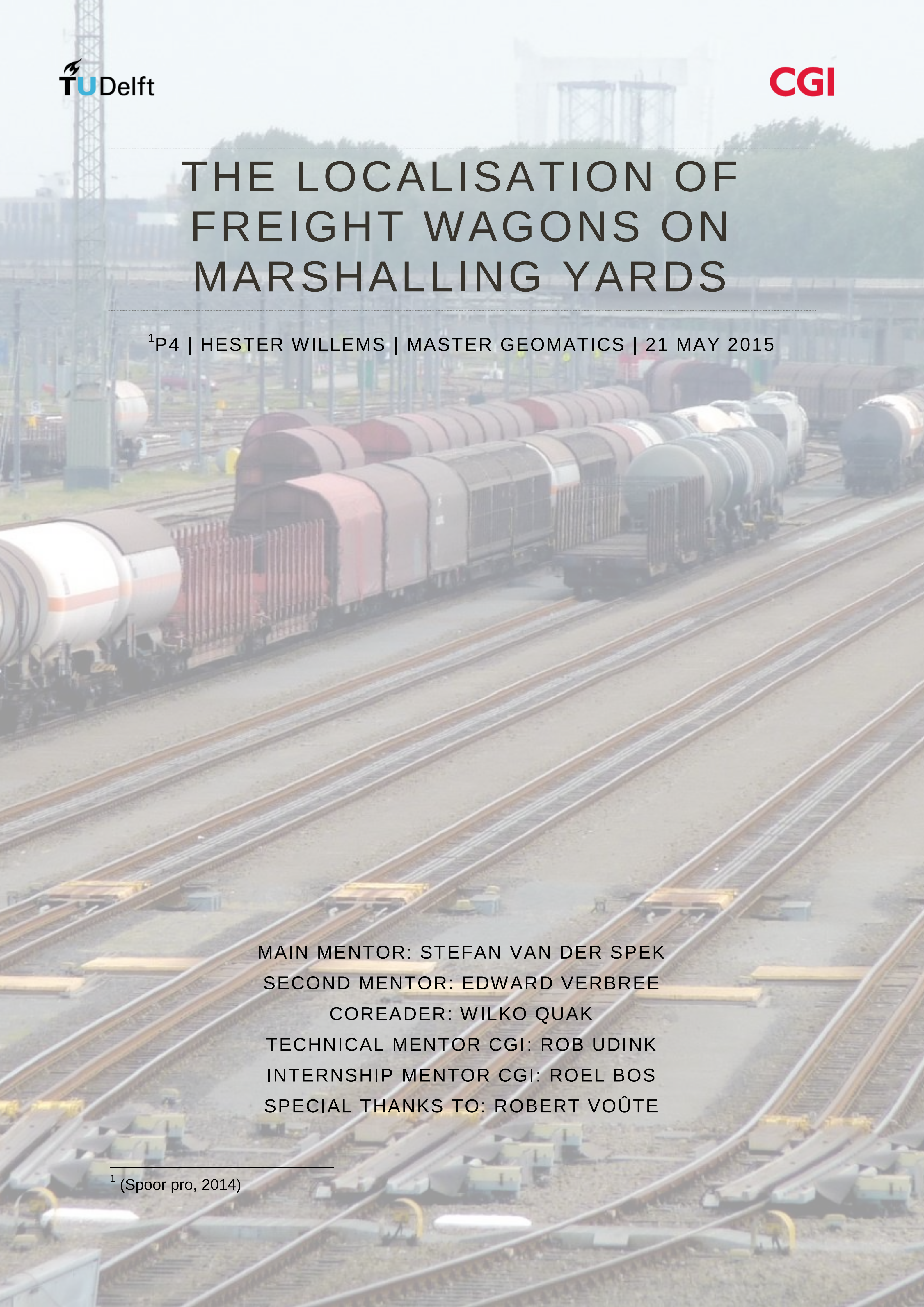


THE LOCALISATION OF FREIGHT WAGONS ON MARSHALLING YARDS

[1]P4 | HESTER WILLEMS | MASTER GEOMATICS | 21 MAY 2015

MAIN MENTOR: STEFAN VAN DER SPEK

SECOND MENTOR: EDWARD VERBREE

COREADER: WILKO QUAK

TECHNICAL MENTOR CGI: ROB UDINK

INTERNSHIP MENTOR CGI: ROEL BOS

SPECIAL THANKS TO: ROBERT VOÛTE

[1] (Spoor pro, 2014)

H.R.S. Willems

Student number: 4006992

Email: h.r.s.willems@student.tudelft.nl/hester.willems@gmail.com

June 25th, 2015, 9.15

Main mentor:

Ass.Prof.Dr. S.C. van der Spek

S.C.vanderSpek@tudelft.nl

Second mentor:

Ir. E. Verbree

E.Verbree@tudelft.nl

Co-reader:

C.W. Quak

C.W.Quak@tudelft.nl

Supervisors CGI:

Roel Bos

Roel.bos@cgi.com

Rob Udink

Rob.udink@cgi.com

Robert Voûte

Robert.voute@cgi.com

Master of Science Geomatics for the Built Environment – Delft University of Technology

The localisation of freight wagons on marshalling yards

Using multiple GPS measurements of stationary freight wagons to determine their location on a marshalling yard.

