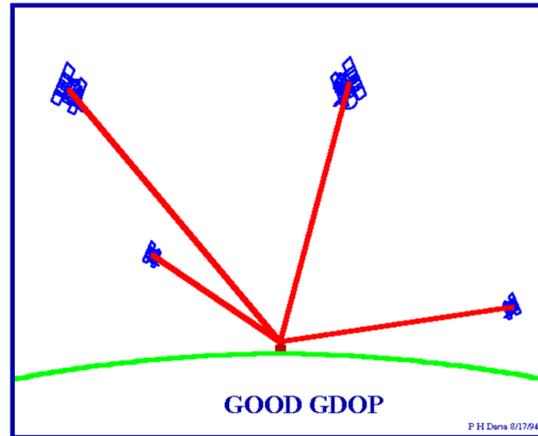


Figure 1.3.: Good GDOP (Dana, 1994)

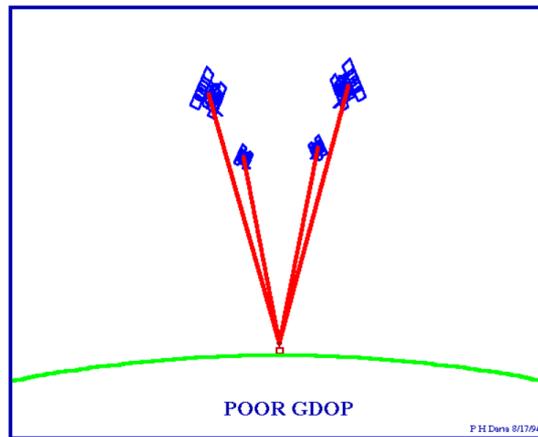


Figure 1.4.: Poor GDOP (Dana, 1994)

As it can be seen in the figures 1.3 and 1.4, the volume of the shape defined by the ranges from the satellites to the receiver and the size of the angles represent the strength of the geometrical configuration of the satellites. A larger volume and angles represent better satellite geometry and thus lower DoP values.

## 1.2. Problem statement

DoP values can be used for the selection of the satellites which give the position solution. (Novatel, 2015)

GNSS mission planning tools are software programs which help land surveyors in assessing the way how the future scenario of surveying mission will go and which satellites will be available at the time of the scenario by creating a simulation. This simulation will return usually the following products under the form of plots:

- Dilution of precision values
- Number of satellites
- Elevation
- Visibility
- Sky Plot

All surveying works will require a certain precision level. Studying DoP values will be helpful for this research as these given measurements will give details about the accuracy of the position determined by the satellite constellation. Not only it will provide information about the obstructed LoS, but also about the distribution of the satellites from the constellation above the receiver and their elevation angle. The study of DoP values can also help to better understand the geometry of the satellite constellation and thus predict configurations that will give better results in terms of precise positioning. DoP values.

Several such applications have been developed. [Gandolfi and La Via](#) focuses on integrating such a tool with the digital elevation models (DEM) in order to create a mask. The paper states that most of these skyplot generating tools return the products related to pure ellipsoids surfaces and by adding a cut-off angle which will filter out the satellites that have the elevation angle lower than the cut-off angle. This mask will be used to determine the visible satellites which can be used for the mission scenario and create a simulation as realistic as possible. In figure 1.5, it is presented the methodology for this GNSS mission planning tool.

## 1. Introduction

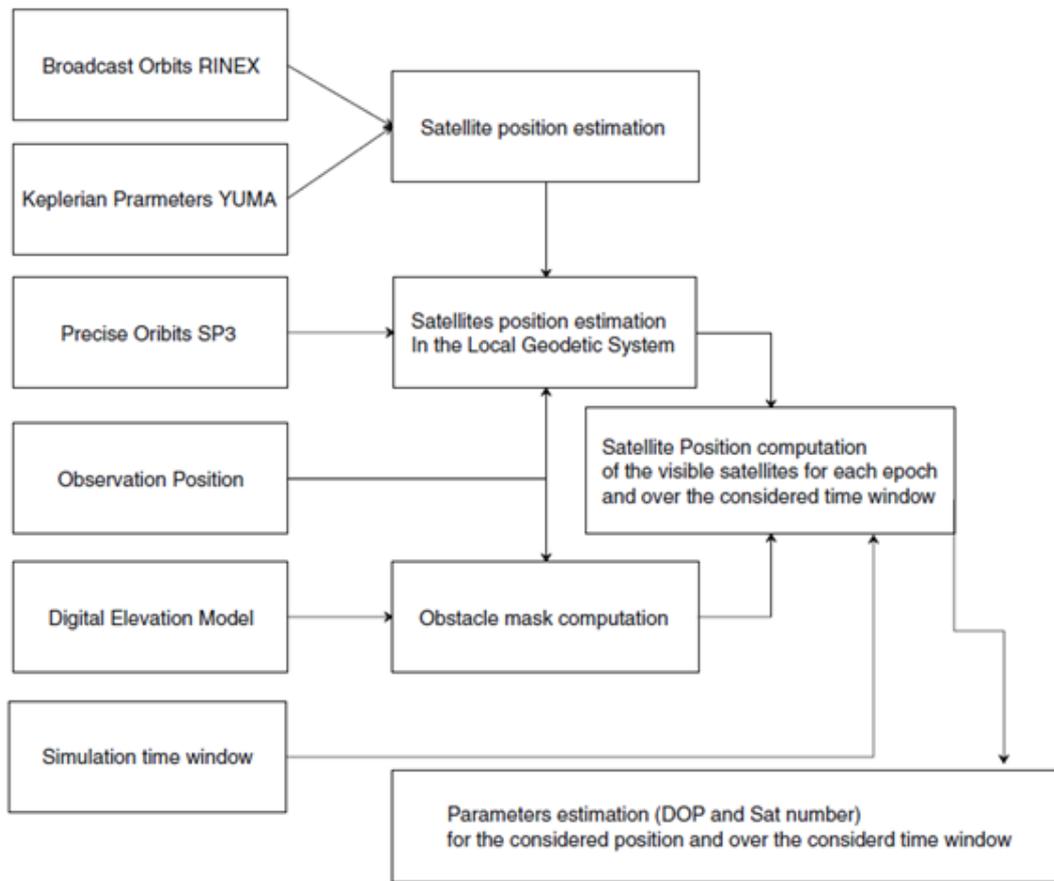


Figure 1.5.: Methodology for the Sky DEM application (Gandolfi and La Via, 2011)

Upon a simple look at the online GNSS mission planning tools, it has been discovered that indeed they lack the necessary obstacle generating tools in order to create a realistic simulation of the future mission. Two of the most used online GNSS mission planning tools were evaluated for this research.

The first example <sup>2</sup> is the Trimble online GNSS planning tool. When it comes to using the tool with limited visibility, the only way is to introduce a simple cut-off elevation angle in degrees and the height of the receiver in metres. It does not contain further built-in features to create a more complex environment.

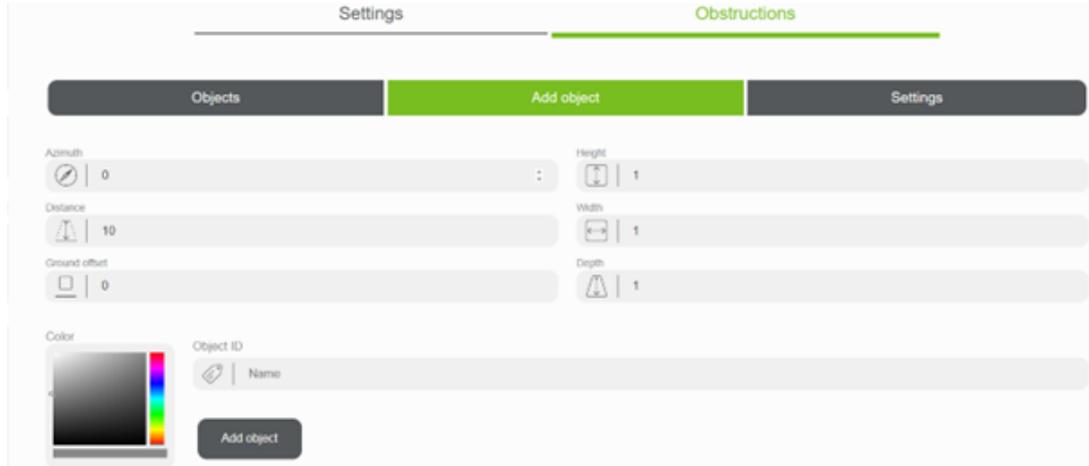
The second example <sup>3</sup> is the online GNSS planning tool by Navmatix. Apart from adding the height and the cut-off angle, this tool does indeed contain a built-in obstruction feature which allows the user to insert three-dimensional objects. While it is up to the user to choose how and where to insert the object based on keyboard introduced characteristics, the introduced object is a simple block, with no further details. While this is a better designed

<sup>2</sup><https://www.gnssplanning.com//settings>

<sup>3</sup><http://gnssmissionplanning.com/App/Settings>

## 1.2. Problem statement

feature for representing the environment in which the GNSS receiver is, it still does not create a realistic enough environment.



The screenshot shows the 'Obstructions' menu in the Navmatix tool. The interface is split into two main sections: 'Settings' and 'Obstructions'. The 'Obstructions' section is currently active, indicated by a green underline. Below this, there are three tabs: 'Objects', 'Add object' (highlighted in green), and 'Settings'. The 'Add object' tab contains several input fields for defining an obstruction: 'Azimuth' (0), 'Height' (1), 'Distance' (10), 'Width' (1), 'Ground offset' (0), and 'Depth' (1). There is also a color selection tool, an 'Object ID' field with a 'Name' input, and an 'Add object' button.

Figure 1.6.: Menu to add the obstruction from the Navmatix tool

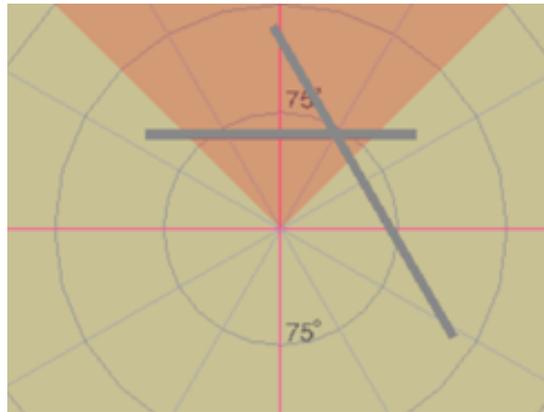


Figure 1.7.: Simple three-dimensional blocks as obstructions

### 1.3. Research question

The main research question of the thesis: To what extent can point cloud data be integrated in a GNSS planning simulation and how does this relate to satellite geometry and influences DoP values?

The following subquestions will also be taken into consideration:

- How are obstructions represented in calculation of DoP?
- How good is the carried analysis compared with the given result by simulation tools / GNSS equipment?
- How can such a simulation of a GNSS mission be helpful?
- What determines the threshold that decides if a line of sight is visible or not?

### 1.4. Research goal and scope

The purpose of this type of research has been explored frequently in other scientific works in this domain. Such a study of GNSS planning tools is useful to understand the importance of satellite geometrical configuration in the land surveying domain. Understanding the geometrical configuration of the satellite constellation helps in improving the results of actual GNSS surveying missions. Further scope of this project is to use and improve existing methods for integrating point clouds into satellite visibility analysis tools and parameter determination such as calculation of the DoP values, azimuth, elevation. The used method shall be compared with the already existing online tools.

The scope of this study is to discover how the environment of the urban areas influences the number of visible satellites and as a result, the DoP values. A certain location in the area of Delft at a given moment of time will be chosen as a site for a test study.

This thesis will focus on two aspects:

- The performance of GNSS tools
- The usage of huge amounts of point cloud data

The first aspect will be covered in comparing the DoP values resulted from the simulation with those gathered from GNSS equipment during the final test survey. The second aspect will be covered in the efficiency of the implemented method used to determine how the used point cloud data obstructs the line of sight between the satellite and the receiver. The method will be first focused on the 2D aspect of and then the 3D aspect of the data.

## 1.5. Reading guide

This thesis consists of 5 chapters:

The Introduction chapter presents to the reader the theoretical background of geometrical aspects of land surveying and positioning systems and how it is combined with 3D data. The introduction then continues with the focus of this thesis (problem statement, research questions and goal).

The second chapter is the Related Work. This chapter goes further into the theoretical background and presents what has been done so far in similar studies such as the one of the thesis. This chapter ends with a small conclusion on what is missing and has not been yet covered.

The third chapter is the Methodology. The chapter starts with a diagram of the steps. Each step is briefly covered in the Overview section to let the reader know about the following sections of the third chapter. The chapter then goes on explain in detail each step in the sections.

The fourth chapter is the Implementation and Results. It begins with enumerating the resources needed for the implementations. This chapter is then split in two parts. The implementation part presents and explains the technical details of how the methodology was implemented. The results part presents the results of this research: the real case versus simulation. The results are then analysed in a separate section.

The last chapter is the Conclusion. This chapter contains the answers and explanations to the research questions from the first chapter. The chapter continues further with the Error section and the Future Work and Recommendations.



## 2. Related work

### 2.1. Researches and methods implemented in similar studies

Studies have been carried out the subject of satellite visibility in point clouds, as point clouds are a reliable source of 3D data.

Lu and Han (2020) presented the method of superimposed column which implied taking into consideration the height of the point cloud object and its position represented in coordinates, reducing considerably the size of the point cloud data that had to be used. After applying this method, the maximum elevation angle is calculated in order to determine the DoP values.

Dandurand et al. (2019) created a prediction model which took in consideration both the geometry of the satellite constellation and the environment in which the receiver is in. The model created will determine based on the shape of the local point cloud if the satellite signal will either have occlusion or absorption. The point clouds used for this study are dense and represent forest areas.

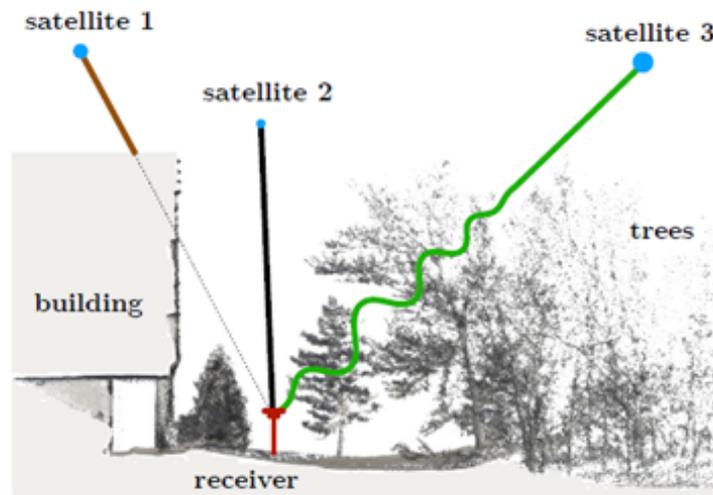


Figure 2.1.: Complex 3D environment (Dandurand et al., 2019)

Zhang et al. (2021) use occlusion analysis methods to determine the visibility between a viewpoint and a target point. Two methods are proposed for building and vegetation point clouds. The methods include choosing a threshold to decide if a LoS is blocked or not based on the number of obstacles surrounding the search point and creating a 3D sphere around a vegetation point to see how many points fall inside it and decide that the line is occluded.

## 2. Related work

The method uses the kdtree to save the number of obstructing points around the target points.

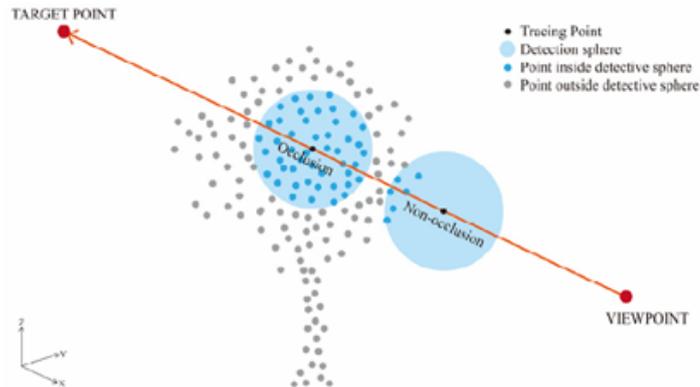


Figure 2.2.: Occlusion algorithm for vegetation points (Zhang et al., 2021)

Another visibility study, although the method described might not be suitable for point clouds, is the one described in Suzuki and Kubo (2015). Using virtual fish eye images and Google Earth simulated images, the algorithm creates a mask which takes into consideration only the open sky and thus the satellites that are above the area of the open sky based on their elevation angles. This method uses only images, having no need for any kind of laser scanning obtained data.

The paper written by Groves (2011) introduces shadow matching as a method to estimate satellite visibility in the urban environment. The method is tested on an ideal 3D model which has the needed width and spacing of streets and buildings and the position is discovered using the geometry between the user and the satellite. The real-life problem also raises the utility of the non-line of sight. This effect appears when the satellite signal is reflected by a building. A combination of LoS and NLoS can also result in accurate positioning. The method however takes into consideration direct LoS and their DoP values.

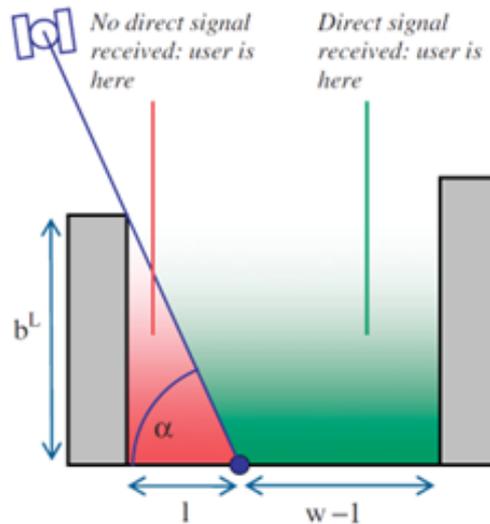


Figure 2.3.: Shadow matching concept (Groves, 2011)

In another study by [Adjrad and Groves \(2017\)](#), it is discussed the performance of the service using also the NLoS. In theory, using the NLoS might not influence the performance of the GNSS services in a good way as using NLoS also adds to the errors. However, in practice, there are very few clear LoS detectable in the urban environment, so using NLoS and signals influenced by multipath might add valuable contribution to the positioning accuracy. The paper presents two positioning algorithms and two integration approaches described in steps. The conclusion was that both the GNSS ranging-based algorithm and the shadow matching one worked better with a covariance based weighting approach rather than a deterministic one.

[Verbree et al. \(2004\)](#) describes a visibility algorithm which uses both GPS and Galileo satellite services to determine positioning in a 3D city model of Delft. This visibility algorithm determines the availability of the satellites. Using both the GPS and Galileo constellation asks for 5 satellites (2 GPS or 3 Galileo or the other way round). The conclusions state that a single satellites constellation is not enough to determine accurate positioning in urban areas. It is also stated that 3 satellites will be enough to determine the position if the height information is determined and constant and even provide better results than with 3 satellites.

[Tiberius and Verbree \(2005\)](#) describes a combination between GPS, EGNOS and Galileo positioning services for more accurate results. The availability study is focused on the user segment as the terrestrial geometry influences the receive of satellite signals. An additional algorithm next to the visibility algorithm is implemented. This algorithm calculates the availability of "enough" GPS and Galileo satellites while the receiver sits in a location. These numbers are represented on a map. The paper also has an outlook on finger-printing outdoor positioning. This method implies finding the user position by finding out which satellites at a given time can be tracked or not in an already known location and then match it with the user's location.

## 2. Related work

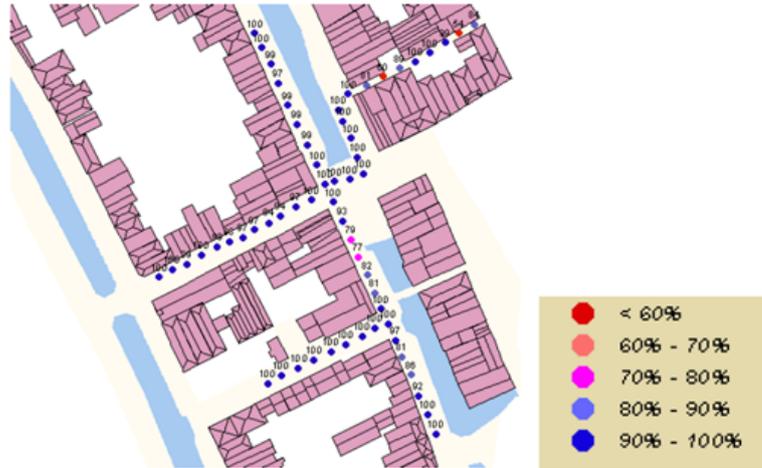


Figure 2.4.: Availability of GPS during a one day period (Tiberius and Verbree, 2003)

The [Gandolfi and La Via \(2011\)](#) paper also mentioned in the 1.3 section presented the steps in the implementation of a mission planning tool in C++. The said application uses as representation of the Earth's surface DEMs. In order to manage huge amounts of 3D data, a filtering algorithm which determines the largest radius used to determine the LoS based on the value of the cut-off angle was used.

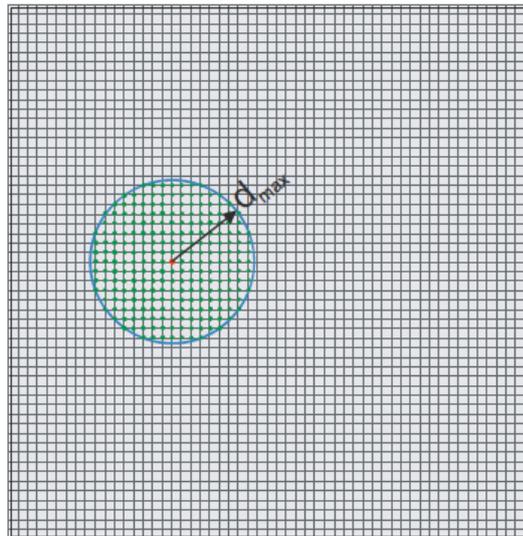


Figure 2.5.: DEM nodes investigated to define the obstacle mask ([Gandolfi and La Via, \(2011\)](#))

The [Bouzas et al. \(2018\)](#) student project written at TU Delft presented an application which determined the field of view from a point and represented it as a sky plot. The obstruction used were the buildings and the trees. The methodology of the project proposes the creation of a dome as a representation of the sky and then projecting the point clouds on it. In order for the computation of the obstructing points to be easier, a 2D approach was adopted after

## 2.1. Researches and methods implemented in similar studies

calculating the projected area of the sections on the XY-plane as the 3D approach would have been too time-consuming.

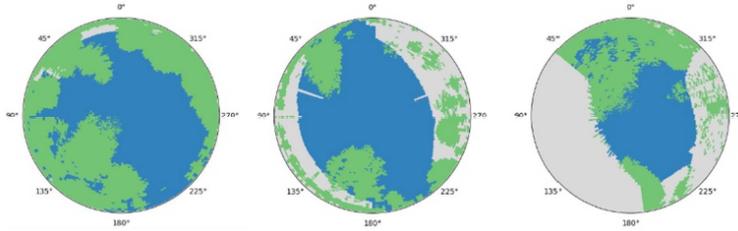


Figure 2.6.: Sky View Factor product (Bouzas et al., 2018)

The thesis written by Yuen (2009) provided research onto the calculation of the DoP values and the influence of the visible satellites based on the GPS constellation alone. The thesis provides insight into the calculation of the satellite positions in time using the almanac and the calendar conversions in the Earth Centered Earth Fixed frame and then converting the positions into East-North-Up to help in easily calculating the DoP values. For the obstruction part, the thesis uses the cut-off angle condition to determine which satellite are visible or not.

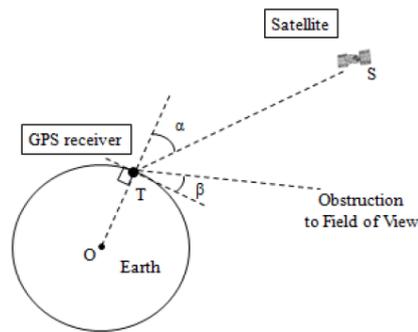


Figure 2.7.: Schematic of identifying visible satellites (Yuen, 2009)

Another student project from TU Delft, van Capel et al. (2022), presented the calculation of the LoS between the Galileo and GPS satellites which were received by the terrestrial antenna during the field measurements in Olpe, Germany as a validation of the service's performance. The visibility algorithm was based on determining the sign of the volume of the form created by the satellite-receiver lines and the 3D model of the buildings which were surrounding the receiver.