By H.R.S. Willems

ABSTRACT

Freight is transported more and more via trucks on the road, which has a negative influence on the environment. Transport via the railway system is environment-friendlier, but one of the main problems of transporting goods via the railway system, is that the process is inefficient and unpredictable, partly because trains can get lost. In an era where everybody walks around with a GPS device in their pockets, it seems strange that a large wagon can get lost for several weeks. To solve this problem, a suitable low-cost localisation technique has been researched and selected in order to localise freight wagons on marshalling yards. After investigation of six different techniques and consideration of and consultation with all stakeholders, GNSS was selected due to the great availability and the installation time and affordability. GNSS also posed an interesting research case towards the influence of taking previous measurements into account to improve the result of the GNSS localisation. A couple of characteristics of localising freight wagons on marshalling yards have been formulated. First of all, the vehicles will remain stationary during the largest part of their stay. This is the characteristic that makes investigation of the influence of previous measurements possible. Two methods have been tested to take these previous results into account, a median method, where the middle result of all results is taken, and an average method. The second characteristic of a marshalling yard is the fact that a freight wagon will always be on a certain track, and cannot move to another track without driving around the marshalling yard. During this project, the network of the marshalling yard has been taken as an extra reference system for the location of the freight wagons. After the previous results had been taken into account, the result is matched to the tracks of the rail network. The last characteristic is the fact that the order of the wagons will also remain the same while the wagons are standing still. It is possible to determine the order of the wagons based on the distance and bearing between the devices. The combination of the track number and the order of the wagons provides a description of the location that is understandable for all stakeholders. The system was build using the GeoEvent Extension for Server with custom java-developed processors.

After hardware selection and software development, the system was tested extensively. The reason to do these tests was twofold, first of all, the system was tested as a whole to see whether all different aspects worked. Secondly and most importantly, the system was tested to see whether it was possible to localise freight wagons on marshalling yard using a low-cost system, and to which extent this was possible. Using the raw GPS data, it would be possible to localise the freight wagon at a certain marshalling yard, possibly in a certain area on a marshalling yard. In raw data some outliers can be present, these are larger errors in the measurements and are removed to prevent influencing the result. After that, the previous measurements are taken into account, from about 15 minutes onwards to have some results to use as previous measurements. There was a different result when the median and when the average calculator was used. To get an idea of the performance, the performance after the map matching operation was analysed. The results of the average calculator were matched correctly in at least 51% of the cases and those of the median calculator in at least 65% of the cases. These numbers seem low, but get a lot more interesting when the time aspect of the system is taken into account. In the case of the median calculation, after about an hour of a half all results are always snapped to the correct track. After the results have been matched to a track, the bearing and distance will make it possible to calculate the order of the wagons on a single track.

ACKNOWLEDGEMENTS

First of all I would like to thank CGI, for welcoming me into their company and giving me continuous motivation at times when I needed it most. Roel Bos, Rob Udink, Robert Voûte, Ben van Tricht and all other members who showed interest and gave me a lot of help and useful input along the way. It was a pleasure working with you!

I also want to thank Edward Verbree for his help in making my rather practical problem a real research challenge and believing in me every step of the way. Stefan van der Spek, who helped me on a variety of things, from practical rules and regulations to the supply of GPS devices to perform my initial tests of the system.

I also want to wholeheartedly thank Thomas Buck for being there for me during the whole process and for sitting on different stations for hours to keep me company in the process or collecting data, thank you for believing in me!

Finally, many thanks also to Pauline van der Hoeven and Dick Willems, my lovely parents, who always believed in me, always gave me space to find my way and supported me in every step I took!