## 2. Related work

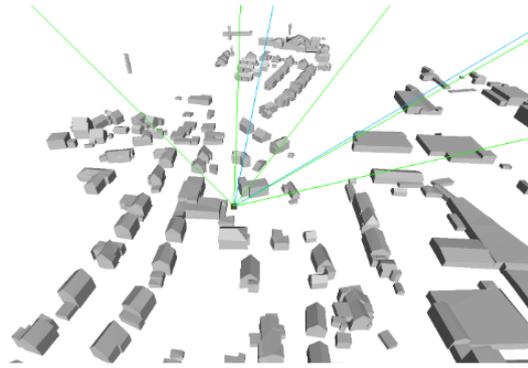


Figure 2.8.: Visible Galileo+GPS satellites from test point (van Capel et al., 2022)

## 2.2. Research of this thesis

GNSS mission planning tools are helpful in offering land surveyors beforehand a theoretical scenario of how the field measurements are going to work and appear based on the parameters of positions of satellites on that time and site. These parameters include the number of satellites that are above the site in that moment and their angles based on their position to the receiver.

As seen in section 1.2, the already existing free online tools offer the user in their interface very limited options (simple 3D blocks, obstruction angle) to approximate the environment geometry in which the receiver antenna is. The geometry of the surroundings of the receiver plays a big role in the precision of the measurements and eventually, to the final product of the surveying campaign.

There seems to be a lack of study regarding the usage of point cloud data in surveying mission tools due to its big size and long computation time. The research of this thesis is to find ways to integrate it in such a manner that it does not take a long time to compute and also without losing important aspects of the data.

The implementation for this thesis will approximate the LoS with a cylinder than with an actual line from the mathematical aspect. This is because the LoS implemented as a mathematical line can easily pierce through point cloud objects. The cylinder LoS is implemented to filter in point clouds inside it and decide based on a threshold if the satellite is considered visible or not.

### 3. Methodology

The methodology presented in this thesis is divided into steps which will be further discussed in the following sections. This chapter begins with an explanation diagram for the order of the steps, with an overview section of the steps and then continues with sections where each step is explained.

Some of the steps in the methodology, such as the usage of the GPS almanac, the conversion from ECEF to ENU and the DoP calculation algorithm, are the same as the ones written by Yuen in his master thesis.

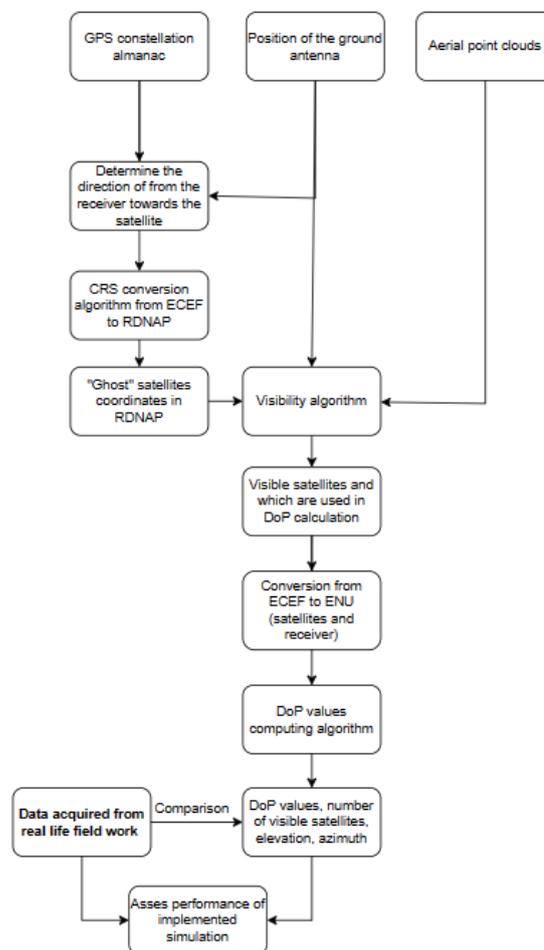


Figure 3.1.: Main methodology for the research

## 3.1. Overview of the methodology

The first step of the methodology is to obtain the needed data for this analysis. This includes choosing the right GPS almanac for the moment in time and period of the surveying campaign, recording the coordinates of the place that the receiver will be in and choosing the 3D data, in this case, the point cloud set, for a better representation of the environment. The data from the orbital elements in the GPS constellation almanac will be transformed to ECEF coordinates as it is easier to use due to its representation of Earth's surface and to work with coordinates expressed in meters rather than the latitude and longitude coordinates represented in degrees, minutes and seconds. As explained [here](#)<sup>1</sup>, ECEF is the geocentric coordinate system, representing locations in X, Y, Z coordinates from the center of the Earth. As its reference frame rotates together with the Earth, it is most useful in tracking satellites and GPS uses ECEF as its primary coordinate system.

Having both the position of the receiver and the coordinates of the satellites expressed in the same coordinate system, the following step is to determine the satellite-receiver direction in order to ease the further steps computations by adding helping points on the line between the satellites and the receiver. The case of study is in the Netherlands which has low height on average and it was unnecessary to work with the whole satellite-receiver distance which is a large value. A way shorter distance from the receiver towards the helping point ("ghost" satellite) will be used. Using the positions of the "ghost" satellites instead of the ones of the real satellites also help in the later step of coordinate conversion.

After calculating the satellite-receiver directions, a CRS conversion must be implemented from ECEF to RDNAP as both the coordinates of the GNSS segments (satellites and receiver) and the 3D environment data have to be in the same coordinate system. As converting the 3D data to ECEF is computationally heavy, the positions of the satellites (namely the "ghost" satellites) and the receiver will be converted to the coordinate system of the 3D data.

The next step after transforming all the needed data to the same coordinate system is to implement the visibility algorithm which has two parts:

- a 2D method to filter the points and leave only on the ones which are on the satellite-receiver trajectory
- a 3D method to actually determine if the points represent objects which are obstacles or not

The result of this algorithm will be the visible satellites which are used in the calculation of the DoP values.

Following the discovery of which satellites are visible for receiver, the coordinates of these satellites will be transformed from ECEF to ENU in order to compute the DoP values to a local topocentric system with its center in the chosen local reference point, thus to refer the these values to the Earth's surface (Yuen, 2009). The DoP values will be computed using an algorithm which uses the ENU coordinates of the local reference point and the satellites.

After implementing the mission planning simulation, the results (DoP values, number of visible satellites, azimuth and elevation angles) will be compared with the data acquired from a real-life case in order to asses the performance of the simulation.

---

<sup>1</sup><https://dirsig.cis.rit.edu/docs/new/coordinates.html>

### 3.2. Reading the satellite data into ECEF coordinate system

These steps and the formulas are taken from [Yuen \(2009\)](#) when calculating the satellite ECEF coordinates by using the GPS almanac. These formulas present the transformation of the orbital elements of the GPS satellites in ECEF coordinates.

The GPS almanac file contains data such as:

- Number of records
- GPS Week Number
- GPS Time of Applicability
- PRN Number
- Eccentricity
- Inclination offset
- Rate of Right Ascension
- Square root of Semi-Major Axis
- Longitude of Orbital Plane
- Argument of Perigee
- Mean anomaly at reference Time
- Satellite Health

Before extracting the information, the date and time need to be specified and represented in the same format as the GPS week and the GPS number of seconds in the almanac file. Since this is difficult to do using the Gregorian calendar, the dates have to be represented in the Julian calendar. [Yuen](#) uses the next equation for conversion between the calendars:

$$JD = 367 * Y - \left(\frac{7}{4} * \left(Y + \left(\frac{M+9}{12}\right)\right)\right) - \left(\frac{3}{4} * \left(Y + \frac{M-9}{7} * 100\right) + 1\right) + \left(\frac{275 * M}{9}\right) + D + 1721028.5 + \frac{hr}{24} + \frac{min}{1440} + \frac{sec}{86400} - \frac{timezone + daylightsaving}{24} + \frac{leapsec}{86400} \quad (3.1)$$

The Julian date for the start of the GPS constellation, 6 January 1980 is therefore 2444244.5 and will be used in equations 3.2 and 3.3.

$$Number_{week} = \left(\frac{JD - 2444244.5}{7}\right) \quad (3.2)$$

$$Number_{second} = (JD - 2444244.5 - 7 * Number_{week}) * 86400 \quad (3.3)$$

The JD represents the conversion from Gregorian to Julian calendar.

### 3. Methodology

The GPS week is then checked if it is more than 1024 and to avoid a rollover error<sup>2</sup>. A rollover happens every 1024 weeks (19-20 years) since the date of the start of the GPS constellation, 6 January 1980. Therefore, to avoid this error, 1024 will be extracted from the values of the GPS week until it has a smaller value than 1024.

Before starting to apply the formulas, these constants had to be taken into consideration:

$$\mu = 3.986005 * 10^{14} m^3/s^2 - \text{Earth's universal gravitational parameter}$$

$$\Omega_e = 7.2921151467 * 10^{-5} rad/s - \text{Earth's rotation rate}$$

The following formulas help in computing the satellite positions into ECEF:

- Mean motion

$$n_0 = \sqrt{\frac{\mu}{A^3}} \quad (3.4)$$

- Time since Time of applicability

$$t_k = (Number_{week} - wn) * 604800 + (Number_{seconds} - ToA) \quad (3.5)$$

- Kepler equation for eccentric anomaly

$$E_k = M_k + e * \sin E_k \quad (3.6)$$

- True anomaly

$$v_k = t g^{-1} \frac{(\sqrt{1 - e^2} * \sin E_k) / (1 - e * \cos E_k)}{(\cos E_k - e) / (1 - e * \cos E_k)} \quad (3.7)$$

- Eccentric anomaly

$$E_k = \cos^{-1} \frac{e + \cos v_k}{1 + e * \cos v_k} \quad (3.8)$$

- Corrected argument of latitude

$$u_k = v_k + \omega \quad (3.9)$$

- Corrected radius

$$r_k = A(1 - e * \cos E_k) \quad (3.10)$$

- Corrected inclination

$$i_k = i_0 + \delta * i_k \quad (3.11)$$

---

<sup>2</sup><https://en.wikipedia.org/wiki/GPS-week-number-rollover>

- Satellite position in orbital plane - X axis

$$x'_k = r_k * \cos u_k \quad (3.12)$$

- Satellite position in orbital plane - Y axis

$$y'_k = r_k * \sin u_k \quad (3.13)$$

- Corrected longitude of the ascending node

$$\Omega_k = \Omega_0 + (\Omega \cdot - \Omega_e) t_k - \Omega_e * T_0 A \quad (3.14)$$

- Satellite coordinate in ECEF

$$x_k = x'_k * \cos \Omega_k - y'_k * \cos y_k * \sin \Omega_k \quad (3.15)$$

$$y_k = x'_k * \cos \Omega_k - y'_k * \cos y_k * \cos \Omega_k \quad (3.16)$$

$$z_k = y'_k * \sin i_k \quad (3.17)$$

### 3.3. Position of the ground antenna

The coordinates of the receiver antenna are the ones displayed by the GNSS application in latitude, longitude and altitude. The altitude presented by the GNSS receiver application is the height above the mean sea level in meters. These coordinates have to be converted from geodetic WGS84 to ECEF for determining the satellite direction, RDNAP to determine the obstacles and finally to ENU in order to transform the satellite coordinates from ECEF to ENU.

To convert the coordinates from geodetic coordinates to ECEF, an intermediary calculation will be used:

Knowing the constants:

$$a = 6378137m \text{ -Earth's Semi Major Axis}$$

$$e = 8.181919084266 * 10^{-2} \text{ -Earth's First Eccentricity}$$

Prime vertical Radius of Curvature:

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \quad (3.18)$$

The formulas for coordinate conversion are:

$$x = (N + alt) \cos \phi \cos \lambda \quad (3.19)$$

### 3. Methodology

$$y = (N + alt) \cos \phi \sin \lambda \quad (3.20)$$

$$z = ((1 - e^2)N + alt) \sin \phi \quad (3.21)$$

The letter N in equation 3.18 is the Prime vertical of the radius curvature and will be used in further sections, not the Geoid height as presented in the figure 3.2.

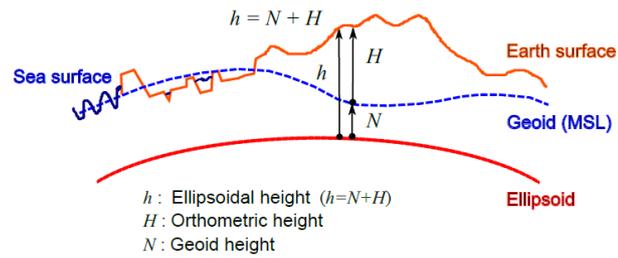


Figure 3.2.: Relation between ellipsoid and geoid (van der Marel, 2020)

### 3.3. Position of the ground antenna

In this figure 3.2,  $N$  represents the height from the ellipsoid to the mean sea level or the geoid shape. Adding further the orthometric height  $H$  from the geoid until the Earth's surface to  $N$  results in the ellipsoidal height which is used in the WGS84 coordinate system.

However, before applying equations 3.19 -3.21, the conversion of the altitude has to be taken into consideration. The altitude of the point needs to be transformed from orthometric to WGS84 ellipsoidal height for the transformation to ECEF to present the correct values. This is done by inspecting the figure 3.3 and adding the geoidal separation value which covers the area of study.

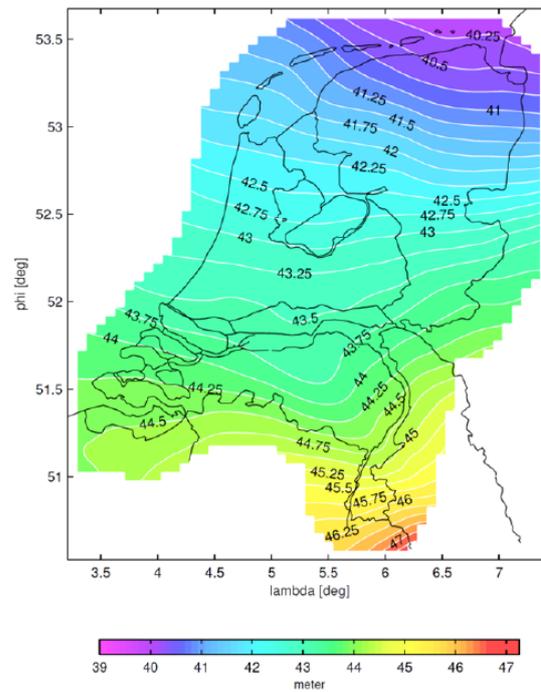


Figure 3.3.: Geoid Height Map for the Netherlands (Bruijne et al., 2005 - in van der Marel, 2020)

### 3. Methodology

Furthermore, in some cases, the height of the receiver should take into consideration also its height above the ground at which the receiver was held during the measurements.

## 3.4. Determining the receiver-satellite direction

As the real satellites are too far from the boundary of the Netherlands, intermediary "ghost" satellites had to be created between the receiver and the real satellites to keep the analysis inside the country. Another reason for the creation of these "ghost" satellites was to not use the large distances between the satellites and the 3D point cloud data which might be time-consuming to compute. The intermediary satellite coordinates were chosen to be 500 metres starting from the receiver towards the satellite.

Below are the formulas used to compute the  $(x,y,z)$  coordinates of the point which lies between the satellite and the receiver 3D line, 500 meters from the position of the receiver:

$$direction = P_{satellite} - P_{receiver} \quad (3.22)$$

$$magnitude = |direction| \quad (3.23)$$

$$unit_{vector} = \frac{direction}{magnitude} \quad (3.24)$$

$$P_{ghost} = P_{receiver} + 500 * unit_{vector} \quad (3.25)$$

## 3.5. CRS conversion pipeline

For the visibility analysis to work, all the coordinates had to be converted to the same coordinate system. As the larger amount of data, the AHN4 point cloud data was in the coordinate system for the Netherlands, the satellite coordinates had therefore to be converted from ECEF to RDNAP.

This was done inspecting the official RDNAPTRANS2018 document, written by [Lesparre et al.](#), which explains the steps of the conversion and uses diagrams for representation.

It has to be mentioned that the elevation data presented in the AHN4 point clouds is not the ellipsoidal height, but the NAP height which is the equivalent of the orthometric height. The difference is presented and explained in figure 3.2.

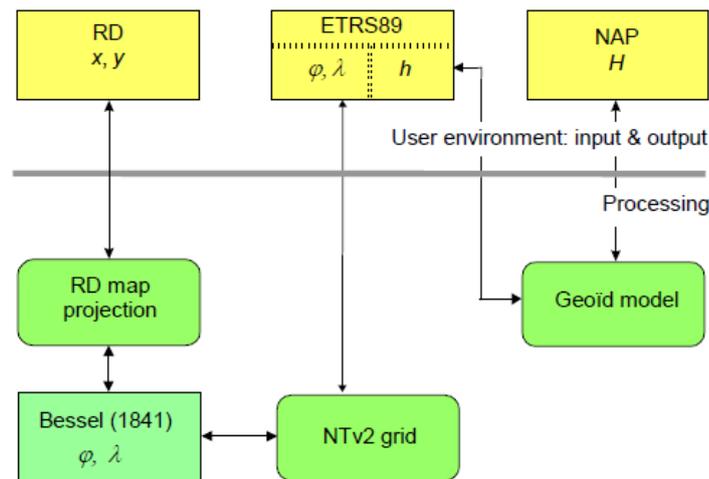


Figure 3.4.: NTV2 transformation version for RDNAPTRANS2018 (van der Marel, 2020)

### 3.6. Visibility algorithm

To determine which satellites are visible in the environment of the receiver represented by the 3D point cloud data, a visibility algorithm was implemented which filters the point cloud file. The visibility algorithm consists of two parts: a 2D only algorithm followed by a 3D algorithm.

To avoid working satellites which are below the Earth, the "ghost" satellites are tested if they are above the receiver's antenna.

Due to the large amount of time that one AHN4 point cloud needs to be processed, a 2D algorithm, working only on the XY-plane was created. This part of the visibility algorithm implies filtering the point cloud using bounding boxes to reduce from its size and use only the areas necessary for the analysis. This algorithm implies splitting the XY-projection of the LoS into segments by calculating points along the line until a certain threshold was met and then creating a rectangle which contains that segment as its diagonal, returning its four corners. For the segments and the boxes to be implemented correctly, the orientation of each satellite towards the antenna had to be calculated.

### 3. Methodology

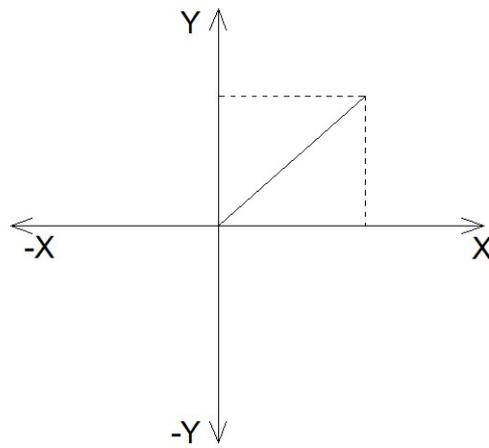


Figure 3.5.: Computing the orientation of the boxes using the differences on the XY axis

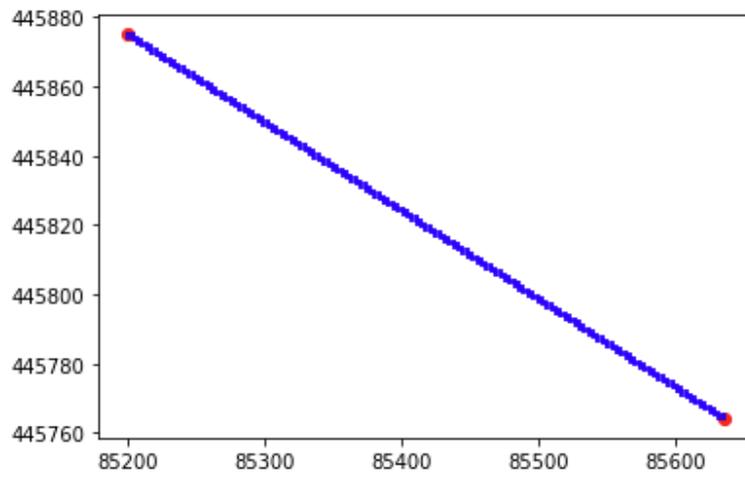


Figure 3.6.: Rectangular bounding boxes along the ray between the satellite and the receiver (X coordinate on the horizontal axis and Y coordinate on the vertical axis)

Following the rectangles on the antenna-satellite XY-projection line computed, to avoid the loss of portions from the point cloud which might contain important data for the analysis, two square bounding boxes are introduced into the filtering algorithm by mentioning the length of the edge. These boxes are drawn for each intermediary point computed on the line, it being their common corner and their direction perpendicular to that of the antenna-satellite line as seen in figure 3.7.

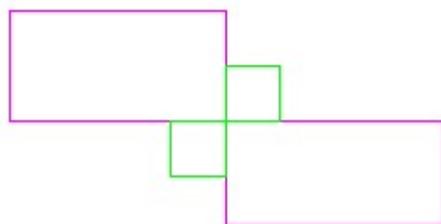


Figure 3.7.: Implementing the square boxes (green) compared to the rectangle boxes (purple)

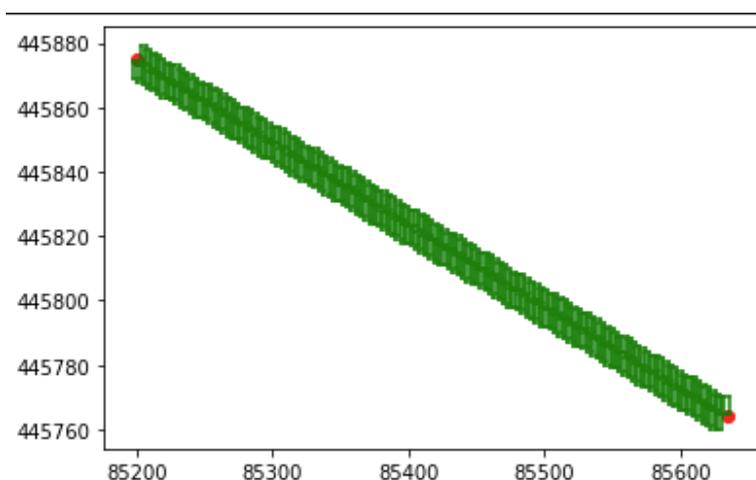


Figure 3.8.: Square bounding boxes on the perpendiculars of the satellite-receiver line (X coordinate on the horizontal axis and Y coordinate on the vertical axis)

If it happens that either the X or Y coordinates of the receiver and the satellite are too close, thus resulting in an almost vertical or horizontal line, no rectangular or square boxes would be created, but instead one big rectangle, with the length of the sides of the square boxes to one of the sides of said rectangle.

The 3D algorithm approximates the LoS with a cylinder shaped ray rather than a line in the geometrical aspect. This is because there is a possibility that a thin line which connects the satellite and the receiver would penetrate a point cloud object without actually hitting any of the points of the object. This is also a filtering step as it keeps only the 3D points which fall inside the cylinder of a given radius.

### 3. Methodology

For the 3D algorithm <sup>3</sup>, a cylinder of an inserted radius is projected between the ground receiver and the satellite point. Each point of the merged point cloud is tested if it is inside the cylinder or not. If the number of points inside the cylinder exceeds a given limit, the line of sight is considered blocked.

Point 1 (receiver) and point 2 (satellite) are the points in which the cylinder is projected.

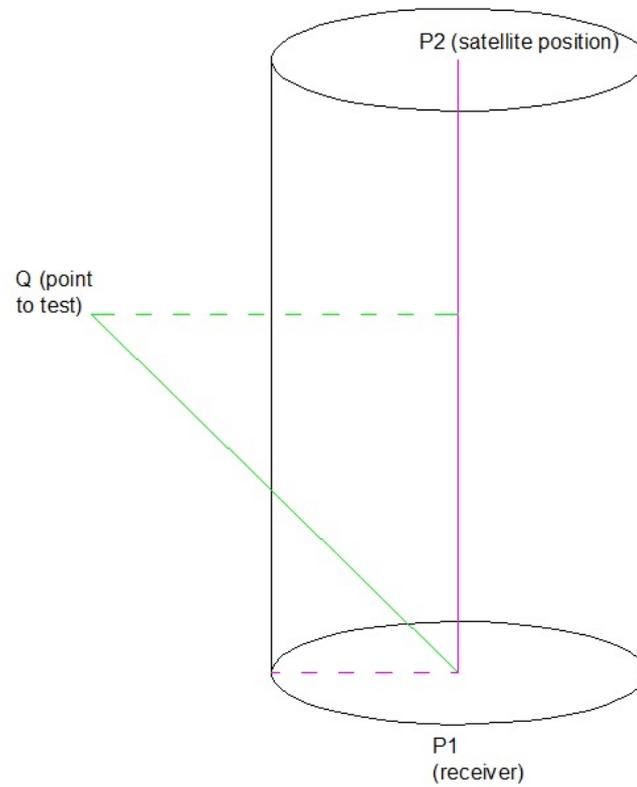


Figure 3.9.: Cylinder between P1 (receiver) and P2 (satellite), with Q being the point to be tested - obtaining vectors P1Q and P1P2

---

<sup>3</sup><https://www.flipcode.com/archives/Fast-Point-In-Cylinder-Test.shtml>

$$dX = x_{P2} - x_{P1} \quad (3.26)$$

$$dY = y_{P2} - y_{P1} \quad (3.27)$$

$$dZ = z_{P2} - z_{P1} \quad (3.28)$$

$$length_{axis} = \sqrt{dX^2 + dY^2 + dZ^2} \quad (3.29)$$

The same calculations are applied for the differences between a point belonging to the point cloud and the bottom point of the cylinder.

$$dX_{P1Q} = x_Q - x_{P1} \quad (3.30)$$

$$dY_{P1Q} = y_Q - y_{P1} \quad (3.31)$$

$$dZ_{P1Q} = z_Q - z_{P1} \quad (3.32)$$

The dot product is calculated between the cylinder axis and the vector between the test point and the bottom point of the cylinder.

$$dot = dX_{P1Q} * dX_{P1P2} + dY_{P1Q} * dY_{P1P2} + dZ_{P1Q} * dZ_{P1P2} \quad (3.33)$$

Knowing the basic trigonometric formulas, with  $\alpha$  being the angle between the two vectors:

$$\sin(\alpha) = \frac{sideopposite}{hypotenuse} \quad (3.34)$$

$$\sin^2(\alpha) + \cos^2(\alpha) = 1 \quad (3.35)$$

$$a \cdot b = \cos(\alpha) * |a| * |b| \quad (3.36)$$

### 3. Methodology

$$a \cdot b = dX_{ab}^2 + dY_{ab}^2 + dZ_{ab}^2 \quad (3.37)$$

It is also known that the radius of the circular base of the cylinder is perpendicular on the axis of the cylinder and using equations 3.34 - 3.37, the distance from the point to the axis of the cylinder, known as  $d_{axis}$  will be calculated using the perpendicularity formula:

$$d_{axis} = \sin(\alpha) * |P1Q| \quad (3.38)$$

$$d_{axis}^2 = \sin^2(\alpha) * |P1Q|^2 \quad (3.39)$$

$$d_{axis}^2 = (1 - \cos^2(\alpha)) * |P1Q|^2 \quad (3.40)$$

$$d_{axis}^2 = \left(1 - \frac{P\vec{1}Q^2 * P\vec{1}P2^2}{|P1P2|^2 * |P1Q|^2}\right) * |P1Q|^2 \quad (3.41)$$

$$d_{axis}^2 = |P1Q|^2 - \frac{P\vec{1}Q^2 * P\vec{1}P2^2}{|P1P2|^2} \quad (3.42)$$

Resulting finally in:

$$d_{axis}^2 = (dX_{P1Q}^2 + dY_{P1Q}^2 + dZ_{P1Q}^2) - \frac{P\vec{1}P2^2 * P\vec{1}Q^2}{length_{axis}^2} \quad (3.43)$$

This computed distance from point Q towards the axis is parallel with the radius of the base and thus its square is tested if it is larger in length than the radius of the base. If not, then the point is inside the cylinder.

The squared distance is then tested if it is larger than the square of the radius.

### 3.7. Conversion from ECEF to ENU

As explained in section 3.1 and used by both Yuen and Dana <sup>4</sup>, in order to obtain the horizontal DoP value referred to the Earth's surface, the true satellite coordinates which were considered visible by the simulation need to be converted to East-North-Up so they are relative to a local reference point on Earth's surface, in this case, the ground receiver.

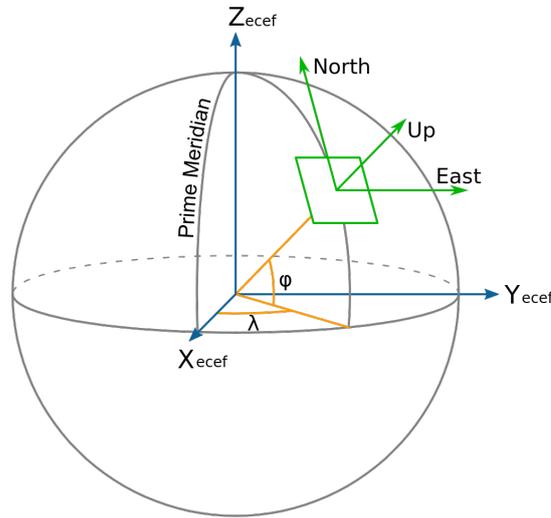


Figure 3.10.: ENU vs ECEF ([https://en.wikipedia.org/wiki/Local\\_tangent\\_plane\\_coordinates](https://en.wikipedia.org/wiki/Local_tangent_plane_coordinates))

It has to be mentioned that for the local reference point  $(x_l, y_l, z_l)$ , which will be used in the conversion, the formulas will be the same as the ones used to convert the position of the receiver into ECEF presented in section 3.2 but with the altitude equal to 0 to position it on the Earth's surface.

Using equation 3.18, the conversion of local reference point from geodetic to ECEF coordinates is:

$$x = (N + sealevel) \cos \phi \cos \lambda \quad (3.44)$$

$$y = (N + sealevel) \cos \phi \sin \lambda \quad (3.45)$$

$$z = ((1 - e^2)N + sealevel) \sin \phi \quad (3.46)$$

Conversion of satellite points from ECEF to ENU:

$$East = -(x_{sat} - x_l) \sin \phi + (y_{sat} - y_l) \cos \lambda \quad (3.47)$$

$$North = -(x_{sat} - x_l) \sin \phi * \cos \lambda - (y_{sat} - y_l) \sin \phi * \sin \lambda + (z_{sat} - z_l) \cos \phi \quad (3.48)$$

<sup>4</sup>[https://foote.geography.uconn.edu/gcraft/notes/gps/gps\\_f.html](https://foote.geography.uconn.edu/gcraft/notes/gps/gps_f.html)

### 3. Methodology

$$Up = (x_{sat} - x_l) \cos \phi * \cos \lambda + (y_{sat} - y_l) \cos \phi * \sin \lambda + (z_{sat} - z_l) \sin \phi \quad (3.49)$$

where sat = satellite and l = local reference point

### 3.8. DoP calculation algorithm

To calculate the DoP values, the distances between the receiver and the satellites had to be calculated

$$\rho = \sqrt{(East_{sat} - East_r)^2 + (North_{sat} - North_r)^2 + (Up_{sat} - Up_r)^2} \quad (3.50)$$

where sat = satellite and r = receiver

The directional derivatives are then calculated and inserted into a nx4 matrix, with n being the number of visible GPS satellites.

$$D_{E_{sat}} = \frac{East_{sat} - East_r}{\rho_{sat}} \quad (3.51)$$

$$D_{N_{sat}} = \frac{North_{sat} - North_r}{\rho_{sat}} \quad (3.52)$$

$$D_{U_{sat}} = \frac{Up_{sat} - Up_r}{\rho_{sat}} \quad (3.53)$$

$$D_t = -1 \quad (3.54)$$

$$D = \begin{pmatrix} D_{E_1} & D_{N_1} & D_{U_1} & D_{t_1} \\ D_{E_2} & D_{N_2} & D_{U_2} & D_{t_2} \\ \dots & \dots & \dots & \dots \\ D_{E_n} & D_{N_n} & D_{U_n} & D_{t_n} \end{pmatrix} \quad (3.55)$$

The DoP values can be discovered as:

$$(D^T * D^{-1}) = \begin{pmatrix} XDOP^2 & & & \\ & YDOP^2 & & \\ & & VDOP^2 & \\ & & & TDOP^2 \end{pmatrix} \quad (3.56)$$

The other DoP values can be calculated as:

- Horizontal DoP

$$HDoP = \sqrt{XDoP^2 + YDoP^2} \quad (3.57)$$

- Position DoP

$$PDoP = \sqrt{XDoP^2 + YDoP^2 + VDoP^2} \quad (3.58)$$

- Geometrical DoP

$$GDoP = \sqrt{XDoP^2 + YDoP^2 + VDoP^2 + TDoP^2} \quad (3.59)$$

Considering the figures 1.3 and 1.4 from the first chapter, a low value closer to 0 for GDOP represents good satellite geometry. I will give an example of how such a matrix works for a good GDOP.

$$(D^T * D^{-1}) = \begin{pmatrix} 0.30128689 & -0.01866862 & 0.09499076 & 0.04345208 \\ -0.01866862 & 0.17753978 & 0.02034522 & -0.00557301 \\ 0.09499076 & 0.02034522 & 0.62688042 & 0.24660771 \\ 0.04345208 & -0.00557301 & 0.24660771 & 0.17504643 \end{pmatrix} \quad (3.60)$$

Using the elements on the diagonal and respecting equation 3.59, the value of GDOP would be:

$$GDOP = 1.1317038$$

And for a poor GDOP:

$$(D^T * D^{-1}) = \begin{pmatrix} 0.53391574 & 0.10571521 & -0.65037274 & -0.47113019 \\ 0.10571521 & 1.09190074 & -1.11100618 & -0.98092091 \\ -0.65037274 & -1.11100618 & 3.90319909 & 2.82757103 \\ -0.47113019 & -0.98092091 & 2.82757103 & 2.21410914 \end{pmatrix} \quad (3.61)$$

And the value of the GDOP would be:

$$GDOP = 2.7826$$

### 3.9. Azimuth and Elevation Angles

As a further validation of the visibility simulation after computing the DoP values and discovering the number of satellites in view, the azimuth and elevation angles are calculated between the position of the receiver and every in-view satellite position. These values would be then compared to the ones from the NMEA strings given by the receiver

The following formulas are calculated using ENU coordinates and are taken from [van der Marel, 2020](#).

Azimuth formula:

$$\alpha = \arctan \frac{dE}{dN} * \frac{180}{\pi} = \arctan \frac{dX}{dY} * \frac{180}{\pi} \quad (3.62)$$

Elevation formula:

$$\zeta = \arctan \frac{dU}{\sqrt{dE^2 + dN^2}} * \frac{180}{\pi} \quad (3.63)$$

The elevation angle can also be calculated using coordinates from local coordinate system as [Lu and Han](#) also used:

$$\zeta = \arctan \frac{H}{D} * \frac{180}{\pi} \quad (3.64)$$

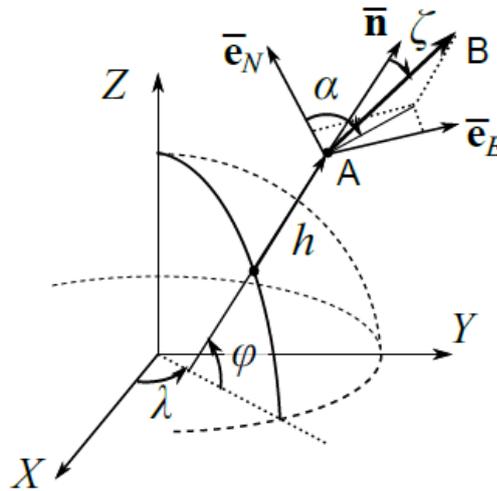


Figure 3.11.: Azimuth and elevation angles computed from point A to point B ([van der Marel, 2020](#))

### 3.10. GNSS data measurements

The real-time GNSS data helps in validating the results given by the simulation through comparison.

The data received by the ground antenna during the data acquisition operation was organised into NMEA strings. NMEA is a standard format supported by GPS manufacturers in which information is transmitted<sup>5</sup>. From these strings, the data from the following ones was used for the research:

```
$GPGSV,3,1,11,05,30,303,41,06,18,199,41,07,68,116,42,09,38,080,43,1*6A
$GPGSV,3,2,11,11,38,232,40,13,14,255,36,20,63,278,43,29,05,313,28,1*61
$GPGSV,3,3,11,30,52,184,49,04,06,078,,16,10,028,,1*51
$GNGSA,A,3,05,06,07,09,11,13,20,29,30,,,,,0.9,0.6,0.8,1*33
$GNGGA,093341.55,5159.870061,N,00422.641143,E,1,12,0.6,1.7,M,47.0,M,,*7F|
```

Figure 3.12.: NMEA strings to be analysed

More details can be found on this website<sup>6</sup> about the NMEA strings.

- GNGGA - gives the UTC time and the position in decimal minutes
- GNGSA - gives the DoP values and the active satellites used for the calculation
- GPGSV - gives information about the satellites in view

<sup>5</sup>GPS World - What is NMEA Data?

<sup>6</sup><https://anavs.com/knowledgebase/nmea-format/>

### 3. Methodology

Talker ID by	XX	All talker IDs usable
Sentence ID	GGA	
UTC of position	093341.55	Fixed length 2 digits after dot
Latitude	5159.870061	Fixed length 4 digits before and 7 after dot
Hemisphere of latitude	N	N if value is positive
Longitude	00422.641143	Fixed length 5 digits before and 7 after dot
Hemisphere of longitude	E	E is longitude is positive
GPS quality indicator	1	GPS fix valid
Number of used satellites for positioning	12	Fixed length 01 for single digits
HDOP	0.6	Variable/fixed length 1 digit after dot, variable before
Altitude geoid height	1.7	Variable/fixed length 2 digits after dot, variable before
Unit of altitude	M	
Geoidal separation	47	Variable/fixed length 2 digits after dot, variable before
Unit of altitude	M	
Age of differential data		
Checksum for error	*7F	

Table 3.1.: GNGGA string explanation

Talker ID by	XX	All talker IDs usable
Sentence ID	GSA	
MODE	A	Automatic
MODE	3	3D fix mode
ID Number of satellites	05, 06, 07, 09, 11, 13, 20, 29, 30	GPS PRN between 1-32
PDOP	0.9	Fixed length 2 digits before and after dot
HDOP	0.6	Fixed length 2 digits before and after dot
VDOP	0.8	Fixed length 2 digits before and after dot
Checksum for error	1*33	

Table 3.2.: GNGSA string explanation

Talker ID by	XX	All talker IDs usable
Sentence ID	GSV	
Total number of messages	3	1-9
Message number	1	1-9
Total number of satellites in view	11	Fixed length: 01 for single digit
Satellite ID number	05, 06, 07, 09	For GPS, PRN is 01-32
Elevation	30, 18, 68, 38	Fixed length: 00 for 0° elevation; between 0-90
Azimuth	303, 199, 116, 080	Fixed length: 000 for 0° azimuth; between 0-360
Signal-to-Noise ratio	41, 41, 42, 43	Fixed length: 05 for 5 db/Hz 0-99

Table 3.3.: GPGSV string explanation



## 4. Implementation and Results

### 4.1. Resources

The simulation for this research only uses the GPS satellite constellation <sup>1</sup>.

The visibility simulation as well as the coordinate conversion pipeline were both written in python programming language. The most relevant libraries used were:

- pyproj for the coordinate system conversion - a coordinate transformer from pipeline was implemented
- matplotlib for visualising the initial data and the results of the visibility algorithms
- laspy for point cloud operations - reading the contents of a point cloud
- numpy for organising the data - organising the points into arrays containing their coordinates
- math for mathematical operations

Due to the huge amount of point cloud data and the large amount of time it needs to complete the visibility methods, the LAStools software package was used to filter the point cloud data based on the boundaries given by the 2D visibility algorithm.

QGIS software was used to visualise the Well-Known Text <sup>2</sup> layer version of the bounding boxes to understand how setting up the threshold influenced the filtering results. QGIS was also used for visualising the filtered point clouds at certain stages of the filtering algorithm. Microsoft Excel was used to create graphs for statistical analysis using the data from the simulation.

#### 4.1.1. AHN4 Point Clouds

The area of study is Delft. An AHN4 point cloud representing a part of Delft has been selected to be used as a representation of 3D data <sup>3</sup>

The format of the AHN point clouds is .LAZ. This point cloud format represents a compressed version of the format .LAS (LASER), which is the most used standard for point cloud data. .LAZ and .LAS are almost identical regarding the record format, with .LAZ being 10 times more compact than .LAS. The AHN3 set of point clouds uses the .LAZ format also with the classifications given in .LAS (Ledoux et al., 2020).

---

<sup>1</sup><https://www.navcen.uscg.gov/gps-nanus-almanacs-opsadvisories-sof>

<sup>2</sup><https://en.wikipedia.org/wiki/Well-known-text-representation-of-geometry>

<sup>3</sup><https://geotiles.nl/>

#### 4. Implementation and Results

Field	Format	Length (bits)	Description
X	int	32	X coordinate.
Y	int	32	Y coordinate.
Z	int	32	Z coordinate.
Intensity	unsigned int	16	The pulse return amplitude.
Return number	unsigned int	3	The total pulse return number for a given output pulse.
Number of returns	unsigned int	3	Total number of returns for a given pulse
Scan Direction Flag	boolean	1	Denotes the direction at which the scanner mirror was travelling at the time of the output pulse. A bit value of 1 is a positive scan direction, and a bit value of 0 is a negative scan direction (where positive scan direction is a scan moving from the left side of the in-track direction to the right side and negative the opposite).
Edge of Flight Line	boolean	1	Has a value of 1 only when the point is at the end of a scan. It is the last point on a given scan line before it changes direction.
Classification	unsigned int	5	Classification code
Scan Angle Rank	int	4	The angle at which the laser pulse was output from the scanner including the roll of the aircraft.
User Data	unsigned int	4	May be used at the user's discretion.
Point Source ID	unsigned int	8	Indicates the file from which this point originated. Non-zero if this point was copied from another file.

Figure 4.1.: LAS Point Data Record format table (Ledoux et al., 2021)

Code	Meaning
0	never classified
1	unclassified
2	ground
3	low vegetation
4	medium vegetation
5	high vegetation
6	building
7	low point (noise)
8	reserved
9	water

Figure 4.2.: The first 10 classes of points from AHN (Ledoux et al., 2021)

#### 4.1.2. NMEA Tools

For the GNSS data acquisition operation, the personal android phone was used, a Samsung Galaxy S20 FE 5G. The GNSS gathering tool is NMEA Tools Pro <sup>4</sup> by Peter Ho.

It has to be mentioned that this application returns results based on each smartphone's built-in receiver. While for some smartphone brands it will return the "-GGA" and the "-GSA" strings based on which satellites from which constellation it receives signal at that moment in time, this was not the case for the personal Samsung smartphone. In this case, it returned

<sup>4</sup>[https://play.google.com/store/apps/details?id=com.peterhohsy.nmeapaserprohl=en\\_US](https://play.google.com/store/apps/details?id=com.peterhohsy.nmeapaserprohl=en_US)

the strings GNGGA and GNGSA, meaning that the signal was generated from satellites from multiple constellations.

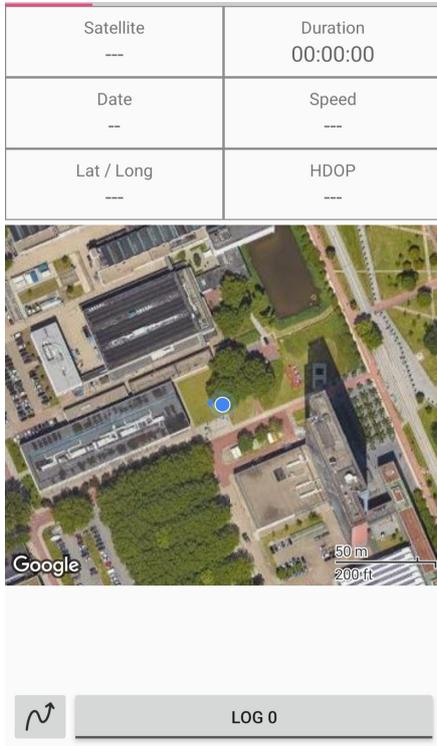


Figure 4.3.: Log screen

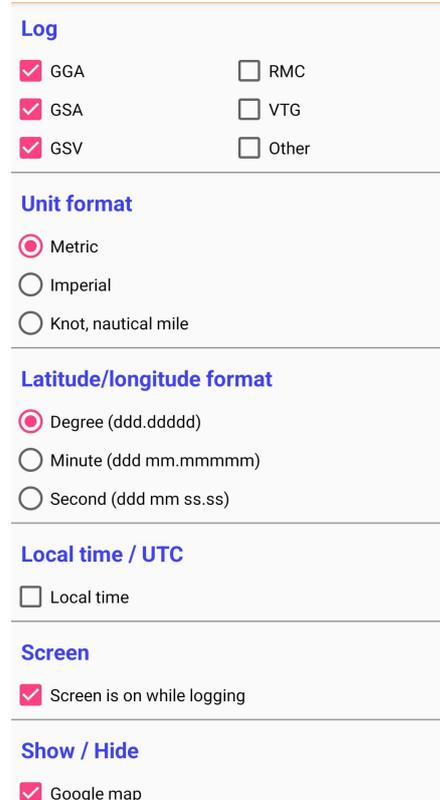


Figure 4.4.: Settings for the strings

As the GNGSA line was returning the satellites used for the calculation of the DoP values, the lines which presented satellited recorded with the numbers between 1 and 32 were chosen as those represented the satellites of the GPS constellation.

### 4.1.3. CloudCompare

CloudCompare is an open-source processing software for point clouds and meshes.<sup>5</sup>

CloudCompare was the main software used for 3D visualisation, as it could display and work with larger amounts of data better than QGIS and matplotlib could and the point clouds obtained from the filtering operations could be visualised in better detail. Additionally, the software can also work with various point cloud formats.

The receiver-satellite cylinder shaped filtering is a 3D algorithm and the Z coordinate of the points between the satellite and the receiver finally decides if the point falls inside the cylinder or not, as the Z coordinate increases from that of the receiver to that of the satellite position. To understand and visualise how the 3D cylinder algorithm works, the RDNAP

<sup>5</sup><https://www.danielgm.net/cc/>

#### 4. Implementation and Results

coordinates of the receiver and the satellite positions are written in a text file which can be introduced into the software and visualised in 3D. The software also presents data manipulation tools, such as creating polylines using the points. This particular tool is used manually insert the cylinder axis between the receiver the satellite position.