TABLE OF FIGURES

Figure 1 Examples of an open freight train and a car freight train (z24, 2013) (Schoubben & Barzeele, 2011)....16
Figure 2 The hill with wagons leaving (Samuel, 2011)....17
Figure 3 Wagons ordered by destination (Samuel, 2011)....18
Figure 4 Methodology (own picture)....20
Figure 5 The idea of RSS (Fuhr, 2008)....24
Figure 6 The principle of determining the angle of arrival with a ULA (Sahinoglu, Gezici, & Guvenc, 2008) (Hennigens, 2012)....25
Figure 7 The principle of TDOA to position a device (Carlsen & Petersen, 2010)....26
Figure 8 Cell identity positioning (Hennigens, Current approaches of WiFi positioning, 2012)....27
Figure 9 Trilateration (Thomas, 2013)....27
Figure 10 Triangulation based on the angle of arrival (Hennigens, Current approaches of WiFi positioning, 2012)....28
Figure 11 Direct path versus multipath (Djebouri, 2014)....28
Figure 12 Positioning principle of GPS (Japan Aerospace Exploration Agency, 2003)....30
Figure 13 Pseudo range GNSS (What When How, Unknown)....31
Figure 14 Carrier phase in comparison with pseudo code....32
Figure 15 Dilution of precision....33
Figure 16 Satellites further away from each other (left) vs closer together (right)....34
Figure 17 The principle of DGPS (Terrier, 2005)....35
Figure 18 The principle of RFID communication (RFID product, 2008)....37
Figure 19 The principle of active, passive and semi-passive tags (Dobkin, 2012)....37
Figure 20 The three different types of keying (Glover & Bhatt, 2006)....38
Figure 21 Form of a UWB pulse....43
Figure 22 point-to-point matching (Rote, 2010)....51
Figure 23 Point-to-curve matching (own image)....52
Figure 24 The whole rail network of the Netherlands (ProRail, Unknown)....53
Figure 25 Marshalling yard de Kijfhoek, Moerdijk (ProRail, Unknown)....53
Figure 26 Overview of the GeoEvent Service....56
Figure 27 The gateway (left) and A single Undagrid GPS node....60
Figure 28 Principle of Wireless Mesh Networks (Benyamina, Hafid, & Gendreau, 2012)....60
Figure 29 The request for the previous results....62
Figure 30 Median and Average Calculator....62
Figure 31 Percentage calculator....64
Figure 32 Calculating Distance & bearing....64
Figure 33 The result in the operations dashboard....65
Figure 34 Approximate location of the devices....68
Figure 35 Locations of the devices during all tests at the station of Geldermalsen....69
Figure 36 Testing at Geldermalsen station (own picture)....69
Figure 37 Raw data with outliers....71
Figure 38 Raw data without outliers....72
Figure 39 Variances in distances of raw data 9th of may 2015, Y-axis: distance (meters), X-Axis: time since first 20 results (minutes)....73
Figure 40 Average calculation all averages....74
Figure 41 Average result of test third of may....75
Figure 42 Average of test of the third of may by time....76
Figure 43 Variances in distances third of May, Y-axis: distance (meters), X-Axis: time since first 20 results (minutes)....77

Figure 44 Variances in distance test fourth of May, Y-axis: distance (meters), X-Axis: time since first 20 results (minutes) 77
Figure 45 Variances in distance test ninth of May, Y-axis: distance (meters), X-Axis: time since first 20 results (minutes) 78
Figure 46 Matched results of test ninth of May 79
Figure 47 Results matching coloured by time 80
Figure 48 Graphs of matched rail, Y-axis: distance (meters), X-Axis: time since first 20 results (minutes) 81
Figure 49 All median results 82
Figure 50 Median results of test third of May 83
Figure 51 Median results coloured by time 84
Figure 52 Variances in distance test of third of May, Y-axis: distance (meters), X-Axis: time since first 20 results (minutes) 85
Figure 53 Variances in distance Test Ninth of May, Y-axis: distance (meters), X-Axis: time since first 20 results (minutes) 85
Figure 54 Test third of may matched results 86
Figure 55 Matched results coloured by time 87
Figure 56 Graphs of matched rail, y-axis: distance (meters), x-axis: time since first 20 results (minutes) 88
Figure 57 Geoinformation chain (Lemmens, 2013) 107
Figure 57 Variances in distances of raw data 9th of may 2015, Y-axis: distance (meters), X-Axis: time since first 20 results (minutes) 119
Figure 58 Variances in distances of raw data 3rd of May 2015, Y-axis: distance (meters), X-Axis: time since first 20 results (minutes) 119
Figure 59 Variances in distances of raw data 4th of may 2015, Y-axis: distance (meters), X-Axis: time since first 20 results (minutes) 120

TABLE OF TABLES

Table 1 Frequency ranges (Glover & Bhatt, RFID Essentials, 2006) 39
Table 2 Comparison of techniques based on performance parameters 45
Table 3 Example X and Y 82

Abstract 5
Acknowledgements 7
Table of figures 9
Table of tables 11
Contents 13
1 Introduction 15
1.1 Problem description 15
1.2 Background 16
1.3 Objectives 18
1.3.1 MoSCoW method 18
1.4 Research questions 19
1.5 Methodology 19
1.6 Research outline 20
2 Literature study 23
2.1 General theory and definitions 23
2.1.1 Positioning vs localisation 23
2.1.2 Active vs passive positioning and active vs passive devices 23
2.1.3 Parameters 24
2.1.4 Computation methods 26
2.1.5 Multipath 28
2.2 Possible localisation methods 28
2.2.1 Performance parameters 29
2.2.2 Possible applicable techniques 29
2.3 Choice of localisation method 45
2.3.1 Other research on GPS and Train Localisation 46
3 Software 49
3.1 Marshalling yard characteristics 49
3.1.1 Stationary vehicles 49
3.1.2 Rail network 50
3.1.3 Order of the wagons 54
3.1.4 Comparable projects 54
3.2 GeoEvent Extension for ArcGIS for Server 55
4 Implementation 59
4.1 Hardware description 59
4.1.1 Wireless mesh networks 60
4.2 Software 61