Figure 4.5.: Trace polyline tool (middle pictogram) and the Point Picking tool (left pictogram)

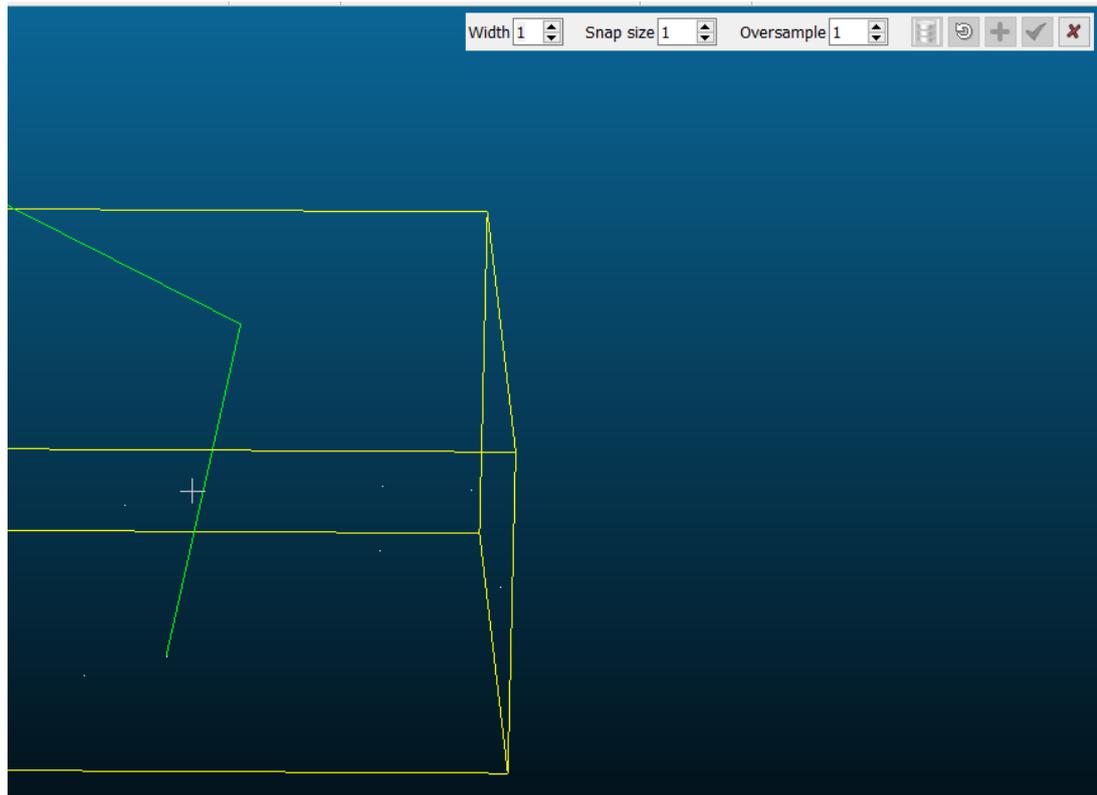


Figure 4.6.: Tracing a line between two points

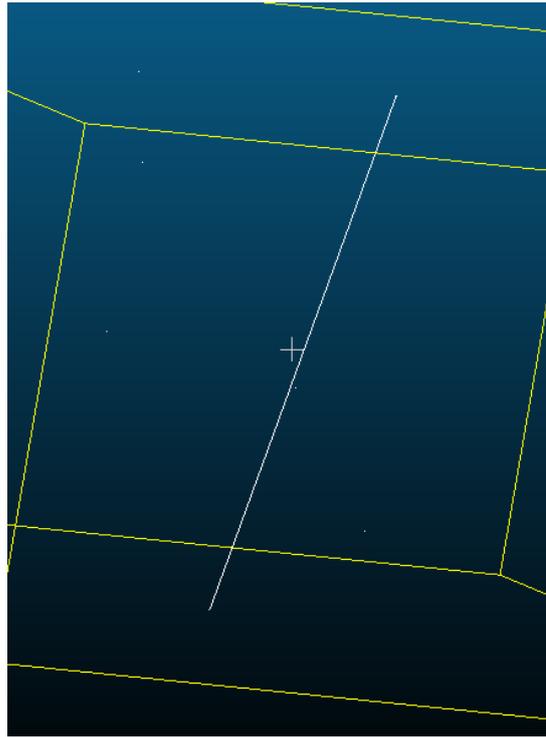


Figure 4.7.: Line that serves as the cylinder ax

In case the receiver presents large differences for the altitude due to unknown errors, the Point Picking tool can be used to analyse the coordinates of the points close to the receiver and it is easier to obtain a value for the Z coordinate.

## 4. Implementation and Results

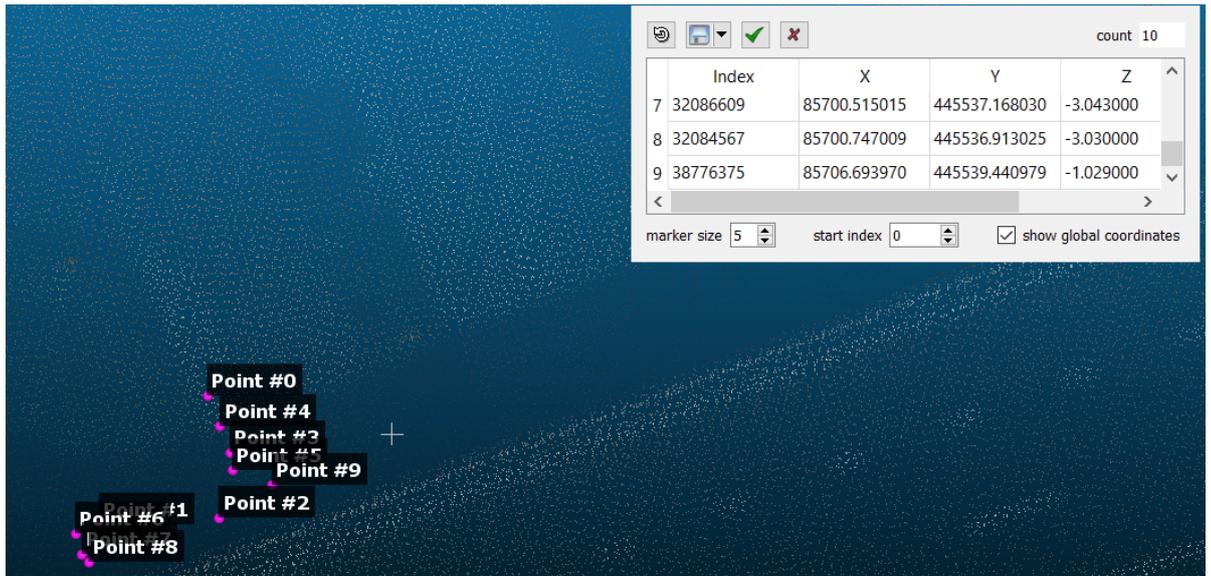


Figure 4.8.: Using the Point Picking tool of CloudCompare

## 4.2. Implementation

### 4.2.1. CRS conversion implementation

The implementation of the CRS conversion is one of the main steps to help in this analysis.

As the PROJ RDNAPTRANS CRS conversion does not work for locations outside the Netherlands, returning the value "inf", the "ghost" satellites were created as stated in the methodology chapter. These points are the ones on which the transformation is applied.

A custom transformer pipeline was created in python, with similar commands from the PROJ software presented in the RDNAPTRANS2018 manual.

Command	Comments
<code>cct -I -o output.txt +proj=pipeline</code>	<i>Set optional output file;</i>
<code>+step +proj=unitconvert +xy_in=deg +xy_out=rad</code>	<i>Convert units (Section 2.1 / 3.4);</i>
<code>+step +proj=axiswap +order=2,1</code>	<i>Swap for PROJ order longitude, latitude;</i>
<code>+step +proj=vgridshift +grids=nlgeo2018.tif</code>	<i>Transformation (Section 2.5 / 3.5);</i>
<code>+step +proj=hgridshift +inv +grids=rdtrans2018.tif</code>	<i>Correction (Section 2.3 / 3.2);</i>
<code>+step +proj=sterea +lat_0=52.156160556 +lon_0=5.387638889</code>	<i>Projection (Section 2.4 / 3.1);</i>
<code>+k=0.9999079 +x_0=155000 +y_0=463000 +ellps=bessel</code>	
<code>input.txt</code>	<i>Set optional input file</i>

Figure 4.9.: PROJ Commands, (Lesparre et al., 2018)

Lesparre, (2023) also presents a series of PROJ operations to convert coordinates from old historic maps of the Netherlands.

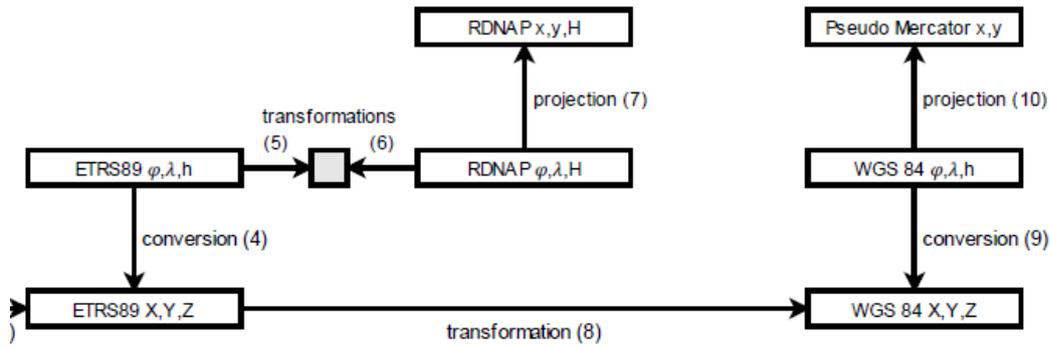


Figure 4.10.: Schema of transformations (Lesparre, 2023)

For the horizontal and vertical conversion to work, the transformation grids have to be downloaded and used in the coordinate transformation algorithm as seen in figure 4.9. <sup>6</sup>

Moving forward to which satellites can be considered in view for the simulation, a simple test was implemented. The condition of the test was that the Z coordinate of the "ghost" satellite in RDNAP had to be higher than the Z coordinate of the receiver in RDNAP in order to avoid satellites which could end up below the earth.

<sup>6</sup>[https://github.com/OSGeo/PROJ-data/tree/master/nl\\_nsgi](https://github.com/OSGeo/PROJ-data/tree/master/nl_nsgi)

## 4. Implementation and Results

### 4.2.2. Azimuth and Elevation computation

As a validation step to evaluate how close are the results of the simulation given by those of the smartphone application, the azimuth and elevation angles are computed and compared to those given in the GPGSV lines. The azimuth angles can be computed in both the ENU and RDNAP coordinates as in both cases it gives similar results to those from the real-life scenario. However, for the azimuth angles to be correct, the orientation of the antenna-satellite line has to be taken into consideration. As the azimuth is measured from the True North line clockwise to the antenna-satellite line, the order of the circle's quadrants are switched.

- First quadrant ( $x > 0, y > 0$ ) - add 0 degrees
- Second quadrant ( $x > 0, y < 0$ ) - add 180 degrees
- Third quadrant ( $x < 0, y < 0$ ) - add 180 degrees
- Fourth quadrant ( $x < 0, y > 0$ ) - add 360 degrees

The elevation angles are computed using RDNAP coordinates, because the RDNAP coordinate system uses the geoid as its reference surface which is a closer approximate to the Earth's surface rather than the ellipsoid.

### 4.2.3. Visibility algorithm

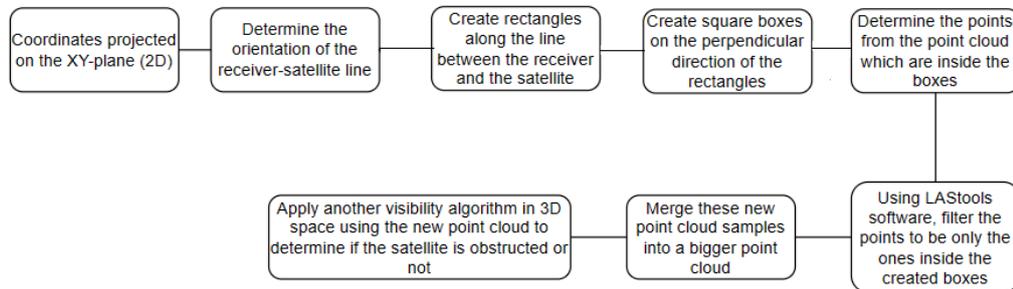


Figure 4.11.: Visibility methodology

To help in making the simulation manageable to run on the personal laptop, prior LAStools<sup>7</sup> point cloud operations were used to make the size of the point clouds smaller.

The first operation was filtering the point cloud data based on the mean-sea level height given by the GNSS tool and taking into consideration that the receiver is close to the ground, not on top of a building. This insures that only the necessary point cloud objects which can represent obstructions between the receiver and the satellite, are taken into consideration.

<sup>7</sup><https://lastools.github.io/>

**las2las -i pc.LAZ -drop-z-below [antenna-height] -o pc-height-filter.LAZ**

The second such operation was creating a lasindex .laz file. A lasindex file contains spatial indexing information for the data from the point cloud and can therefore speed up the searching and filtering operations. The type of spatial index used is the bounding box spatial indexing.

**lasindex -i pc-height-filter.LAZ**

After the creation of both the rectangular and the square bounding boxes, the coordinates of the corner of the bounding boxes are introduced into LASTools software command, las2las to filter the point cloud and keep only the points inside these.

**las2las -i pc-height-filter.laz -inside-rectangle [bounding-box-coordinates] -o small-bb-pc-satellite-number.laz**

The rectangular and square shaped small point clouds are then merged into a single point cloud representing one ray from the receiver towards the satellite and then the 3D visibility algorithm is applied as this is more time-saving than merging all the point cloud rays and then run the 3D algorithm.

This 2D algorithm helps the final 3D algorithm to use as input only the box-filtered point cloud and thus save computation time.

**lasmerge -i pc-folder/\*.laz -o pc-satellite-number.laz**

The small point clouds are then deleted so they do not occupy space.

The quantity of points inside the cylinder-shaped ray seems to depend mostly on user-input values for length of the sides of the square, smaller boxes and the radius of the base of the cylinder. The LASTools inside bounding box filtering query only allows for  $(x_{min}, y_{min})$  and  $(x_{max}, y_{max})$  pairs of coordinates, resulting always in a rectangle, with its sides parallel to the XY axis of the point cloud. The projection of the satellite-receiver cylinder on the XY will result in a parallelogram rotated by the value of the azimuth. The computation time the cylinder-ray filtering algorithm takes is also based on the number of 3D points that are to test. This number slightly increases with the length of the radius, which finally results in the computation time to increase if the chosen radius is larger. If the radius of the cylinder is to be change then the threshold which determines which satellites are visible or not based on the number of points which fall inside it also needs to be changed. These two geometrical instances, the chain of rectangular and square boxes and the parallelogram are different and will not overlap entirely. Thus, I have tried to link the length of the square boxes and the radius of the base of the cylinder so that the two sets of geometries would overlap as much as possible.

#### 4. Implementation and Results

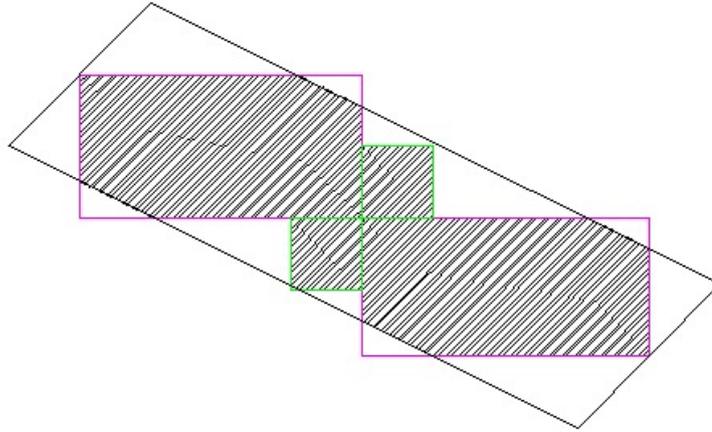


Figure 4.12.: Choosing the radius of the cylinder - the diameter (twice the radius - green line) of the cylinder in 2D (black parallelogram) is twice the diagonal of a square bounding box

Therefore,

$$r_{cylinder} = l * \sqrt{2} \quad (4.1)$$

where  $l$  represents the size for the square boxes.

It has to be mentioned that the number of square bounding boxes is given influenced by the length of the diagonal of the rectangular ones. A shorter diagonal results in the line between the receiver and the satellite to be split in more intermediary points around which the pair of square boxes is created. Thus, there are more las2las queries that take longer time. A longer length for the side of the square boxes also takes more time for the las2las queries and also takes more times as well for the 3D algorithm if equation 4.1 is to be taken into consideration.

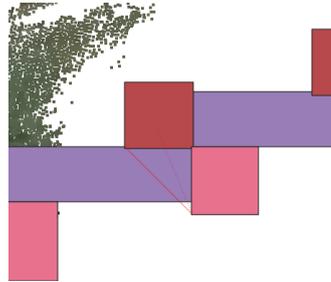


Figure 4.13.: 20 meters limit for the diagonal

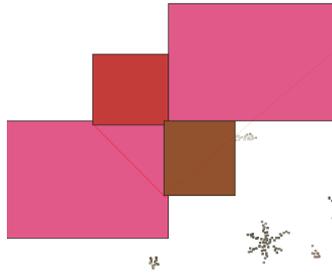


Figure 4.14.: 50 meters limit for the diagonal

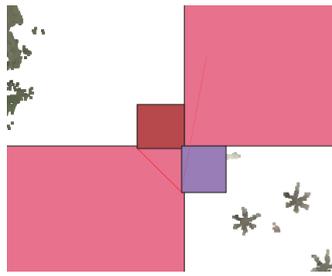


Figure 4.15.: 100 meters limit for the diagonal

The parameter that influences the computation time the most is the threshold for the rectangular bounding boxes on the LoS between the satellite and the receiver in the 2D algorithm. This can be observed in figures 4.13 - 4.15. Using a large threshold, for example a 150 - 220 meter threshold, results in less filtering queries and thus, together with the 3D algorithm, results in less than a minute to generate the number of points lying inside the cylinder, resulting in a waiting time of 5 minutes or less for 10-12 satellites together with the DoP calculations. If for example, the threshold was decreased to 20 meters, the whole visibility algorithm would take 3-4 minutes per satellite. The result of the computing time for each threshold is presented in figure 4.16. Although the threshold can still be increased, the waiting time seems to remain mostly the same, with a slight increase in time probably due to larger amount of point cloud data being generated after the filtering.

#### 4. Implementation and Results

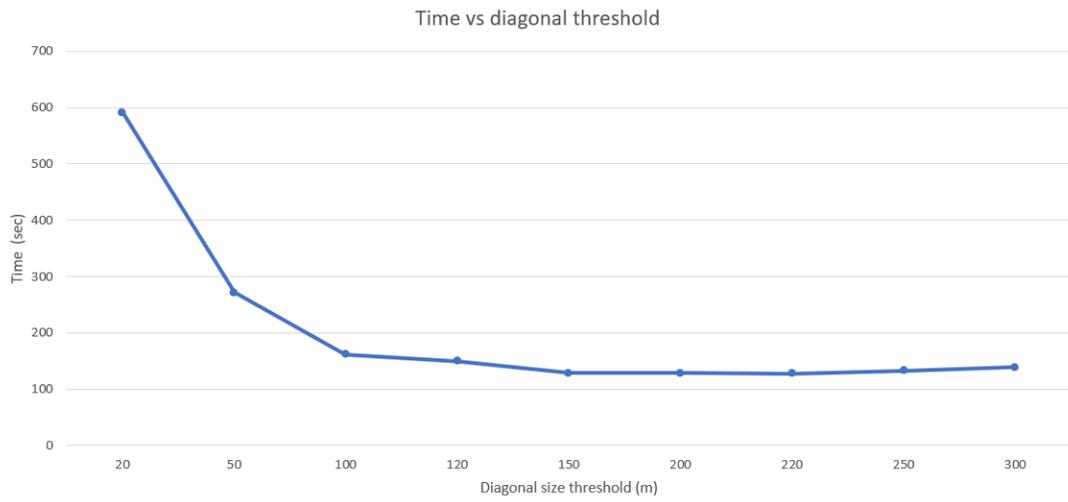


Figure 4.16.: Evaluation of the time used by the visibility algorithm based on the size of the diagonal threshold

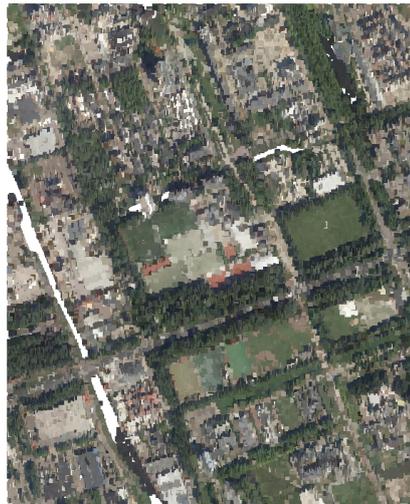


Figure 4.17.: Whole point cloud used as representation of the environment



Figure 4.18.: Point cloud after LASTools height filter

#### 4. Implementation and Results



Figure 4.19.: Point cloud after the 2D algorithm filter

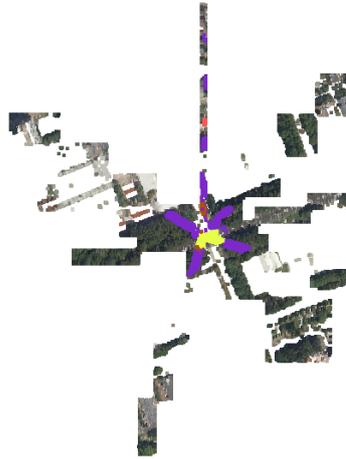


Figure 4.20.: The 2D algorithm and the 3D algorithm

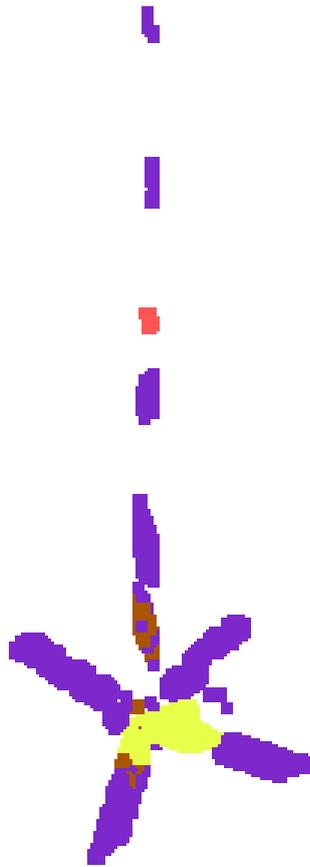


Figure 4.21.: Point cloud after the 3D algorithm filter

### 4.3. Results

*The code for the implementation can be found in my Github repository <sup>8</sup>.*

All the locations where the data acquisition operation was carried were on the TU Delft Campus.

The first point of measurement is on the Drebbelweg street, near the Computer Science faculty, on the 21st of November, 2023. I chose this location as its actual environment is formed out of the tall trees and the tallest building on the university's campus which could behave as obstructions in this analysis.

The second point is near the small lake, next to the Science Center of TU Delft which presents more open sky than the first location as there are no trees or very tall buildings. The day of the data acquisition operation was also the 21st of November.

The third point is close to the highway bridge, after the X building. The data acquisition took place on the 5th of December, 2023. Compared to the environment of the 1st location, the bridge presents a more defined obstruction object.

The fourth point is in the dense tree area near the Architecture faculty, on the 18th of December. The environment presents tall trees which could form an obstruction.

The threshold length chosen for the diagonal was therefore 220 meters due to having the lowest computation time. The length for the squares was 5 meters as to avoid any loss of important point cloud data. The radius of the cylinder is calculated using equation 4.1. For the number of points that fall inside the cylinder, a limit of 9000 points was chosen as the point cloud objects which fall inside the cylinder-shaped ray could be small but dense and can thus contain a few thousand points.

Presented in tables are the visible satellites for both the simulation and the real-life scenario which are used in calculating the DoP values.

---

<sup>8</sup><https://github.com/AIGheorg/Official-thesis-jupyternb>

### 4.3.1. First Test Site

Coordinates: 51.999275 degrees latitude, 4.372454 degrees longitude, 0.5 mean sea level height

Time: 09:00:00 - 09:20:00 UTC time (10:00:00 - 10:20:00 for the Netherlands), 21st of November, 2023

The first test site is located on Drebbelweg, next to the Computer Science Building.

The skyplot figures 4.22 - 4.29 show which satellites are visible from the region where the receiver is and indicate what their elevation and azimuth can be. These figures can be used as a validation for the simulation results. The skyplots are obtained using the Navmatix<sup>9</sup> online tool. It has to be said that as both locations are on the TUD campus and there is less than a 10 kilometer distance between them, the satellite-receiver configuration will not change much.

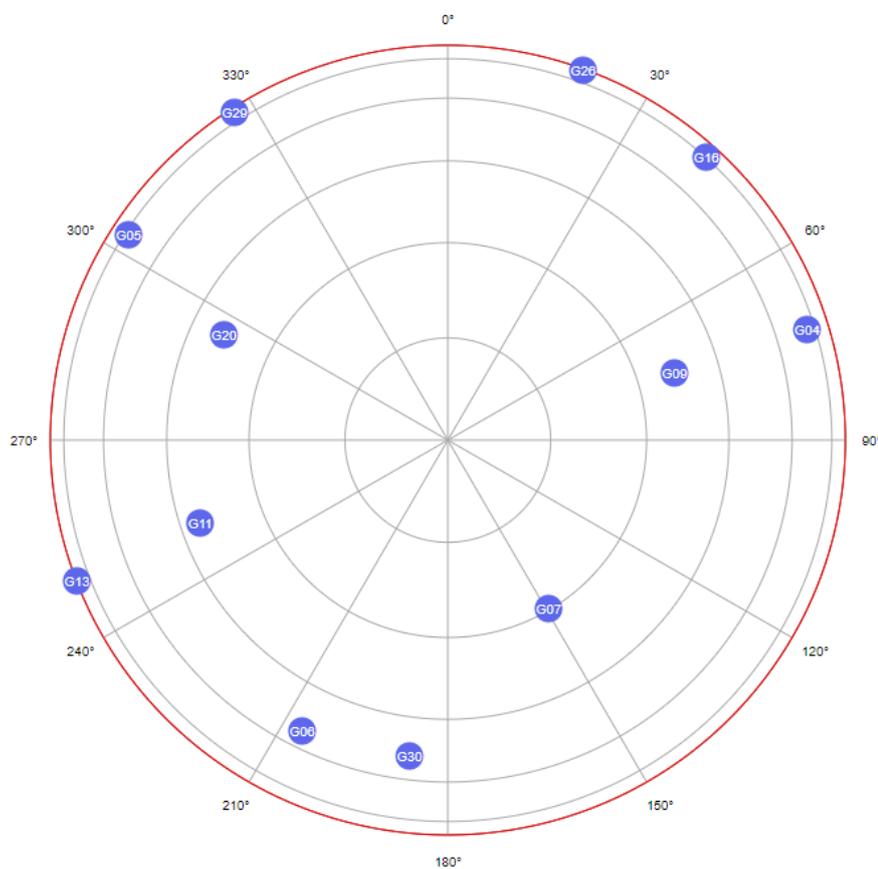


Figure 4.22.: Skyplot for the 9:00:00 UTC time - 21st of November

<sup>9</sup><http://gnssmissionplanning.com/App/Settings>

#### 4. Implementation and Results

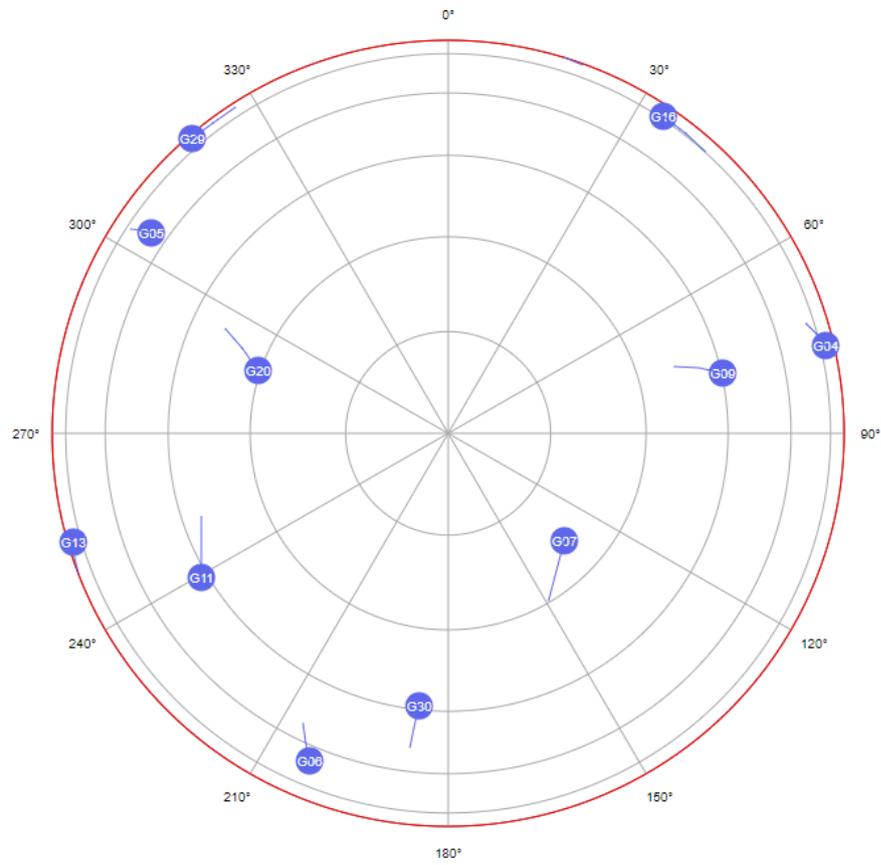


Figure 4.23.: Skyplot for the 9:20:00 UTC time - 21st of November

The number of visible GPS satellites throughout the data acquisition operation was constantly 12.

Moment of time 09:00:00 UTC

**GNGSA,A,3,04,05,06,07,09,11,16,20,29,30,,,0.8,0.5,0.6,1\*3E**



Figure 4.24.: First test site used for data acquisition

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	04, 05, 06, 07, 09, 11, 16, 20, 29, 30	
Blocked - simulation		

Table 4.1.: Visibility analysis for test site 1 - 09:00:00

Moment of time 09:05:00 UTC

**GNGSA,A,3,05,06,07,09,11,16,20,30,,,,,0.9,0.6,0.7,1\*32**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	05, 06, 07, 09, 11, 16, 20, 30	04
Blocked - simulation		

Table 4.2.: Visibility analysis for test site 1 - 09:05:00

Moment of time 09:10:00 UTC

**GNGSA,A,3,05,06,07,09,11,13,16,20,29,30,,,0.8,0.5,0.6,1\*38**

4. Implementation and Results

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	05, 06, 07, 09, 11, 16, 20, 29, 30	04
Blocked - simulation	13	

Table 4.3.: Visibility analysis for test site 1 - 09:10:00

Moment of time 09:15:00 UTC

**GNGSA,A,3,04,05,06,07,09,11,16,20,29,30,,,0.8,0.5,0.6,1\*3E**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	04, 05, 06, 07, 09, 11, 16, 20, 30	26
Blocked simulation	29	

Table 4.4.: Visibility analysis for test site 1 - 09:15:00

Moment of time 09:20:00 UTC

**GNGSA,A,3,04,05,06,07,09,11,16,20,30,,,1.0,0.7,0.7,1\*3F**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	04, 05, 06, 07, 09, 11, 20, 30	
Blocked - simulation	16	

Table 4.5.: Visibility analysis for test site 1 - 09:20:00

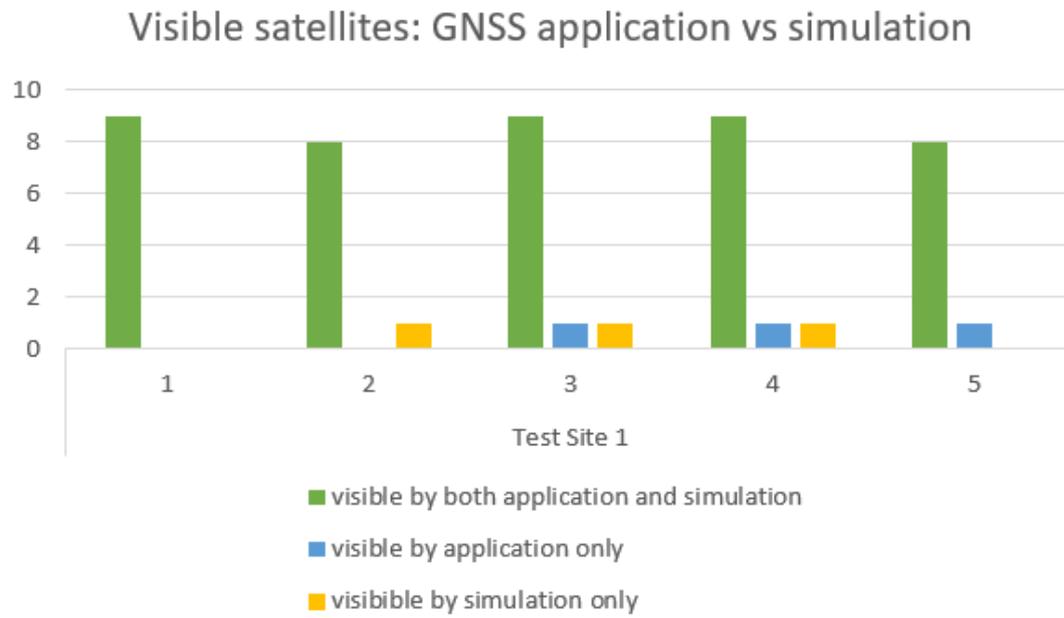


Figure 4.25.: Statistics for the visible satellites for the chosen threshold - Test Site 1

#### 4. Implementation and Results

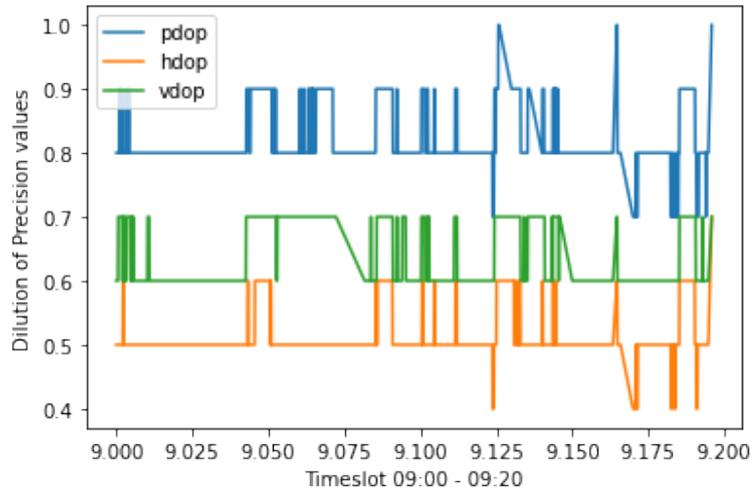


Figure 4.26.: DoP values calculated by the smartphone application - Test site 1

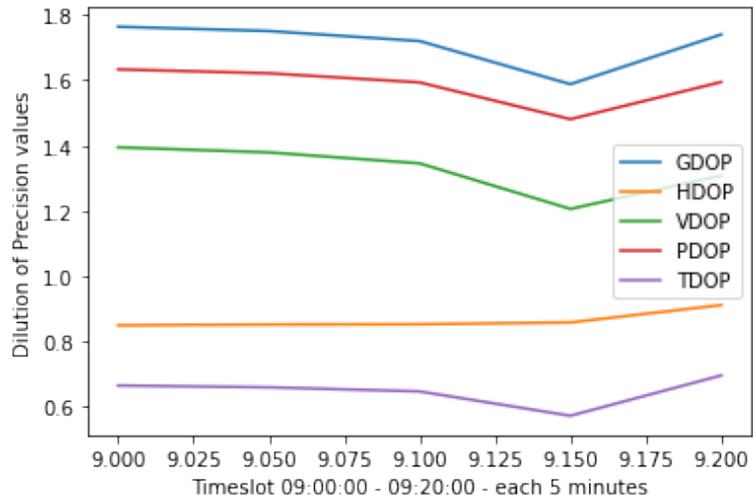


Figure 4.27.: DoP values calculated by the simulation (each 5 minutes) - Test site 1

### 4.3.2. Second Test Site

Coordinates: 51.997803 degrees latitude, 4.377388 degrees longitude, -1.2 mean sea level height

Time: 09:33:41 - 09:53:41 UTC time (10:33:41 - 10:53:41 for the Netherlands), 21st of November, 2023

The second test site is next to the pond, near the TU Delft Science Center.

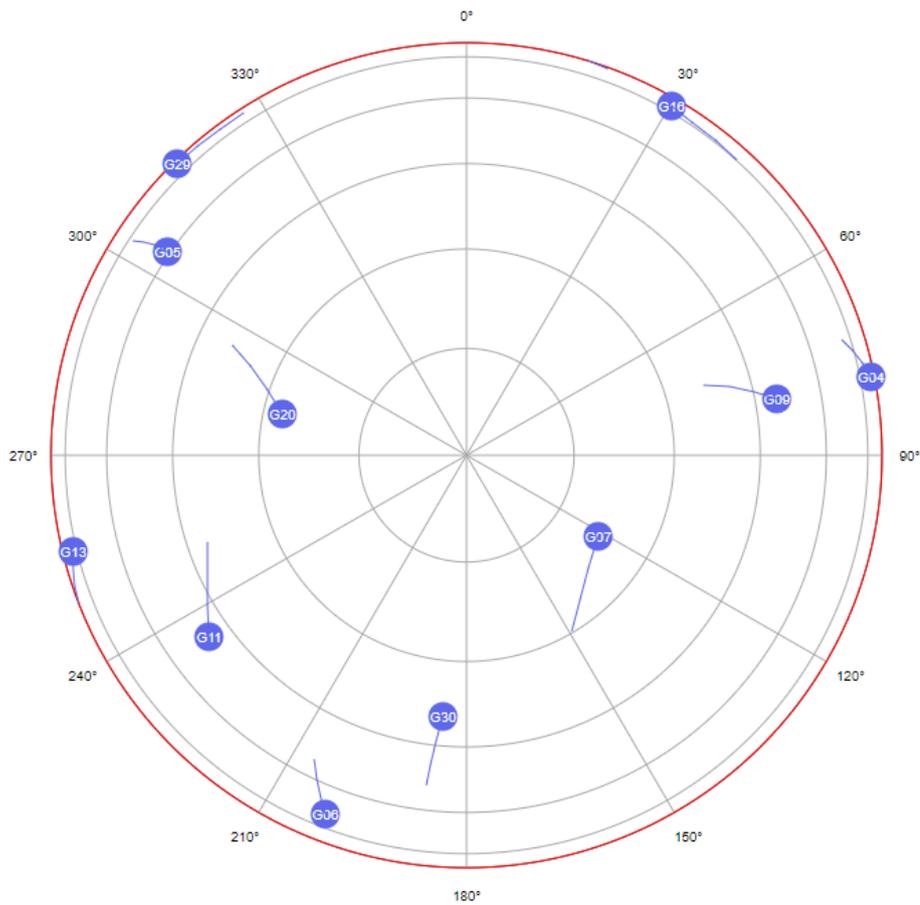


Figure 4.28.: Skyplot for the 9:30:00 UTC time - 21st of November

#### 4. Implementation and Results

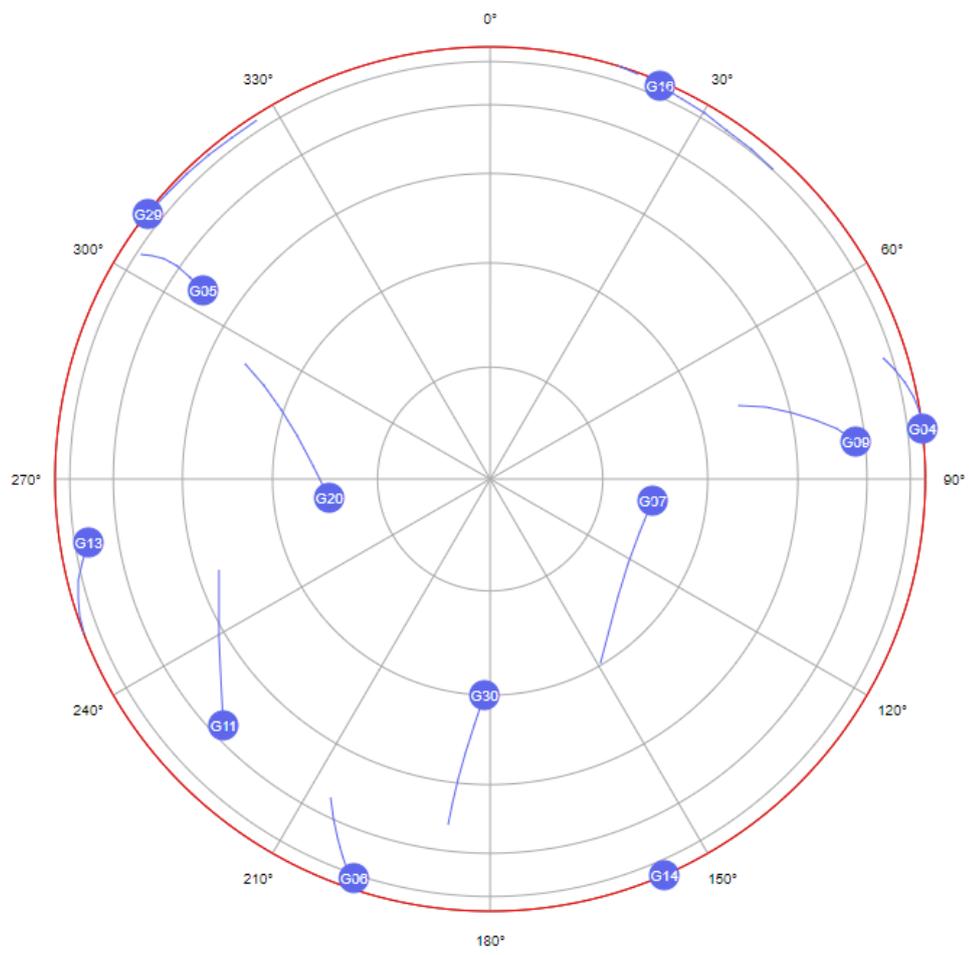


Figure 4.29.: Skyplot for the 9:50:00 UTC time - 21st of November

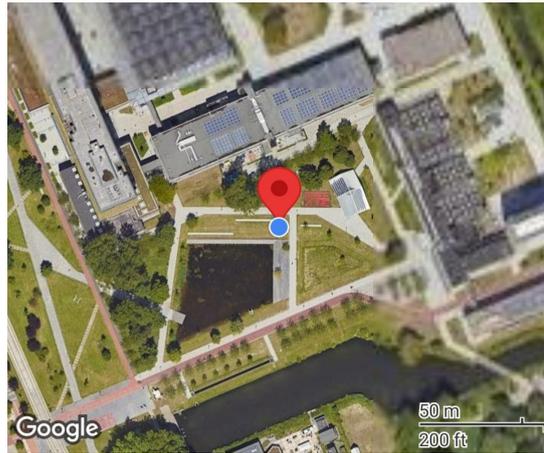


Figure 4.30.: Second test site used for data acquisition

The number of visible GPS satellites throughout the data acquisition operation was constantly 12.

Moment of time 09:33:41 UTC

**GNGSA,A,3,05,06,07,09,11,13,20,29,30,,,,,0.9,0.6,0.8,1\*33**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	05, 06, 07, 09, 11, 20, 30	
Blocked - simulation	13, 29	

Table 4.6.: Visibility analysis for test site 2 - 09:33:41

Moment of time 09:38:41 UTC

**GNGSA,A,3,05,06,07,09,11,13,20,30,,,,,0.8,0.5,0.6,1\*34**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	05, 06, 07, 09, 11, 20, 30	
Blocked - simulation	13	

Table 4.7.: Visibility analysis for test site 2 - 09:38:41

#### 4. Implementation and Results

Moment of time 09:43:41 UTC

**GNGSA,A,3,05,06,07,09,11,13,20,30,,,,,0.8,0.5,0.6,1\*34**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	05, 06, 07, 09, 11, 13, 20, 30	
Blocked - simulation		

Table 4.8.: Visibility analysis for test site 2 - 09:43:41

Moment of time 09:48:41 UTC

**GNGSA,A,3,06,07,09,11,13,20,30,,,,,1.0,0.6,0.7,1\*3A**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	06, 07, 09, 11, 13, 20, 30	05
Blocked - simulation		

Table 4.9.: Visibility analysis for test site 2 - 09:48:41

Moment of time 09:53:41 UTC

**GNGSA,A,3,05,06,07,09,11,13,20,30,,,,,0.8,0.5,0.6,1\*34**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	05, 06, 07, 09, 11, 13, 20, 30	
Blocked - simulation		

Table 4.10.: Visibility analysis for test site 2 - 09:53:41

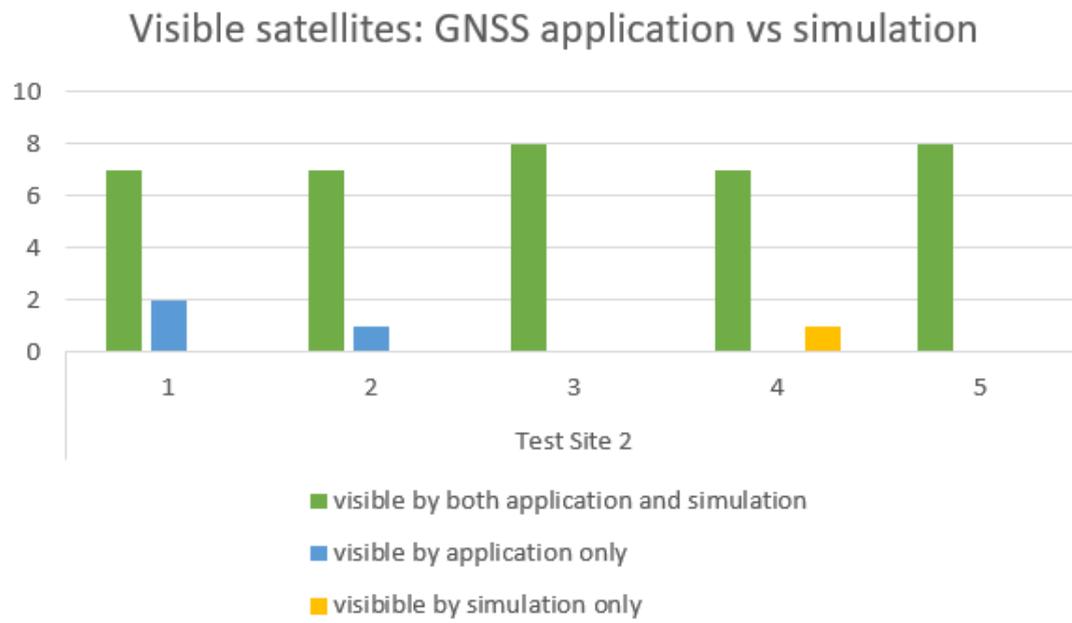


Figure 4.31.: Statistics for the visible satellites for the chosen threshold - Test Site 2

#### 4. Implementation and Results

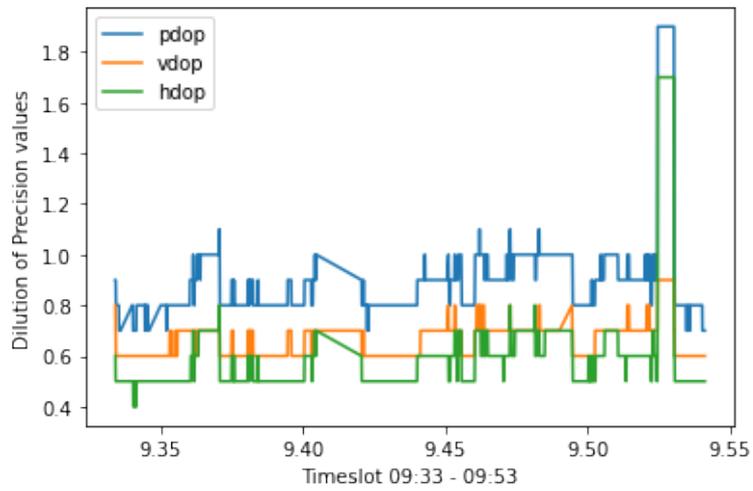


Figure 4.32.: DoP values calculated by the smartphone application - Test site 2

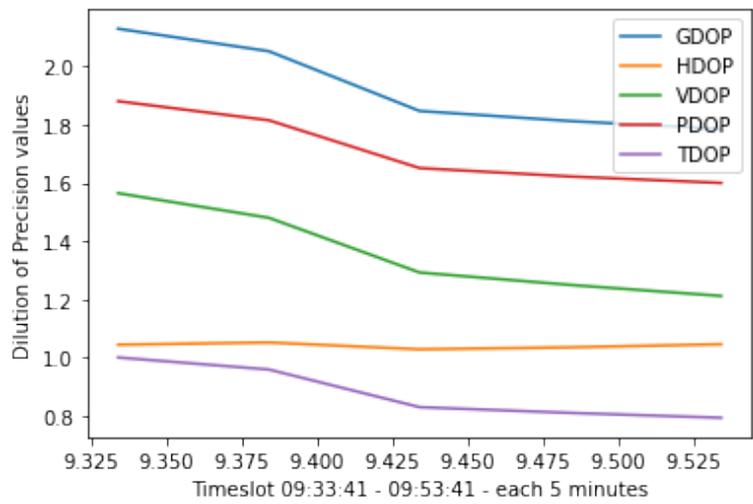


Figure 4.33.: DoP values calculated by the simulation - Test site 2

### 4.3.3. Third Test Site

Coordinates: 51.993984 degrees latitude, 4.378241 degrees longitude, 0.56 mean sea level height

Time: 15:41:57 - 15:53:57 UTC time (16:41:57 - 16:53:57 for the Netherlands), 5th of December, 2023

The third test site is close to the highway bridge, after the X building. For this location, the GNSS receiver was held at a 1.56 height above the ground.