4.2.1 Input from Undagrid server 61
4.2.2 Processors description 62
4.2.3 Output visualisation 64
5 Tests and analysis 67
5.1 Testing scheme 67
5.1.1 Original scheme 67
5.1.2 Actual Scheme 67
5.1.3 RD versus WGS84 70
5.2 Analysis of raw data 70
5.3 Analysis of average calculation 73
5.4 Analysis of the median calculation 81
5.5 Calculating the order of the wagons 88
5.6 Conclusions testing 89
5.7 Practical applicability 90
6 Conclusions and future work 93
6.1 Conclusions 93
6.1.1 Localisation techniques 93
6.1.2 Does the system work (tests and conclusions) 94
6.1.3 Final conclusion 95
6.2 Future work 96
7 Bibliography 99
8 Appendices 105
A. Plans to test the system 105
B. Reflection 106
C. Source codes GeoEvent processor 108
B1 Input Transport Undagrid data 108
B2 Average or median processor 108
B3 Map matching processor 111
B4 Percentage on track processor 113
B5 Order of wagons processor 116
D. Test results 118
C1. Graphs of raw data 118

1 INTRODUCTION

1.1 PROBLEM DESCRIPTION

The fact that freight wagons get lost sounds strange, but transport companies do actually loose freight wagons and locomotives during transport. Not only are these wagons and locomotives very expensive and therefore costly to loose, losing them also makes the transportation of goods on the rail very inefficient. The fact that this is a very relevant problem at this moment could be seen only a month ago in an article on the NOS website. In this article it became clear that transport over road is actually growing instead of shrinking, as companies do not want to use rail transportation. Rail transportation is too inefficient and too unreliable. A literal quote from this article is:

"… There are also a lot of complaints about lacking service on the rail. According to Sitskoorn (Joost Sitskoorn, organisation of tenders of freight EVO red.) *a freight train can easily be lost for a week. 'You don't believe this in the time of GPS, but it is the case…'* (NOS.nl, 2015)

There are more examples to underline the need for localisation of freight wagons on marshalling yards. On the 14th of January 2011 there was a fire on the marshalling yard of Rotterdam, the Kijfhoek. This fire was near a coach loaded with ethanol, which posed a dangerous situation. Fortunately, the fire was extinguished and no harm was done, but very dangerous situations could come from unclear information on marshalling yards. But during the fire the location of the other wagons and their load was investigated by the Inspectie voor Verkeer en Waterstaat (IWV). This investigation showed that the information about the trains on the marshalling yard did not match the actual situation. After this conclusion, the IWV decided to do an investigation on 40 marshalling yards in the Netherlands, where similar conclusions had to be drawn (Inspectie van Verkeer en Waterstaat, 2011).

These two examples show that it is very important to get a better overview on the location and load of the freight wagons. Missing or unclear information on the location of freight wagons and locomotives is a problem in rail transportation. New technologies may help to provide this information in an efficient and transparent way. A large part of the problem takes place on marshalling yards, where the information is fragmented and often not correct. A research on 40 marshalling yards in the Netherlands from the IVW found that the information present at marshalling yards is often incomplete or unclear. Managers often have trouble providing information about trains with hazardous materials and can only give information if a train number is already known. Another problem is that there are often no rail numbers on the marshalling yard to provide a reference point. Locomotives that are leased by a transporter often cannot be linked to a transporter and can therefore also not serve as a reference point. The information itself also falls short; the manager can often not give the rail occupation or the presence of dangerous materials. If extra wagons are added to a train, the manager often only has information about the original train. Documents of the transporter are frequently not available or not complete and do not match with the information in online registration system, Online Vervoer Gevaarlijke Stoffen (OVGS). The actual situation on emplacements does not always match the documents of the transporter or the information in the OVGS (Inspectie van Verkeer en Waterstaat, 2011).

The lack of information on the location of wagons on marshalling yards causes a couple of problems; first of all, incomplete information about the location of hazardous materials can pose a dangerous situation in cases of emergency, not only for train- and emplacement employees, but also for the employees of emergency services and for the surrounding area.

A second problem is that wagons regularly get lost on marshalling yards. This happens all over the world and costs a lot of money for transporters. In France, train companies were desperate because more than 150 wagons where lost in a couple of years, so they set out a reward to employees that found one of the wagons. The costs of the lost wagons in France was about one million euro's per year (NOS, 2013).