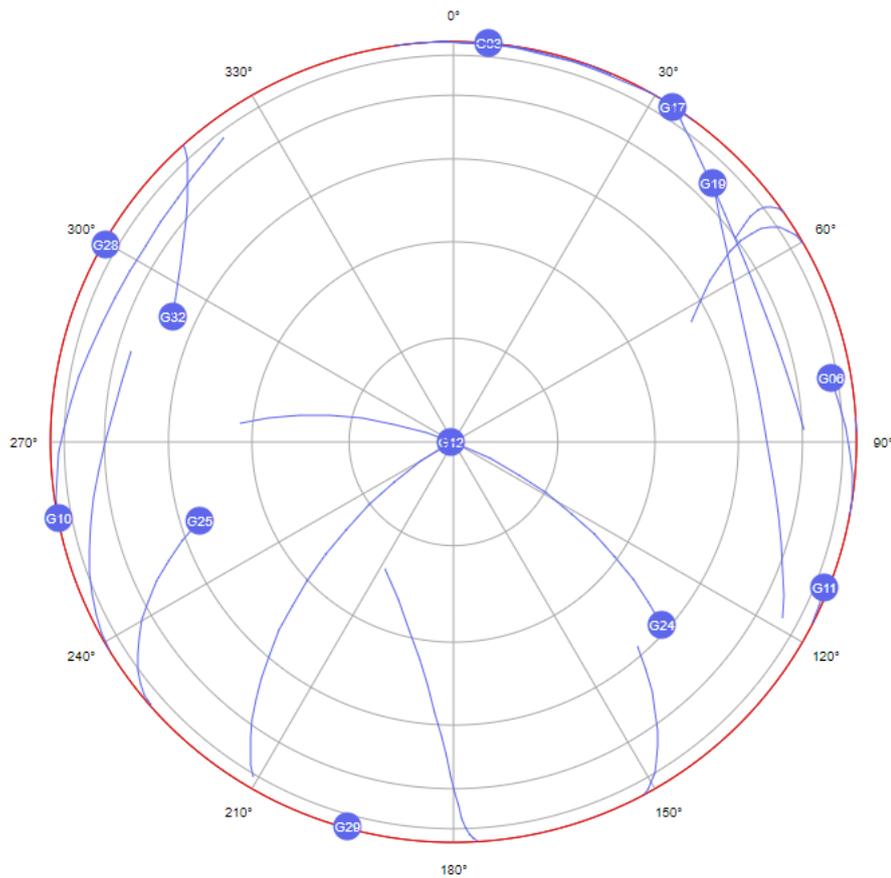


Figure 4.34.: Skyplot for 15:40:00 - UTC Time (5th of December)

#### 4. Implementation and Results

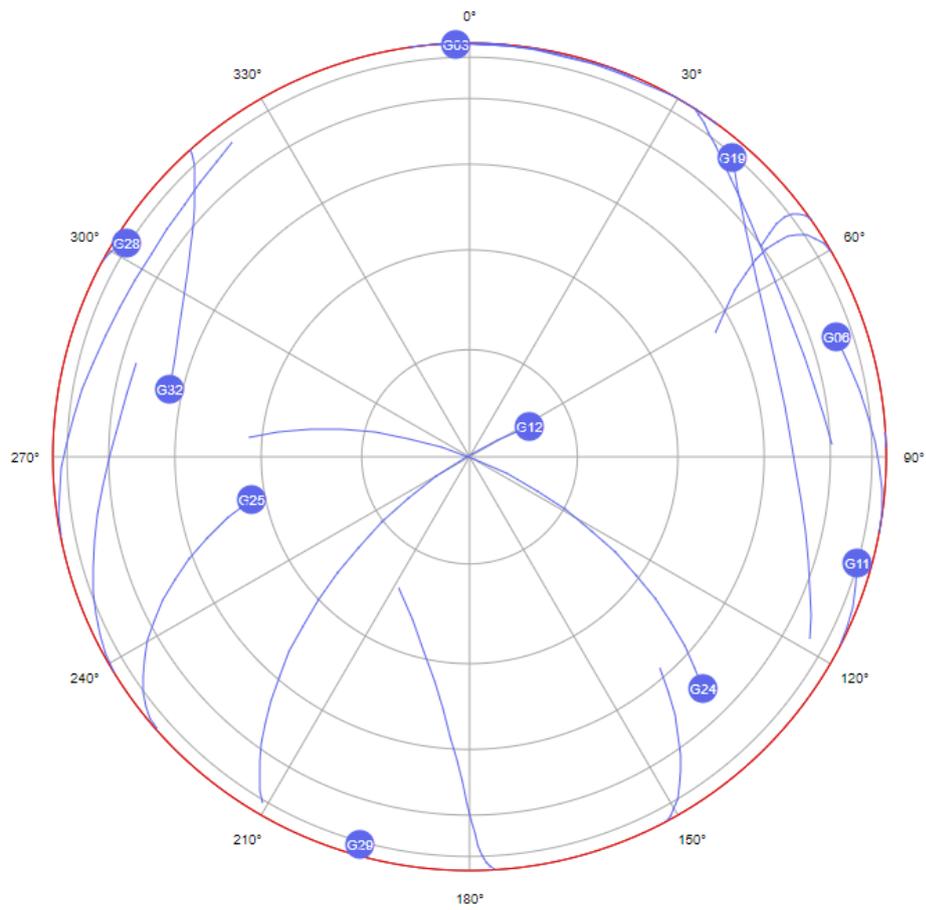


Figure 4.35.: Skyplot for 15:50:00 - UTC Time (5th of December)

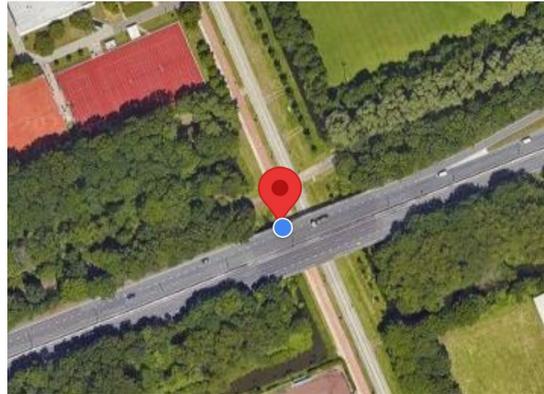


Figure 4.36.: Third location of data acquisition

**GNGSA,A,3,06,11,12,19,32,,,,,,,,,1.5,1.2,0.9,1\*33**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	06, 12, 19, 32	24, 25
Blocked - simulation	11	

Table 4.11.: Visibility analysis for test site 3 - 15:41:57

**GNGSA,A,3,12,19,24,25,32,,,,,,,,,1.3,0.9,0.9,1\*38**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	12, 24, 25, 32	06
Blocked - simulation	19	

Table 4.12.: Visibility analysis for test site 3 - 15:44:57

**GNGSA,A,3,11,19,24,25,32,,,,,,,,,1.1,0.7,0.8,1\*36**

4. Implementation and Results

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	24, 25, 32	06, 12
Blocked - simulation	19	

Table 4.13.: Visibility analysis for test site 3 - 15:47:57

**GNGSA,A,3,11,12,19,24,25,32,,,,,,1.0,0.7,0.7,1\*3B**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	12, 25, 32	06, 24
Blocked - simulation	11, 19	

Table 4.14.: Visibility analysis for test site 3 - 15:50:57

**GNGSA,A,3,06,11,12,19,25,32,,,,,,1.0,0.7,0.7,1\*3B**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	06, 12, 25, 32	24
Blocked - simulation	19	

Table 4.15.: Visibility analysis for test site 3 - 15:53:57

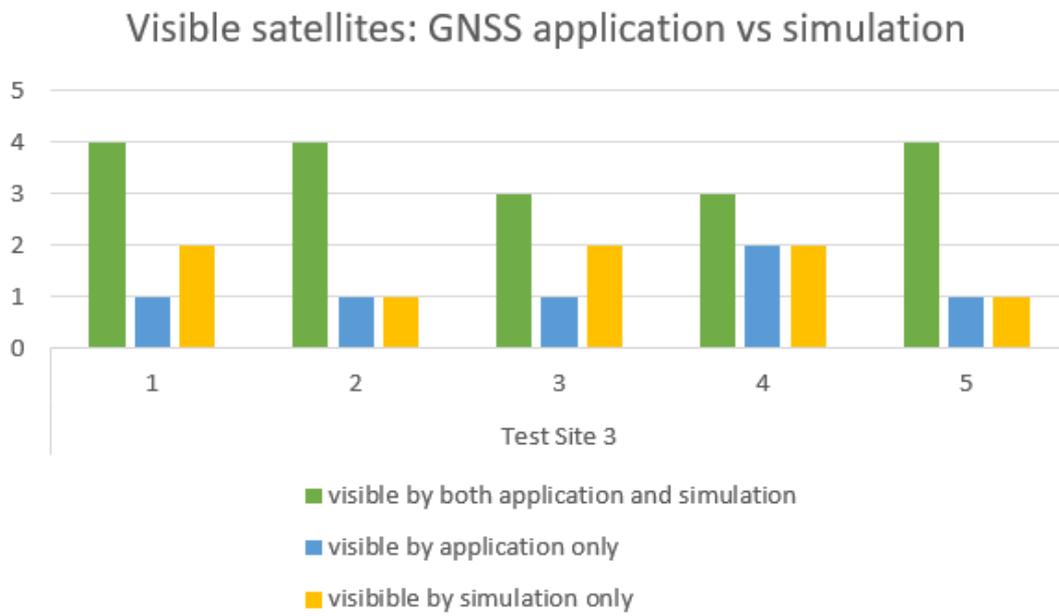
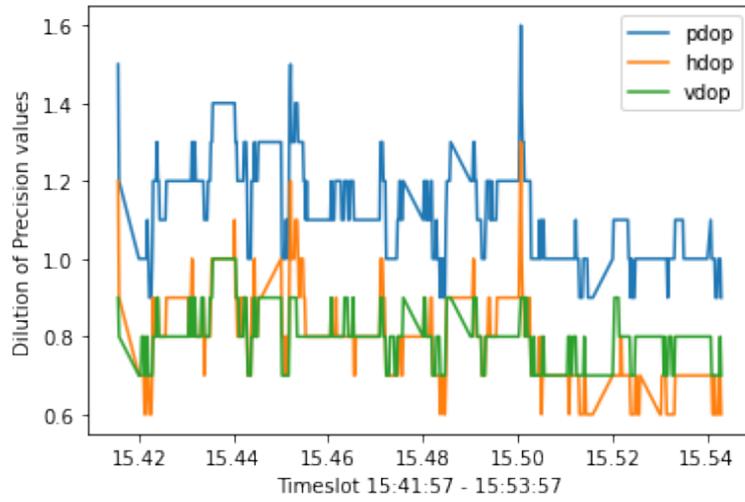


Figure 4.37.: Statistics for the visible satellites for the chosen threshold - Test Site 3

#### 4. Implementation and Results



#### 4.3.4. Fourth Test Site

Coordinates: 52.005319 latitude, 4.373444 longitude, 1.0 mean sea level height Time: 15:02:00 - 15:14:00 UTC Time (16:02:00 - 16:14:00 for the Netherlands), 18th of December, 2023

The fourth test site is in the dense tree area, near the faculty of Architecture.

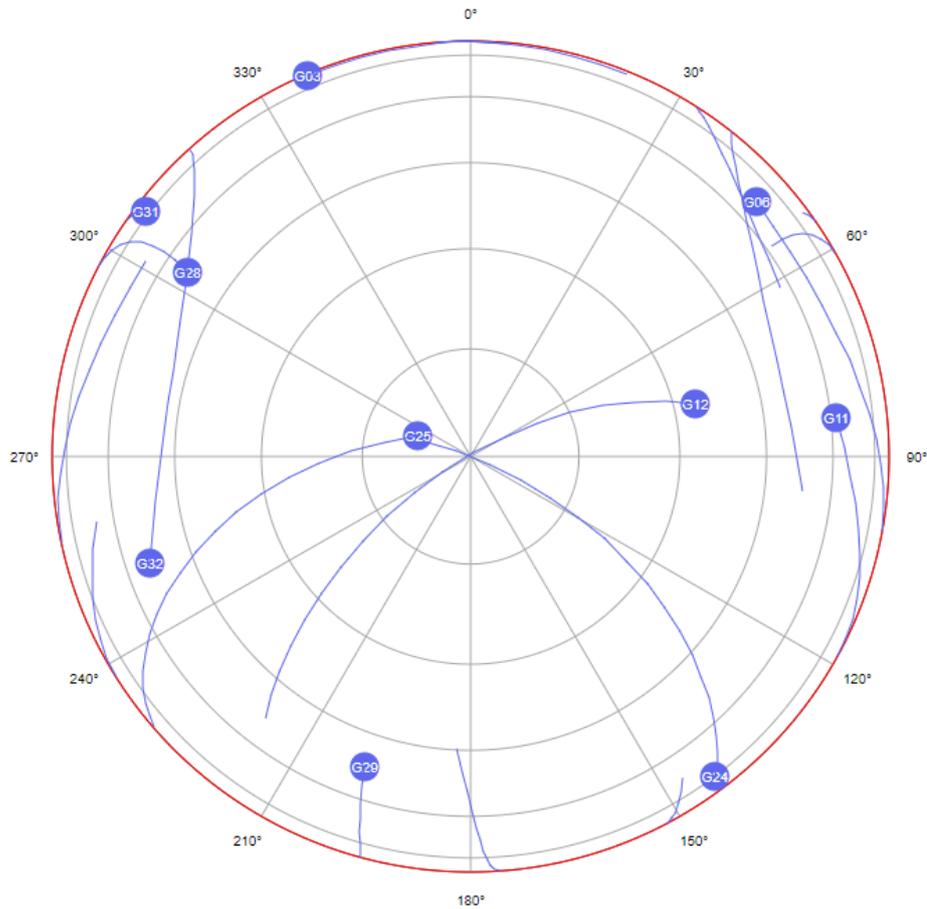


Figure 4.40.: Skyplot for 15:00:00 - UTC Time (18th of December)

#### 4. Implementation and Results

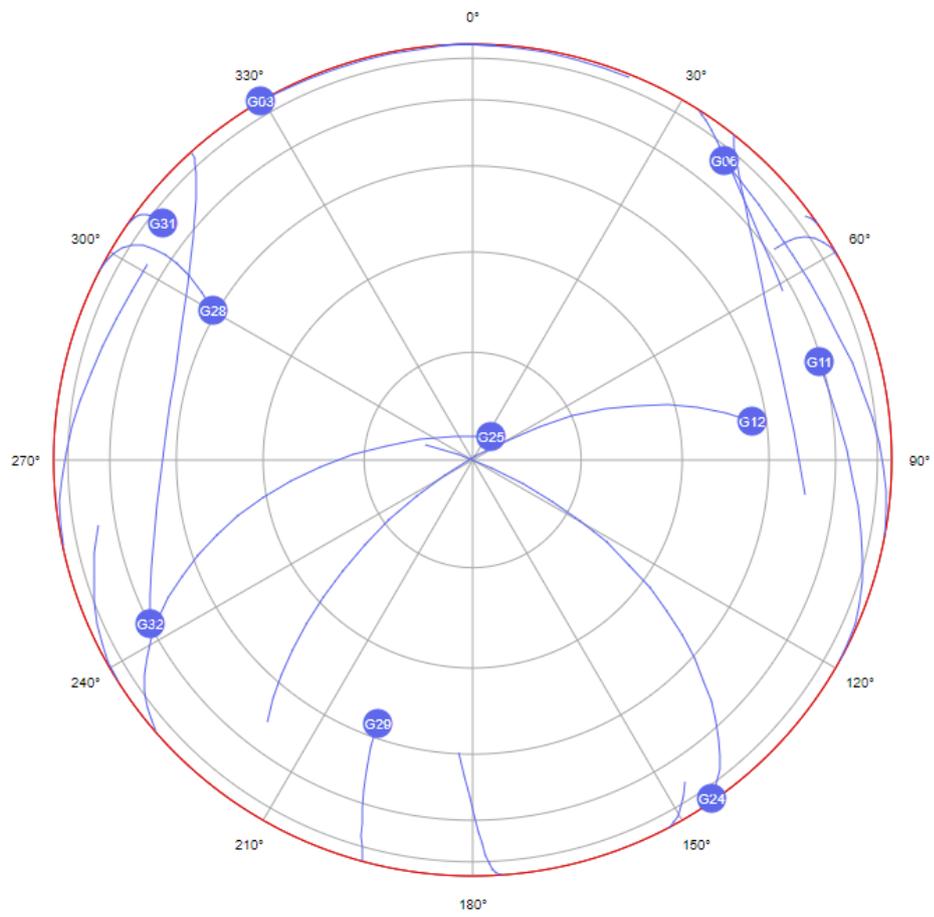


Figure 4.41.: Skyplot for 15:10:00 - UTC Time (18th of December)



Figure 4.42.: Forth location of data acquisition

GNGSA,A,3,06,11,12,19,24,25,28,29,32,,,,,0.8,0.5,0.6,1\*36

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	11, 12, 24, 25, 29, 32	
Blocked - simulation	06, 19, 28	

Table 4.16.: Visibility analysis for test site 4 - 15:02:00

GNGSA,A,3,06,11,12,19,24,25,28,29,32,,,,,1.0,0.7,0.7,1\*3C

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	11, 12, 24, 25, 29, 32	
Blocked - simulation	06, 19, 28	

Table 4.17.: Visibility analysis for test site 4 - 15:05:00

4. Implementation and Results

**GNGSA,A,3,06,11,12,19,24,25,28,29,32,,,,0.9,0.6,0.7,1\*35**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	11, 12, 24, 25, 29, 32	
Blocked - simulation	06, 19, 28	

Table 4.18.: Visibility analysis for test site 4 - 15:08:00

**GNGSA,A,3,06,11,12,19,24,25,28,29,32,,,,0.9,0.5,0.7,1\*36**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	11, 12, 25, 29, 32	
Blocked - simulation	06, 19, 24, 28	

Table 4.19.: Visibility analysis for test site 4 - 15:11:00

**GNGSA,A,3,06,11,12,19,24,25,28,29,32,,,,0.8,0.5,0.7,1\*37**

	Visible - smartphone tool	Blocked - smartphone tool
Visible - simulation	11, 12, 25, 29, 32	
Blocked - simulation	06, 19, 24, 28	

Table 4.20.: Visibility analysis for test site 4 - 15:14:00

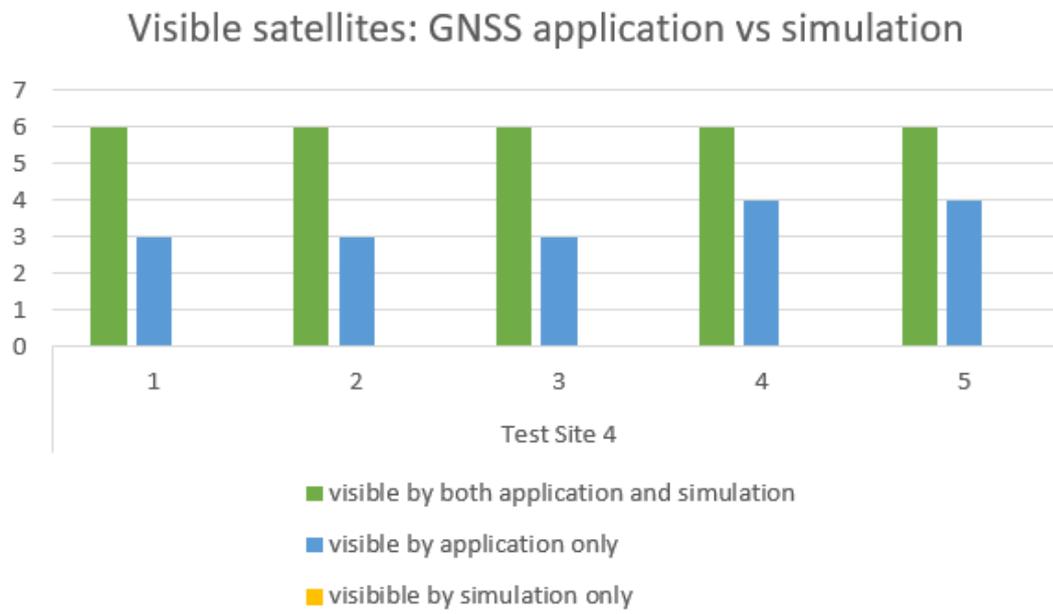


Figure 4.43.: Statistics for the visible satellites for the chosen threshold - Test Site 4

#### 4. Implementation and Results

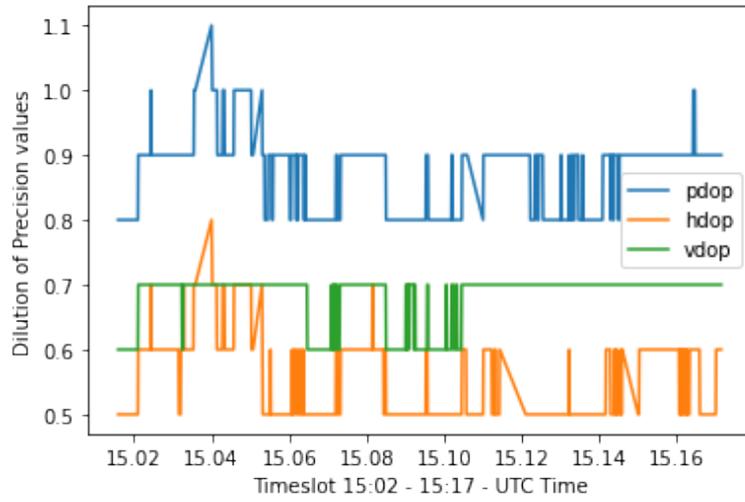


Figure 4.44.: DoP values calculated by the smartphone application - Test site 4

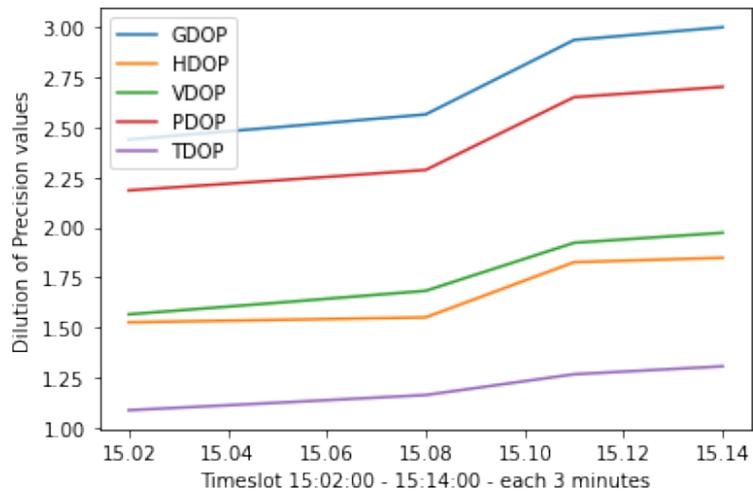


Figure 4.45.: DoP values calculated by the simulation - Test site 4

### 4.3.5. Analysis of results

As seen in the DoP plots at the end of every satellite visibility analysis by location, the values estimated by the implemented simulation are visibly larger than those computed by the NMEA Tools. As seen in table 4.21, there is a discrepancy of 0.7-1.0, with the DoP values given by the simulation being larger than the ones given by the GNSS tool. The third test site, which is the location that is most influenced by the obstruction compared to the others, presents the largest values for DoP in the simulation as well as the real-life scenario and also the largest discrepancy. The second test site, which presents an open sky case, has a sudden spike in value PDoP and HDOP probably due to an error of the receiver as seen in figure 4.32.