The localisation of freight wagons on marshalling yards will be researched in this master thesis at the company CGI, an international IT company that focusses on business consulting, system integration and IT outsourcing services. CGI has a GEO-ICT department, where I worked during this project.

BACKGROUND

The transportation of goods is very important in every economy, so also in the Netherlands. The harbour of Rotterdam receives about 440 million tons of goods every year, and all those goods have to be distributed all over Europe (Corporate Communications, 2013). Multiple means of freight transportation are possible, transportation on the road, river, plane or sea are all possibilities, but this thesis focusses on rail transportation. Transportation by rail has the advantage that a lot of goods can be transported in one go, for which only one driver is needed. Also, it uses a lot less energy than for example transportation by trucks. A disadvantage is that not every location is reachable by train, because rails do not lead everywhere (Kamer van Koophandel, 2014).

To develop a system to track freight trains on marshalling yard, it is important to gain some inside knowledge about the process of rail transportation and the different processes that take place on marshalling yards. Freight trains consist of a locomotive and a number of wagons. The locomotive pulling the train can either run on electricity or diesel or a combination of those two. One of the processes that take place on marshalling yard is changing the locomotives, for example from electrical to diesel or to one that can handle a larger voltage. There are a lot of different types of wagons, open wagons for iron ore and coal, tank wagons for liquids, platforms on which containers can be placed, wagons especially made for car transportation and many more (Belinter Logistic, 2012) . Container wagons are relatively new and have the advantage that different types of goods can be loaded into the same container, and that loading them on the freight train is relatively easy. A freight train in the Netherlands can have a length of maximum 680 meter, excluding the locomotive(s).

FIGURE 1 EXAMPLES OF AN OPEN FREIGHT TRAIN AND A CAR FREIGHT TRAIN (Z24, 2013) (SCHOUBBEN & BARZEELE, 2011)

A lot of stakeholders are involved in the transportation of goods via rails, for example the national rail-traffic controller ProRail, the driver, the yardman, the transporter and the wagon examiner, to name a few (Samuel, 2011).

Ideally when transporting goods, one wagon can be assembled with a single type of freight that only drives back and forth over the same route. Unfortunately, this is rarely the case, usually different types of freight with different departures and destinations have to be combined and arranged. Marshalling yards, hubs and emplacements exist for this purpose. Hubs are larger marshalling yards placed on strategic points to reach a lot of destinations and origins and emplacements are small marshalling yards at a company's premises.

Goods are usually loaded into or onto the wagons at the premises of a company or at a harbour or hub. The wagons will be moved to a marshalling yard nearby as soon as possible. There are a couple of processes that take place on marshalling yards.

1. The locomotive can be changed, on marshalling yard a smaller and non-electric locomotive can be used, so the electric or large locomotive is decoupled and will be parked on a separate rail and a smaller diesel locomotive will take its place.
2. It can also be necessary for the train to change direction. The locomotive will be decoupled, drives around the marshalling yard and is attached to the train on the other side.
3. The main processes on marshalling yards consist of waiting for all wagons to arrive and combining them. There are several ways to arrange the wagons. First there is the pull or push system, where a locomotive drives back and forth to move trains from one track to another. A more efficient way is to use a hill, the train moves up this hill and releases the wagons one by one. A computer system arranges the switches to move the wagons to the appropriate track and automatic brakes will make sure the wagons do not crash. (Samuel, 2011). Five minutes before departure at the latest the lists with the wagons have to be submitted to the Online Vervoer Gevaarlijke Stoffen (OVGS) (Tenpierik, 2014).

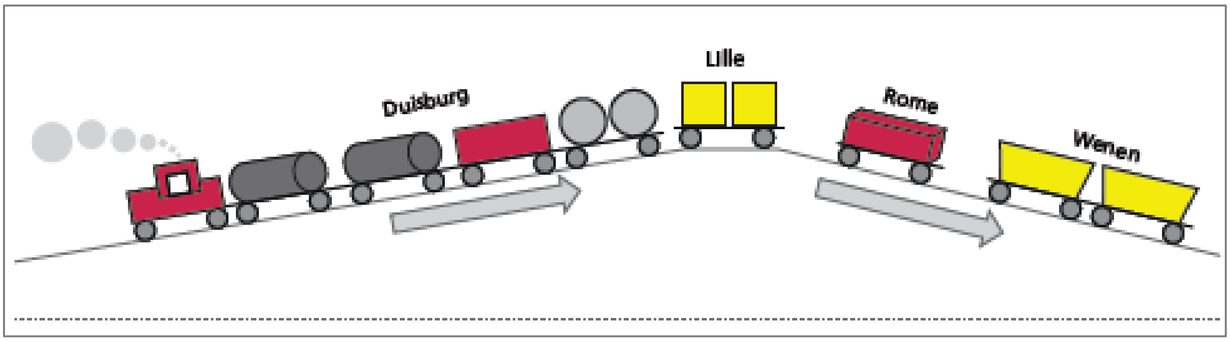


FIGURE 2 THE HILL WITH WAGONS LEAVING (SAMUEL, 2011)

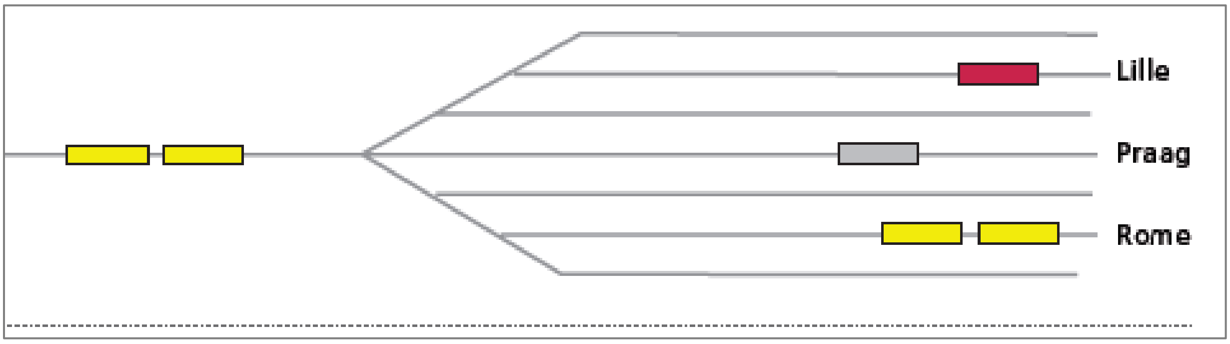


FIGURE 3 WAGONS ORDERED BY DESTINATION (SAMUEL, 2011)

1.3 OBJECTIVES

The overall goal of this project is to be able to know where which freight wagon is located on the marshalling yards in the Netherlands. A system developed by another intern at CGI focusses on the localisation of the train that is not on a marshalling yard but driving around. The system should either be able to locate a wagon on request and/or provide a live map or schematic overview showing the location of different wagons and their loads.

The goal is to discover the best method to localise freight wagons on marshalling yards, to implement this method in a system and to validate the results. This system should show where a certain freight wagon is located on the marshalling yard, by means of a track number and the order of the wagons on one track. This basic system could be used to store more data, for example about load, times of arrival and departure and even whether a wagon still has to be arranged. In order to achieve this goal, several objectives are proposed:

- Objective 1: find the most suitable solution to localise trains on marshalling yards
- Objective 2: find the best set up to localise trains on marshalling yards (based on the chosen method of positioning)
- Objective 3: have a working system architecture with simulated data from the chosen positioning method
- Objective 4: validate the method and get the system architecture working
- Objective 6: give a scale on which the wagons can be located (track number and order)

1.3.1 MOSCOW METHOD

In a project, it is important to have a good overview over what is absolutely necessary to do in order to make it a success. It is not only important to have a clear set of requirements, but also to rank them. This helps everyone to understand which requirements are most important. A method called MoSCoW can help in this situation. MoSCoW stands for must, should, could and would. The following definitions can be given to it (Haughey, 2011):

M	Must have this requirement for a successful result
S	Should have this requirement if possible, but the project success does not rely on it
C	Could have this requirement if it does not affect anything else in the project
W	Would like to have this requirement later, but it will not be delivered this time.

Knowledge about main methods: GPS, RFID, NFC, UWB, Wi-Fi and Bluetooth	M
Knowledge about secondary methods: Video Recognition, Infrared, Barcodes and Ultrasound	S
Knowledge about current situation:	
• Knowledge process of train transport	S
• Knowledge processes on marshalling yards	M
• Knowledge other cases	S
Knowledge of performance parameters	M
Snap locations to rail	M
Snap wagons so they do not overlap	C

Improve location by implementing multiple measurements	M
Have a working infrastructure	M
Test the infrastructure	M
• On a marshalling yard	S
• Combined with the other part of the system	C
Push the system online	M
• On a map	C
• In a systematic overview of the wagons	C
Have a pull system for stakeholders	C
Other options	
• Send a warning when something is wrong	C
• Automatically submit the train composition online	W

1.4 RESEARCH QUESTIONS

Based on the goal and the objectives, a research question can be formulated, along with several sub questions. This research question is the following:

To what extent is it possible, using current and affordable positioning techniques, to localise freight trains on marshalling yards in such a way that it can be determined on which shunting track the coach is located and what the order of the wagons is?

A shunting track is defined as the part in between the switches at the entrance of the marshalling yard and the switches at the exit. This part of the marshalling yard only consists of straight tracks

There are also a couple of sub questions that are defined to split the research question into manageable pieces that give structure to the project.