	Test Site 1	Test Site 2	Test Site 3	Test Site 4
Real Life	PDoP: 0.8-1.0; VDoP: 0.6-0.7; HDOP: 0.4-0.6	PDoP: 0.7-1.8; VDoP: 0.5-0.8; HDOP: 0.4-1.6	PDoP: 0.9-1.6; VDoP: 0.7-0.9; HDOP: 0.6-1.3	PDoP: 0.8-1.1; VDoP: 0.6-0.7; HDOP: 0.5-0.8
Simulation	PDoP: 1.5-1.7; VDoP: 1.2-1.4; HDOP: 0.8-1.0	PDoP: 1.7-1.9; VDoP: 1.2-1.6; HDOP: 1.05	PDoP: 2.4 - 2.55; VDoP: 2.0; HDOP: 1.4-1.75	PDoP: 2.2-2.6; VDoP: 1.55-1.9; HDOP: 1.5-1.75

Table 4.21.: DoP Values

Although the DoP values from the simulation seem to be very different from the real scenario, they seem to be similar to what the Navmatix<sup>10</sup> online tool presents for the DoP plot when only the GPS satellites are selected for the mission planning.

This plot was generated without inserting any obstruction objects or cut-off angle and yet the DoP values are very high.

This concludes indeed that the geometrical configuration of the satellites which are above the location and at the moment in time of the receiver is an important detail to take into consideration when planning a real surveying scenario.

---

<sup>10</sup>Navmatix

4. Implementation and Results

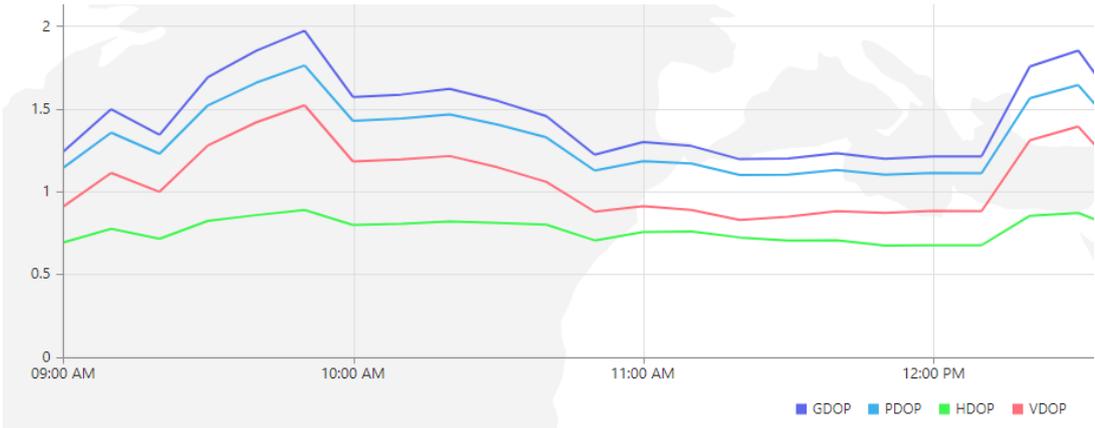


Figure 4.46.: DoP values presented by the Navmatix mission planning tool - 21st of November

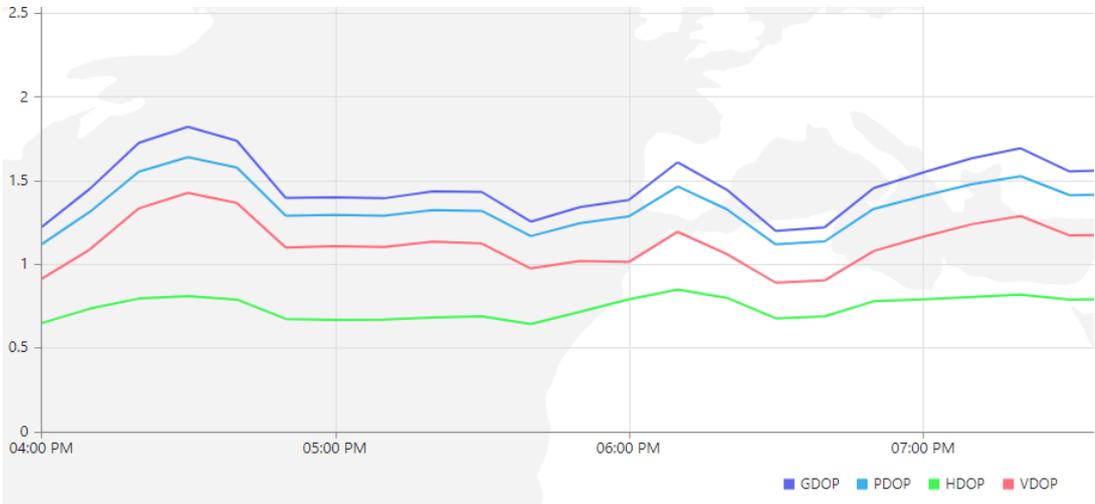


Figure 4.47.: DoP values presented by the Navmatix mission planning tool - 5th of December

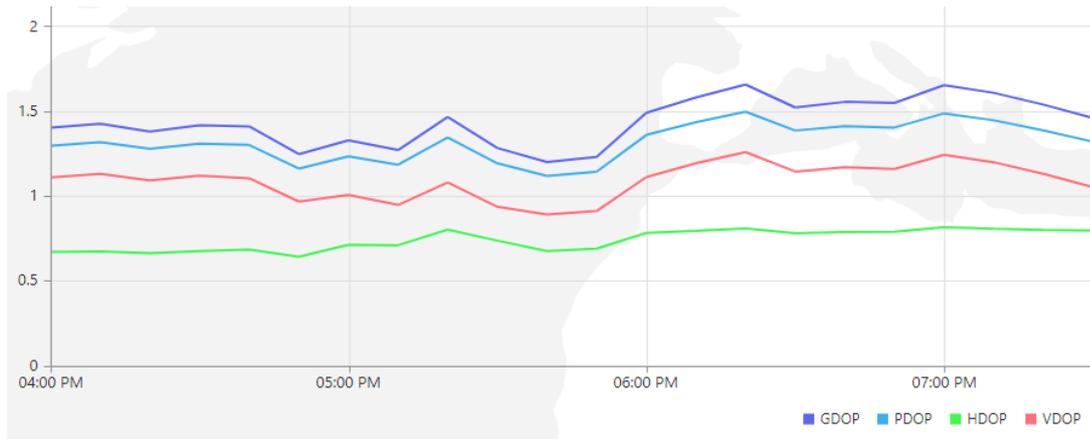


Figure 4.48.: DoP values presented by the Navmatix mission planning tool - 18th of December

Following, I will present the distribution of points which fall inside the cylinder ray during the simulation and determine if the satellite is considered blocked or visible.

The filtered point clouds belong to the LoS between the satellites and the receiver placed in the location of the third test site, on the 5th of December. As seen in the figure 4.49 GPS Satellite 03, which is considered blocked by both the simulation and the GNSS acquisition tool with a total of 30882 points inside its ray, the axis of the satellite-receiver cylinder does not start from a high elevation angle, which results in the cylinder filtering algorithm gathering a lot of points inside the ray and the simulation considering it a blocked LoS. It also presents building points inside its cylinder shaped ray.

GPS Satellite 06 is considered visible by both the GNSS acquisition tool and the simulation with 6086 points inside its cylinder ray. As seen in figure 4.50 and table 4.22, most of the points which fall inside the LoS of the satellite are considered bridge class or class 26. A small group of trees can be seen near the axis, but they are too short to fall inside the cylinder ray.

GPS Satellite 19 is considered visible by the GNSS acquisition tool and blocked by the simulation with 19026 points inside its cylinder ray. As seen in figure 4.51, points from a tall group of trees seem to fall inside the cylinder filter, considered unclassified points according to the table, which result in the simulation considering the satellite blocked. In the case of Satellite 19, the biggest percentage of point which represent the obstructions are the "unclassified" trees points.

As seen in the figures presenting the statistical analysis, there are bigger differences related to which satellites are visible to both the simulation and the GNSS receiver. In sites 1 and 2, which present mostly open sky, the differences between the simulation and the smartphone application are only by 1 or 2 satellites. In sites 3 and 4, which present obstructed sky, the number of different satellites raises to 3-4 satellites. This means that the 9000 points threshold gives different results between the sites with open sky.

#### 4. Implementation and Results

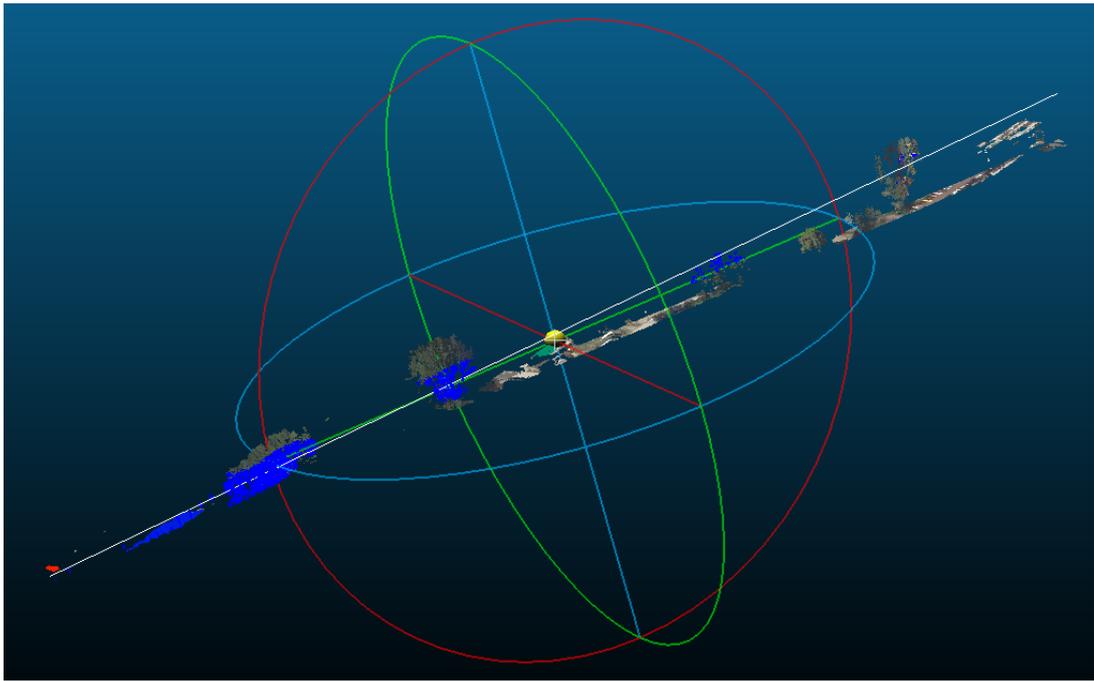


Figure 4.49.: Point distribution for cylinder ray towards Satellite 03 - Test Site 3, MoT 15:53:57 UTC, 5th of December, blocked

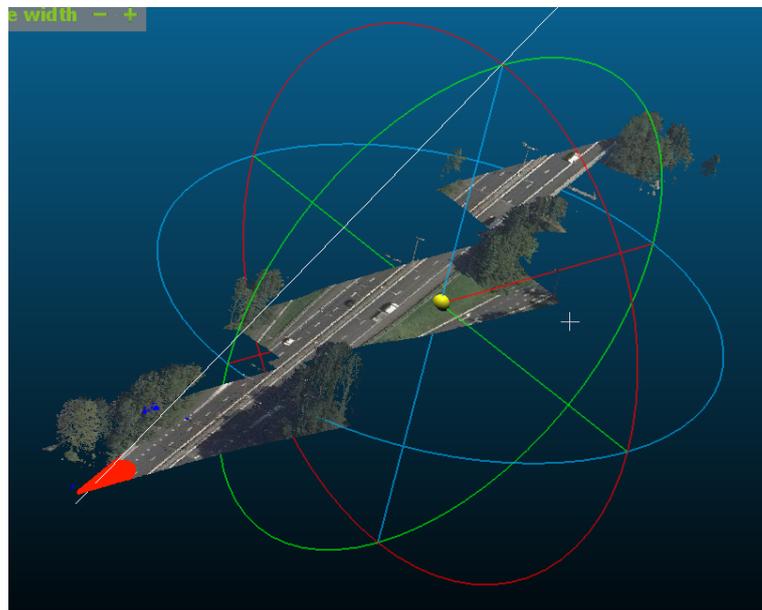


Figure 4.50.: Point distribution for cylinder ray towards Satellite 06 - Test Site 3, MoT 15:53:57 UTC, 5th of December, visible

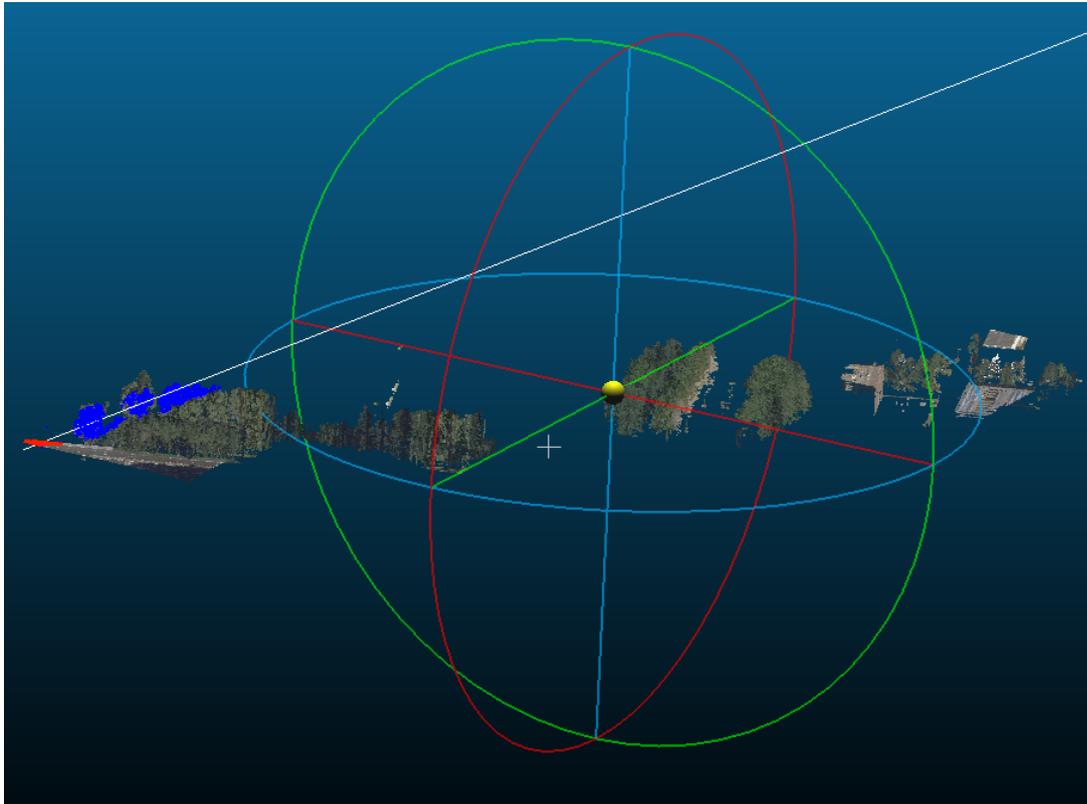


Figure 4.51.: Point distribution for cylinder ray towards Satellite 19 - Test Site 3 MoT 15:53:57 UTC, blocked

	Satellite 03	Satellite 06	Satellite 19
Unclassified	65.977	3.325	89.309
Ground	30.024	0.014	0.005
Building	2.189	-	-
Bridge	1.810	96.660	10.685

Table 4.22.: Percentage of AHN classes which serve as obstructions inside the cylinder ray



## 5. Conclusion and Discussion

### 5.1. Answering the secondary research questions

- How are obstructions represented in calculation of DoP?

Before answering, I would like to make it clear that the projected satellite-receiver line of sight (LoS) is not a "line" in the mathematical sense as studies approximating it with a geometrical line have already been done. In my implementation, I refer to the LoS as a cylinder ray to use it in 3D point cloud representation of the environment, as a simple thin line can go through dense point cloud objects and consider the LoS visible.

Dilution of precision values are calculated using satellites which are visible for the ground antenna. This means it is considered there are no obstacles in the projected line of sight between the antenna and the satellite. If the satellites are evenly distributed in the sky and are not too close to one another (see figure 1.4) and the receiver has full view of the sky, then the DoP values will be low and this means the GNSS mission carried in that area and in that moment of time will have good results in terms of position precision. As stated in Chapter 1, a tall building, a forest formed out of dense and tall trees or high terrain can easily represent obstructions. Such an obstruction in the implemented simulation blocks a certain part of the sky to the antenna and thus, there is no satellite information received from there which increases the values of the DoP as the antenna does not have access to satellites' geometry from the entire sky.

- How good is the carried analysis compared with the given result by simulation tools / GNSS equipment?

As observed from subsection 4.3.5, the results from the implemented simulation are different from those of the given by the smartphone application tool, with the GNSS measurements from the smartphone giving better results than the simulation for the computation of the DoP values. An immediate cause for this can be that the built-in GNSS receiver from the android smartphone is good enough and it also connects to more satellite constellations, which can influence the results.

There are two situations present to be analysed here:

- 1. Satellite is used in the DoP calculation by the simulation, not by the GNSS receiver - optimistic case
- 2. Satellite is used in the DoP calculation by the receiver, not by the simulation - pessimistic case

## 5. Conclusion and Discussion

Usually in the first situation it happens that the satellite is in-view and appears in the GPGPSV line, but not in the GNGSA line, due to a further selection by the receiver which might have to do with the strength of signal received from it and not necessarily with its geometry.

In the second situation, it is mostly because the satellite-receiver cylinder ray has too many points inside it and the satellite is considered blocked to the receiver. This could also mean that the implemented threshold for the quantity of the points is not good. Moreover, it is possible that the LoS in the real life case pierces through "tree" points as seen in the case of GPS Satellite 19 in figure 4.51. This could also be due to the data gathering operations taking place in early winter November-December, when the trees do not have leaves and the point clouds are created using data from spring or summer, when the trees present rich crowns of leaves.

As a further validation test after computing the DoP values and the number of satellites, the azimuth and the elevation angles were also computed inside the simulation to compare them to the ones given by the smartphone application. To give the best results, only the azimuth was computed using ENU coordinates and the elevation was computed using RDNAP coordinates as computing it from ENU gave bad results such as a negative angle. The reason could be the differences in the reference surfaces of the different coordinate systems that the smartphone application and the simulation use for the calculation of the angles. Although with slight differences in the azimuth, the quadrants in which the satellites are seen to match as seen in figure 4.22.

- How can such a simulation of a GNSS mission be helpful?

As discussed first in section 2.2, the purpose of a GNSS mission planning tool is to offer information regarding how the land surveying mission will go at a chosen moment of time based on the satellite geometry configuration.

Since most of the existing GNSS planning tools do not integrate properly 3D data as a representation of the environment of the receiver, a simulation which uses AHN4 point cloud data has been implemented to take into consideration obstructing objects.

Although not perfect as it can also have gaps due to the angle the laser scanner during the data acquisition (for example, the facade of buildings present huge gaps in point clouds) the point cloud data is closest to being a good replica of the real life environment, as it also represents tall vegetation. Rather than manually inserting the obstacles which takes a long time as the obstacle needs to have a certain height, orientation and distance from the receiver. The point cloud is already representing the geometry of the environment which can present obstructions to a certain satellite or not.

- What determines the threshold that decides if a line of sight is visible or not?

The threshold that determines if the LoS is visible or not for the implemented GNSS simulation is represented by the number of points that fall inside the cylinder ray from the satellite to the receiver. As a conclusion, choosing the radius of the cylinder should be linked to 4.1, thus to the size of the square boxes, which should not be too big, as a large radius can catch point cloud objects that are not necessarily blocking the LoS or outlier points and a small radius might not give proper information about geometry of the point cloud object.

If the size of the square boxes, and thus the radius of the cylinder, is to be increased, the limit for the number of points that determines the threshold should also be increased, as the cylinder algorithm will catch a larger number of points.

As seen in the statistical analysis figures for the threshold of 9000 points, this threshold gives variable results for each test site, especially for the test sites that present obstructions instead of open sky. This means that this point threshold should be adapted depending on the location of test site. The class the points representing the obstruction belong to could also be a further indicator if the satellite should be considered blocked or not. As seen in table 4.22 for Satellite 19, which is considered visible in the real scenario, presents the "unclassified" category of points as forming most of the obstructions in its LoS for the simulation. It seems to be a similar case regarding the "unclassified" tree points for Test Site 4, where the receiver is surrounded by trees and as seen in figure 4.43, the simulation considers 3-4 satellites blocked while they are visible in the real life case. The thesis does not further study the correlation between the threshold and the number of points which fall inside the cylinder as the results can vary between the test sites.

Small and dense point cloud object or a smaller part from a big object can contain hundreds of points. In real life, although the cylinder shaped LoS can contain in the simulation a few thousand points, that LoS would not be considered blocked and the satellite will be taken into consideration for DoP calculation. Outliers from the point cloud can also fall inside the cylinder ray which normally are not present in real life.

The simulation does store the distribution of the points inside the ray for them to be visualised and identifies what objects they are based on their AHN4 classification (buildings and trees), but it does not further analyse it.

## 5.2. Answer to the main research question

*To what extent can point cloud data be integrated in a GNSS planning simulation and how does this relate to satellite geometry and influences DoP values?*

The word "extent" in the main research question mostly refers to the largest possible quantity of point cloud data which can be used in my simulation that runs in python while taking into consideration the lowest computation time for one moment of time. The purpose of this thesis is to further analyse the results of the GNSS planning simulation that uses this quantity of point cloud data.

A downloaded .LAZ point cloud for a small tile is usually 500 megabytes, which in one take is a lot to compute for a simple machine, using the python programming language. Two filtering algorithms have been implemented to help in better managing the time for a GNSS simulation by selecting necessary data without the loss of important detail.

Applying first the height filter command from LASTools minimizes the data by half. Then, applying the 2D algorithm, the box shaped point clouds for one moment of time altogether contain already somewhere between 40 - 55 megabytes, which means that this algorithm already filters more or less to a tenth of the original point cloud. Further applying the 3D

## *5. Conclusion and Discussion*

algorithm, the point cloud samples for one MoT obtained from the cylinder-shaped ray filter can contain up to 2 megabytes of data and can go lower than 1 megabyte.

Simply setting up a threshold and applying it for all the satellites, in every situation of the test sites is not enough. As seen in the result analysis, there are often cases where the simulation presents different results compared to the GNSS data gathering tool, depending of the environment of the receiver.

As answered in the first subquestion, an obstruction blocks the access of the satellite to the receiver and the receiver has less satellite positions from which to compute the DoP values, therefore higher DoP values.

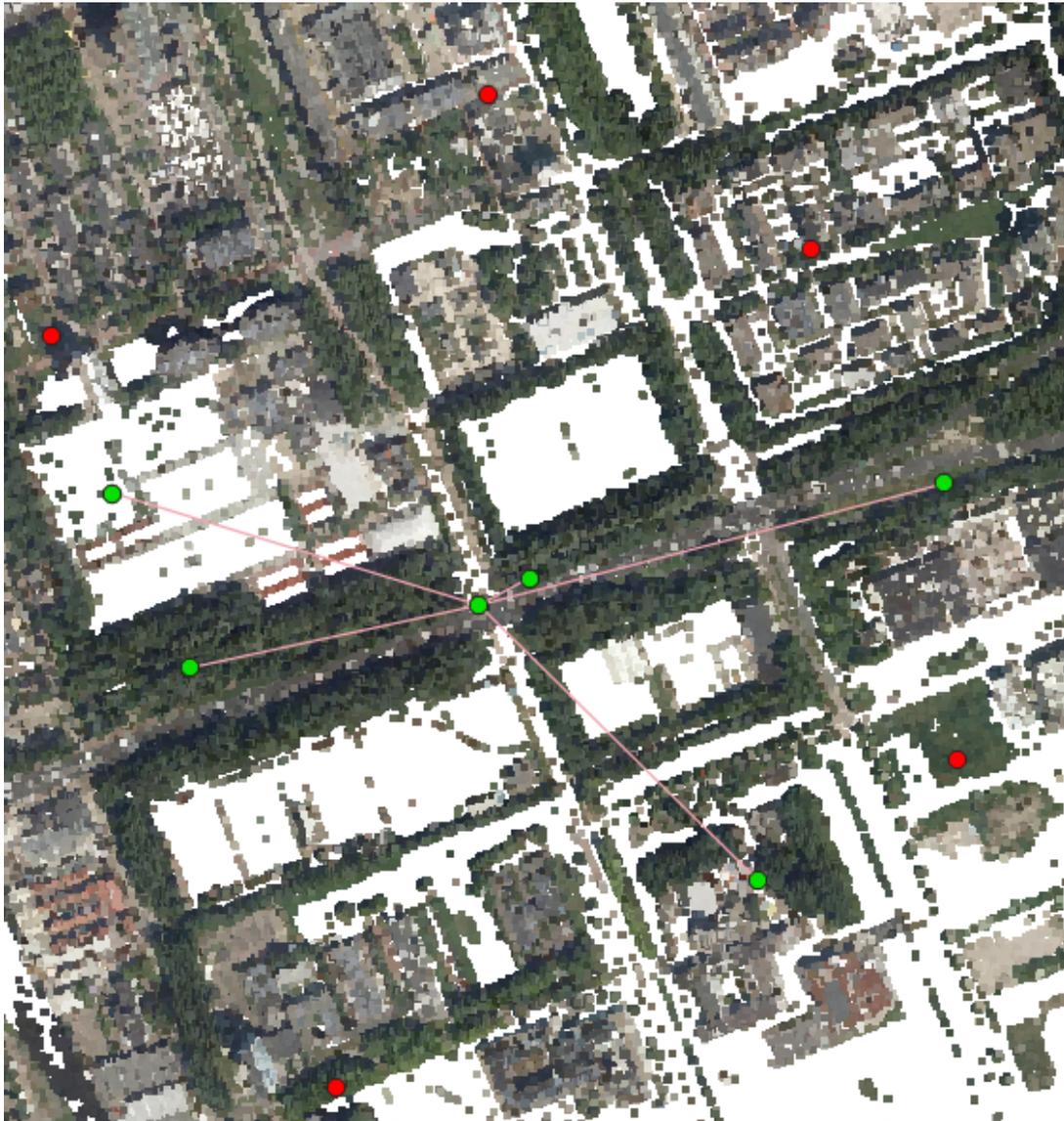


Figure 5.1.: Visible satellites (green dots) and the LoS (pink lines) towards the receiver

In figure 5.1, the green points are the satellite positions considered visible by the simulation. As it can be observed, the satellites from the North and South are considered blocked due to the number of points from the point cloud which fall in their LoS exceeding the threshold and the simulation computes the DoP values only with the satellite positions from the central part.