1. What techniques could be used for the localisation of freight wagons and what are the advantages and disadvantages?
2. What localisation technique would be optimal to use in this particular case, based on its performance parameters?
3. What would be the best set up to use for the chosen method?
4. Does the method and system architecture work? Does it generate the expected results?
5. On what scale can freight wagons be found using the system?

METHODOLOGY

The methodology of this project consists of several steps, which can be found in Figure 4.

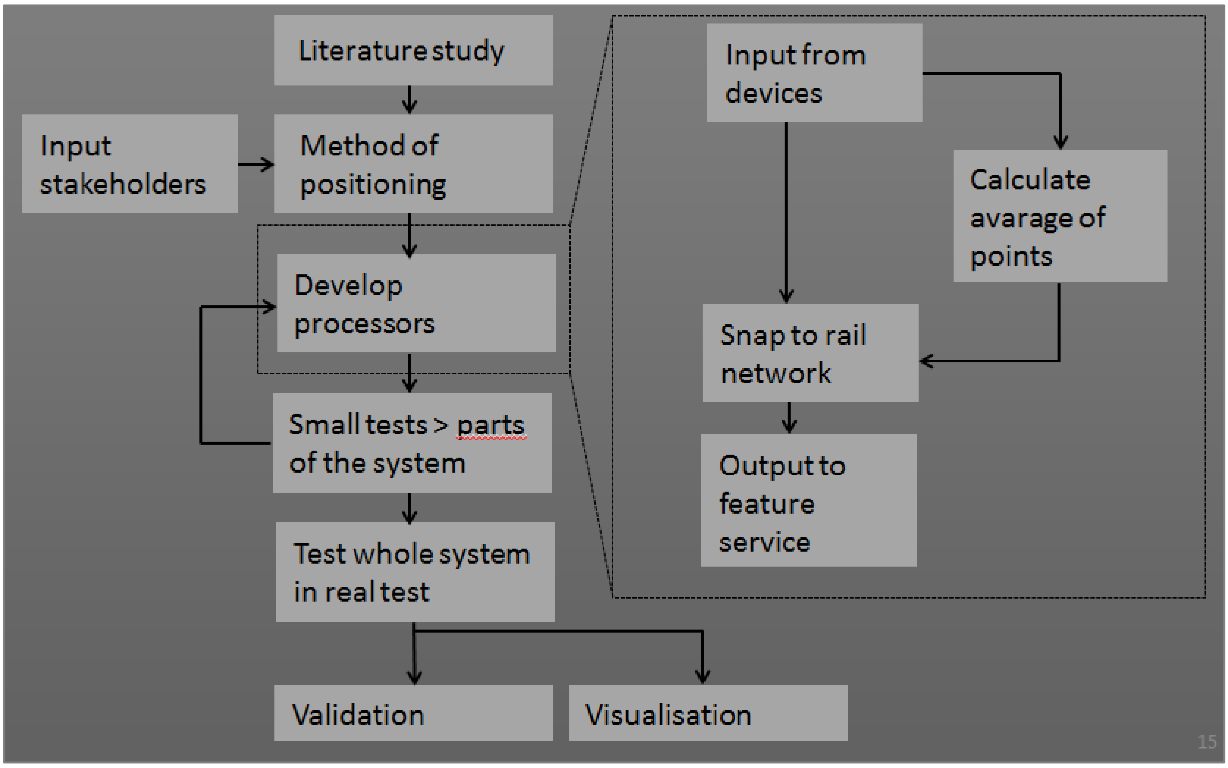


FIGURE 4 METHODOLOGY (OWN PICTURE)

The project started with information collection by means of a literature study on several themes. Information about the process with freight trains and the context on the project was gathered and researched, but more importantly a large literature study on the possible localisation techniques was performed. Based on this literature study, the appropriate technique was chosen in consultation with all stakeholders involved. During the information collection phase other cases were also researched. Because research on systems like this for rail transportation is barely available, similar systems in freight and people transportation have been researched instead.

After the choice for a localisation method had been made, the algorithms necessary for processing the data have been created, along with a lot of small tests of parts of the system. This was an iterative process of developing and testing the different algorithms.

The whole system was tested on a simulated marshalling yard on the station of Geldermalsen and the data gathered there was analysed and validated afterwards to get an answer to the research question.

1.6 RESEARCH OUTLINE

This research contains three main parts, the choice of the appropriate system, the implementation of this system and the tests of the system on (simulated) marshalling yards.

Chapter 2: Literature study

In this chapter, an extended summary of the literature study towards the best localisation technique is given. In this chapter six possible techniques are described along with the choice and reasoning for that choice. Research towards other cases is also described in this chapter.

Chapter 3: Software

This chapter starts with a description of several characteristics that are important when localising freight trains on marshalling yards. These characteristics are very important for the development of algorithms and the analysis of the test results in this case. Apart from a description of these characteristics, a way to use them is also described, as well as the tool that was used to process the measurements real time, the GeoEvent Extension.