Relating to the geometry, low elevation satellites usually present a lot of points inside their cylinder shaped ray and result in the simulation considering it blocked. The opposite usually happens to high elevation satellites although not always, depending on the how tall the obstructing objects are.

## 5. Conclusion and Discussion

It has to be said that for the simulation to work, the satellites and the receiver positions had to be computed to RDNAP as the point cloud data was represented in RDNAP. As the satellites are far up in the sky, "ghost" satellites had to be created to fit in the boundaries of the Netherlands, which could sometimes give unexpected results, as ECEF uses the ellipsoid as the reference surface, while RDNAP and other projected coordinate systems use the geoid as the reference surface. If this simulation were to be applied for another country, the positions of the satellites would have to be translated in the country's respective coordinate system.

### 5.3. Sources of errors and limitations

While gathering the data for the real case, it has to be said that the combination of NMEA tools and an android phone might not be the best solution. Sometimes, due to weather conditions, the location precision given by the android smartphone can give bad results. NMEA Tools would sometime stop during the time of measurement probably because the phone was situated in the same place for too long. If slightly moved, the positioning given in the GNGGA string would drastically change, the most being on the height coordinate as it would fluctuate meters. In case of such a problem, the CloudCompare tool in figure 4.8 can be used to identify the Z coordinate of the receiver.

For the simulation, computing the elevation and azimuth angles in different coordinate systems seems to give different results between RDNAP and ENU. A first cause would be the reference surface and if the coordinate system is projected or not. Calculating the elevation angle in RDNAP seems to give more realistic results and closer to the real case than calculating them in ENU.

Although in figure 4.2 from [Ledoux et al.](#) it states that AHN4 uses the number 5 to classify high vegetation, it was not the case for the used point cloud samples, as points belonging to trees would be classified as 1. The objects were identified as trees by using visualising software as seen in images 4.49 - 4.51 and table 4.22.

### 5.4. Recommendations and future work

As seen in section 4.3 and discussed in the second subquestion, the results from the real scenario could be better because the smartphone application has access to more than just the GPS constellation when registering the NMEA strings. A first recommendation for the future work would be to add in this simulation other satellite constellations such as Galileo and GLONASS to better match the results given by GNSS receivers used in this domain as most of them can receive signal from all the satellite constellations orbiting the Earth.

I would also recommend using for the data acquisition operation an actual GNSS receiver combined with a high-precision antenna and especially designed for positioning and land surveying.

#### 5.4. Recommendations and future work

Secondly, a future feature I would implement is analysing the point distribution from inside the cylinders using clustering algorithms such as DBSCAN to learn more about the positioning of the point cloud objects inside the ray and to finally further decide if it can still be considered a visible or a blocked line of sight even if the amount of points inside the cylinder exceeds the threshold. Another type of future analysis I would implement is trying to determine the point threshold based on the volume of the cylinder ray and the number of points which fall inside it.

Thirdly, I recommend testing the implemented simulation and comparing it with data gathered in unfavorable locations (under a bridge, in a forest etc.) and further observe the simulation to see if it displays the right results.

For a further study in this domain, I would recommend integrating my simulation into an online mission planning tool which uses actual 3D data of the surroundings of the receiver depending on the location the user has inserted and larger tiles. I think a more developed and advanced online tool would be helpful for future land surveying missions.



## A. GNSS Measurement Data

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
04	71.0	18.0	70.2036	18.9873	7480	visible
05	302.0	16.0	297.6383	16.7830	6326	visible
06	205.0	34.0	215.8231	34.4307	418	visible
07	149.0	59.0	171.0113	60.2631	1856	visible
09	73.0	53.0	68.0368	53.4428	5207	visible
11	250.0	48.0	254.5724	48.8937	458	visible
13	248.0	1.0	251.3771	1.9015	60729	blocked
16	42.0	14.0	34.7010	14.5486	4675	visible
20	295.0	51.0	288.6939	51.5196	3092	visible
26	19.0	4.0	11.2581	4.1627	11246	blocked
29	326.0	9.0	319.6458	9.1514	15926	blocked
30	185.0	35.0	198.5434	36.3391	192	visible

Table A.1.: GNSS measurements for test site 1 - 09:00:00

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
04	73.0	17.0	71.4031	17.0645	6533	visible
05	302.0	17.0	297.9140	18.8094	4120	visible
06	205.0	33.0	214.6592	32.1734	1231	visible
07	146.0	61.0	169.0579	62.0186	1902	visible
09	73.0	52.0	69.7493	51.2459	5186	visible
11	248.0	47.0	251.9415	47.8351	446	visible
13	248.0	2.0	252.3580	3.7270	40691	blocked
16	40.0	14.0	32.3480	14.3469	5488	visible
20	293.0	52.0	287.4704	53.5855	3091	visible
26	18.0	2.0	9.7788	3.0580	10796	blocked
29	324.0	8.0	317.8355	8.9828	16021	blocked
30	185.0	37.0	198.8143	38.6748	239	visible

Table A.2.: GNSS measurements for test site 1 - 09:05:00

A. GNSS Measurement Data

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
04	73.0	15.0	72.6195	15.1616	5579	visible
05	302.0	19.0	298.1240	20.8540	2734	visible
06	203.0	30.0	213.5784	29.9177	2372	visible
07	143.0	62.0	166.6199	63.6336	1933	visible
09	74.0	49.0	71.4077	49.0687	5215	visible
11	246.0	46.0	249.4169	46.6227	434	visible
13	250.0	4.0	253.3537	5.56319	35783	visible
16	37.0	14.0	30.0530	14.0330	7317	visible
20	292.0	54.0	287.4704	55.6157	3075	visible
26	16.0	2.0	8.3716	1.8888	9943	blocked
29	323.0	8.0	316.0409	8.7128	16208	blocked
30	185.0	40.0	199.0932	41.0225	278	visible

Table A.3.: GNSS measurements for test site 1 - 09:10:00

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
04	73.0	15.0	73.8515	13.2780	4753	visible
05	303.0	21.0	298.2632	22.9158	2114	visible
06	203.0	28.0	212.5705	27.6677	3918	visible
07	139.0	63.0	163.5543	65.0785	1963	visible
09	74.0	47.0	73.0237	46.9115	5283	visible
11	243.0	45.0	247.0067	45.2723	424	visible
13	248.0	2.0	252.3580	7.4105	28188	blocked
16	36.0	13.0	27.8239	13.6089	9046	visible
20	291.0	56.0	284.3336	57.5965	3038	visible
26	0.0	0.0	7.0350	0.6595	7525	blocked
29	322.0	8.0	317.8355	8.3434	18055	blocked
30	185.0	42.0	199.3770	43.3803	306	visible

Table A.4.: GNSS measurements for test site 1 - 09:15:00

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
04	75.0	11.0	75.0976	11.4135	4993	visible
05	303.0	25.0	298.3258	24.9935	2153	visible
06	202.0	33.0	211.6259	25.4274	5876	visible
07	132.0	66.0	159.6745	66.3208	1986	visible
09	73.0	52.0	74.6059	44.7744	5419	visible
11	77.0	44.0	244.7154	43.7997	408	visible
13	253.0	9.0	255.3848	9.2692	21078	blocked
16	33.0	12.0	25.6673	13.0773	10772	blocked
20	288.0	59.0	282.3766	59.5110	3001	visible
29	317.0	7.0	312.5148	7.8774	20102	blocked
30	185.0	45.0	199.6618	45.7455	333	visible

Table A.5.: GNSS measurements for test site 1 - 09:20:00

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
04	78.0	6.0	78.5692	6.4061	26121	blocked
05	303.0	30.0	298.0532	30.7421	6290	visible
06	199.0	18.0	209.2968	19.3705	5312	visible
07	116.0	68.0	142.8933	68.4189	1338	visible
09	80.0	38.0	78.8086	39.0306	519	visible
11	232.0	38.0	239.0564	39.2653	53	visible
13	255.0	14.0	258.2349	14.4148	10686	blocked
16	28.0	10.0	20.1772	11.1011	22684	blocked
20	278.0	63.0	275.4970	64.2228	1035	visible
29	313.0	5.0	307.8665	6.1327	20035	blocked
30	184.0	52.0	200.4063	52.2325	371	visible

Table A.6.: GNSS measurements for test site 2 - 09:33:41

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	Lo
04	80.0	4.0	79.8566	4.6076	38856	blocked
05	303.0	32.0	297.7693	32.8613	6734	visible
06	199.0	17.0	208.5174	17.1919	4948	visible
07	112.0	68.0	133.8360	68.6278	1330	visible
09	80.0	38.0	80.3088	36.9673	594	visible
11	230.0	37.0	237.2039	37.4658	325	visible
13	257.0	15.0	259.2925	16.3196	9799	blocked
16	9.0	26.0	18.3312	10.2012	27163	blocked
20	275.0	64.0	272.3853	65.6668	277	visible
29	312.0	4.0	306.2273	5.33824	18221	blocked
30	182.0	53.0	200.6466	54.5953	113	visible

Table A.7.: GNSS measurements for test site 2 - 09:38:41

A. GNSS Measurement Data

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
04	80.0	2.0	81.1523	2.8255	26868	blocked
05	303.0	33.0	297.3731	34.9849	7098	visible
06	199.0	15.0	207.7658	15.0355	4236	visible
07	106.0	68.0	133.8360	68.5195	1323	visible
09	81.0	35.0	81.7940	34.9234	762	visible
11	229.0	35.0	235.4589	35.6088	596	visible
13	258.0	17.0	260.3566	18.2376	7057	visible
14	0.0	0.0	166.2434	1.2153	52729	blocked
16	9.0	26.0	16.5736	9.2141	26841	blocked
20	258.0	66.0	268.9506	66.9026	43	visible
29	310.0	4.0	306.2273	4.4661	16889	blocked
30	182.0	55.0	200.8561	56.9459	9	visible

Table A.8.: GNSS measurements for test site 2 - 09:43:41

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	Lo
04	80.0	2.0	82.4551	1.0592	19652	blocked
05	302.0	36.0	296.8550	37.1087	7296	visible
06	198.0	12.0	207.0363	12.9029	3726	visible
07	106.0	67.0	111.8772	68.1017	1319	visible
09	82.0	33.0	83.2657	32.8984	958	visible
11	226.0	33.0	233.8149	33.7048	1071	visible
13	260.0	19.0	261.4261	20.1692	3837	visible
14	156.0	2.0	165.4241	3.0394	32469	blocked
16	7.0	23.0	14.9045	8.1449	19396	blocked
20	265.0	67.0	265.2092	67.8882	0	visible
29	307.0	3.0	303.0509	3.5215	20619	blocked
30	181.0	58.0	201.0227	59.2772	0	visible

Table A.9.: GNSS measurements for test site 2 - 09:48:41

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
05	302.0	38.0	296.2041	39.2279	7464	visible
06	198.0	10.0	207.0363	10.7961	7903	visible
07	92.0	67.0	100.6647	67.39504	1317	visible
09	84.0	30.0	84.7251	30.8919	1015	visible
11	224.0	31.0	232.2650	31.7631	1783	visible
13	261.0	21.0	262.4996	22.1148	1394	visible
14	154.0	4.0	164.6217	4.8936	14888	blocked
16	7.0	23.0	13.3236	6.9992	11396	blocked
20	258.0	68.0	261.1948	68.5825	0	visible
29	306.0	2.0	301.5146	2.5092	18953	blocked
30	180.0	61.0	201.1311	61.5803	0	visible

Table A.10.: GNSS measurements for test site 2 - 09:53:41

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
03	0.0	0.0	355.4801	0.6811	55700	blocked
06	78.0	18.0	78.2291	18.7042	8068	visible
10	257.0	1.0	259.35737	2.0274	119459	blocked
11	109.0	9.0	115.5260	9.4599	34808	blocked
12	37.0	89.0	267.7805	89.4317	559	visible
17	32.0	2.0	25.2336	2.4780	40716	blocked
19	43.0	23.0	36.2408	23.4044	8454	visible
24	130.0	45.0	144.7376	45.6267	4548	visible
25	253.0	49.0	256.0360	49.6928	3945	visible
28	299.0	6.0	295.5331	6.7210	50832	blocked
29	194.0	5.0	203.2793	5.9520	18953	blocked
32	292.0	40.0	287.9769	40.5697	2906	visible

Table A.11.: GNSS measurements for test site 3 - 15:41:57

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
03	0.0	0.0	354.4486	1.2218	51188	blocked
06	78.0	18.0	76.7264	19.3429	7795	visible
10	257.0	1.0	258.5942	1.0711	136130	blocked
11	109.0	9.0	114.5390	10.4384	34808	blocked
12	327.0	88.0	271.7915	87.9912	2154	visible
17	32.0	2.0	35.6220	1.3908	55948	blocked
19	43.0	23.0	36.2408	22.3281	10617	visible
24	130.0	45.0	144.9550	45.6267	4763	visible
25	253.0	49.0	256.7798	51.0402	3613	visible
28	299.0	6.0	295.8506	7.8583	37202	blocked
29	194.0	5.0	203.3899	7.2100	18953	blocked
32	292.0	40.0	286.6138	41.0306	3020	visible

Table A.12.: GNSS measurements for test site 3 - 15:44:57

A. GNSS Measurement Data

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
03	0.0	0.0	353.4044	1.7336	33297	blocked
06	77.0	18.0	75.1959	19.9508	7448	visible
10	0.0	0.0	257.8379	0.1131	154031	blocked
11	109.0	10.0	113.5316	11.4118	35914	blocked
12	53.0	87.0	276.7826	86.5330	795	visible
17	0.0	0.0	24.9967	0.3010	38704	blocked
19	42.0	21.0	35.0533	21.2375	13418	blocked
24	132.0	42.0	145.1907	42.9088	5004	visible
25	253.0	51.0	257.5301	52.3912	3266	visible
28	299.0	8.0	296.1521	9.0023	30899	blocked
29	194.0	7.0	203.5152	8.4773	26544	blocked
32	289.0	40.0	285.2230	41.4383	3099	visible

Table A.13.: GNSS measurements for test site 3 - 15:47:57

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
03	1.0	1.0	352.3484	2.2156	21757	blocked
06	75.0	19.0	73.6386	20.5262	7166	visible
11	108.0	11.0	112.5021	12.3789	37078	blocked
12	56.0	86.0	283.1890	85.0763	812	visible
19	42.0	21.0	34.5325	20.1340	16337	blocked
24	132.0	42.0	145.4420	41.5580	5216	visible
25	254.0	52.0	258.2870	53.7455	2967	visible
28	299.0	8.0	296.4369	10.1533	29774	blocked
29	194.0	7.0	203.6547	9.7536	27150	blocked
32	289.0	40.0	283.8074	41.7908	3147	visible

Table A.14.: GNSS measurements for test site 3 - 15:50:57

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
03	1.0	1.0	351.2815	2.6670	30882	blocked
06	75.0	19.0	72.0558	21.0678	6946	visible
11	108.0	11.0	111.4489	13.3386	37525	blocked
12	56.0	86.0	291.6618	83.6228	837	visible
19	42.0	21.0	34.0574	19.0192	19026	blocked
24	132.0	42.0	145.7066	40.2128	5476	visible
25	254.0	52.0	259.0504	55.1031	2727	visible
28	299.0	8.0	296.7044	11.3110	29003	blocked
29	194.0	7.0	203.8082	11.0391	27732	blocked
32	289.0	40.0	282.3702	42.0864	3179	visible

Table A.15.: GNSS measurements for test site 3 - 15:53:57

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
03	358.0	2.0	359.9110	3.036	31665	blocked
06	74.0	21.0	75.5591	21.2600	21470	blocked
11	105.0	13.0	107.5251	13.7886	8855	visible
12	60.0	82.0	61.3082	83.5716	3016	visible
19	40.0	18.0	42.6103	18.5804	16290	blocked
24	133.0	38.0	134.9614	39.2524	8423	visible
25	255.0	55.0	258.0228	55.8829	4560	visible
28	300.0	11.0	302.3926	12.5553	9316	blocked
29	195.0	11.0	196.4097	11.5233	7301	visible
32	284.0	42.0	285.8814	42.4093	7262	visible

Table A.16.: GNSS measurements for test site 4 - 15:02:00

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
03	358.0	2.0	358.8292	3.4339	9142	blocked
06	73.0	21.0	74.2427	21.7673	10556	blocked
11	105.0	14.0	106.5665	14.7342	8637	visible
12	60.0	82.0	62.4279	82.1233	3122	visible
19	40.0	18.0	42.1540	17.4517	16446	blocked
24	133.0	38.0	135.4850	37.9175	8697	visible
25	257.0	55.0	258.9818	57.2461	4344	visible
28	300.0	12.0	302.7091	13.7261	9290	blocked
29	195.0	11.0	196.5092	12.8209	7125	visible
32	284.0	42.0	284.0806	42.6084	6925	visible

Table A.17.: GNSS measurements for test site 4 - 15:05:00

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
03	357.0	2.0	357.7343	3.7987	14277	blocked
06	73.0	21.0	72.9082	22.2377	10383	blocked
11	105.0	14.0	105.5877	15.6692	8400	visible
12	60.0	81.0	63.4266	80.6792	3225	visible
19	40.0	16.0	41.7364	16.3144	16346	blocked
24	133.0	37.0	136.0013	36.5884	8947	visible
25	257.0	56.0	259.9658	58.6121	4049	visible
28	300.0	13.0	303.0068	14.9034	9359	blocked
29	195.0	12.0	196.5092	14.1275	6846	visible
32	282.0	42.0	282.2556	42.7472	6594	visible

Table A.18.: GNSS measurements for test site 4 - 15:08:00

A. GNSS Measurement Data

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
03	355.0	3.0	356.6273	4.1301	16462	blocked
06	71.0	21.0	71.5570	22.6697	10311	blocked
11	104.0	15.0	104.5877	16.5923	7956	visible
12	61.0	80.0	64.3544	79.2397	3317	visible
19	40.0	15.0	41.3560	15.1697	16363	blocked
24	135.0	35.0	136.5102	35.2651	9390	blocked
25	258.0	58.0	262.0202	61.3513	3639	visible
28	300.0	14.0	303.5421	17.2776	9739	blocked
29	195.0	14.0	196.8609	16.7666	6802	visible
32	279.0	42.0	278.5565	42.8407	6279	visible

Table A.19.: GNSS measurements for test site 4 - 15:11:00

PRN	azim-real (deg)	elev-real (deg)	azim-sim (deg)	elev-sim (deg)	num pts	LoS
03	354.0	4.0	355.5091	4.4274	16504	blocked
06	68.0	22.0	70.1906	23.0621	10197	blocked
11	102.0	16.0	103.5654	17.5021	7481	visible
12	63.0	78.0	65.2380	77.8050	3398	visible
19	39.0	14.0	41.0117	14.0186	16243	blocked
24	135.0	34.0	137.0116	33.9477	9390	blocked
25	260.0	60.0	262.0202	61.3513	4049	visible
28	302.0	16.0	303.0068	17.2775	9359	blocked
29	195.0	15.0	196.5092	14.1275	6846	visible
32	278.0	42.0	282.2556	42.7472	6594	visible

Table A.20.: GNSS measurements for test site 4 - 15:14:00

# Bibliography

- Adjrad, M. and Groves, P. D. (2017). Intelligent Urban Positioning: Integration of Shadow Matching with 3D-Mapping-Aided GNSS Ranging. *Journal of Navigation*, 71(1):1–20.
- Bouzas, V., de Groot, G. J., Sajadian, M., Tzounakos, N., and Wu, T. (2018). Urban Horizon.
- Dana, P. H. (1994). Global Positioning System Overview. *The Geographer's Craft Project*.
- Dandurand, P., Babin, P., Kubelka, V., Giguere, P., and Pomerleau, F. (2019). Predicting GNSS satellite visibility from dense point clouds. *Research Gate*.
- Gandolfi, S. and La Via, L. (2011). SkyplotDEM - a tool for GNSS planning and simulations. *Applied Geomatics*, 3(1):35–48.
- Groves, P. D. (2011). Shadow Matching: A New GNSS Positioning Technique for Urban Canyons. *Journal of Navigation*, 64(3):417–430.
- Ledoux, H., Arroyo Ohori, K., and Peters, R. (2020). *Computation of modelling terrains*. TU Delft.
- Lesparre, J. (2023). Approximate transformation of geographic coordinates from Krayenhoff's triangulation and coordinates in Bonne projection used for historic map series of the Netherlands. *Kadaster, NSGI (draft)*.
- Lesparre, J., Huisman, L., Alberts, B., and de Ligt, H. (2018). RDTRANS2018. *NSGI*.
- Lu, Y.-H. and Han, J.-Y. (2020). GNSS SATELLITE VISIBILITY ANALYSIS BASED ON 3D SPATIAL INFORMATION IN URBAN AREAS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2020:123–128.
- Novatel (2015). *An Introduction to GNSS*. Novatel Inc.
- Suzuki, T. and Kubo, N. (2012). Simulation of GNSS Satellite Availability in Urban Environments Using Google Earth.
- Tiberius, C. and Verbree, E. (2004). GNSS positioning accuracy and availability within Location Based Services: The advantages of combined GPS-Galileo positioning. *2nd ESA/Estec workshop on Satellite Navigation User Equipment Technologies*, G.S. Granados (Ed), ESA publications division, pages 1–12.
- van Capel, M., Chontos, C., Gheorghiu, I., and Mbwanda, T. (2022). Availability of Galileo High Accuracy Services.
- van der Marel, H. (2020). *Reference Systems for Surveying and Mapping*. TU Delft.
- Verbree, E., Tiberius, C., and Vosselman, G. (2004). Combined GPS-Galileo positioning for Location Based Services in urban environment. *Symposium Location Based Services Telecartography*, page 15.

## *Bibliography*

Yuen, M. F. (2009). DILUTION OF PRECISION (DOP) CALCULATION FOR MISSION PLANNING PURPOSES.

Zhang, G., Verbree, E., van Oosterom, P., and Wang, X. (2017). Study of visibility analysis taking into account vegetation: An approach based on 3D airborne point clouds. *Proceedings of the 24th International Seminar on Urban Form: City and Territory in the Globalization Age*, page 13.

Zhang, G.-T., Verbree, E., and Wang, X.-J. (2021). An Approach to Map Visibility in the Built Environment From Airborne LiDAR Point Clouds. *IEEE Access*, 9:44150–44161.

## Colophon

This document was typeset using  $\text{\LaTeX}$ , using the KOMA-Script class `scrbook`. The main font is Palatino.