Chapter 4: Implementation

In this chapter, a description of the hardware is given. The different processors that were developed are described extensively.

Chapter 5: Tests and analysis

The chapter starts by a description of the test setup, followed by the tests and an analysis of the tests result and some initial test conclusions. Also a view on the practical availability is given.

Chapter 6: Conclusion and future work

In this chapter, the project is concluded and answers to the research questions are given. After that, work that could continue the research on this topic is stated.

2 LITERATURE STUDY

2.1 GENERAL THEORY AND DEFINITIONS

Before the different localisation techniques can be discussed, a couple of theories, definitions and methods of computing a location that are often used have to be explained, in order to make the discussion of localisation techniques possible. Some error sources that can appear are also discussed. The theories and error sources that will be discussed in this paragraph are: passive versus active positioning, some parameters that can be used (angle of arrival, time of arrival, received signal strength and time difference of arrival), some computing methods (trilateration, triangulation and cell identification) and some error sources (multipath).

Most localisation and positioning techniques work with electromagnetic signals that are send out and received and based on the signals that a certain device receives, it can determine its own position or the position of the device sending out the signal. The electromagnetic waves can have different frequencies, a well-known example of an electromagnetic wave is visible light or ultraviolet light. Examples of commonly known devices that work with electromagnetic signals are radios and mobile telephones.

Devices can have two roles in the process of positioning with electromagnetic waves, they can be fixed stations, devices that remain on the same spot and have a fixed position in a certain reference systems. The second role for a device is to be a rover, a device that can move around, have an unknown location and this location is determined using the positioning technique in question.

2.1.1 POSITIONING VS LOCALISATION

Even though positioning and localisation are two words that are often used interchangeably, in this context the definitions differ slightly. Positioning is a term that is used when a device or feature is pinpointed on the map. With positioning, the result is an exact location on a map with a certain accuracy and precision. This location will be given with coordinates in a particular coordinate reference system.

Localisation refers when something is said about the location, without giving actual coordinates. The location can be given in everyday text, instead of in coordinates. Examples of locations can be room 5.267 in a building or rail segment 83, in other words, identifiable places.

In this project, the goal is a combination of the two. As the goal is to give information to the different stakeholders which is easy to understand, localisation will be the main aim of this project. It will have to be possible to say on which track a wagon is and which wagon is in front or after the wagon in question. However, because the results will also be depicted on a map, positioning will also be necessary, as coordinates in a certain coordinate system are needed to be able to visualise the result on a map.

2.1.2 ACTIVE VS PASSIVE POSITIONING AND ACTIVE VS PASSIVE DEVICES

Within positioning and localisation techniques, a distinction is often made between active and passive. This is a very complicated distinction, as it can refer to two different aspects of positioning, active and passive positioning and active and passive devices. Both distinctions will be explained here.

First, there is the distinction between active and passive positioning techniques. With active positioning, both devices actively take part in the determination of the position of the moving device. In this case, the location will be calculated by the moving device, e.g. a smartphone, based on a signal sent by the device with known position, e.g. a satellite.

In passive positioning, the moving device does not actively take part in the localisation process. This can happen when a device sends out signals constantly, usually for another purpose than localisation. These signals are intercepted by a fixed station that can determine the position of the device without interfering with it. The difference with active positioning is that with passive positioning, the device that is being localised may not be aware that this is happening, as it does not have to make a connection. With active positioning, the mobile device is actively involved and cannot be unaware that positioning is taking place.

Some localisation devices can participate in both active and passive positioning, and it depends on the application which one will be used. An example of a technique that can work with both active as passive positioning is Wi-Fi/Bluetooth localisation (Hennigens, 2012) (Glover & Bhatt, 2006) (Hunt, Puglia, & Puglia, 2007).

The difference between active and passive devices is that active devices send out a signal and passive devices only listen to signals that are sent out by other devices and possibly reflect the signal. The distinction between these two and between the two different versions of active and passive will become clearer when the different localisation techniques are described.

2.1.3 PARAMETERS

There are a number of parameters that are used in multiple positioning techniques, therefore it is convenient to define them here. These parameters are Received Signal Strength (RSS), Angle of Arrival (AOA), Time of Arrival (TOA) and Time Difference of Arrival (TDOA).

2.1.3.1 RECEIVED SIGNAL STRENGTH (RSS)

Received Signal Strength (RSS) is based on the fact that radio waves spread out evenly and gradually decrease in signal strength with increasing radius (Sahinoglu, Gezici, & Guvenc, 2008). The signal strength decreases because of energy transformation that takes place when the signal travels through the air or through other materials. This energy transformation causes the amplitude of the signal to become smaller and smaller. In an ideal situation this decreasing is log-normal, as is visualised in Figure 5.

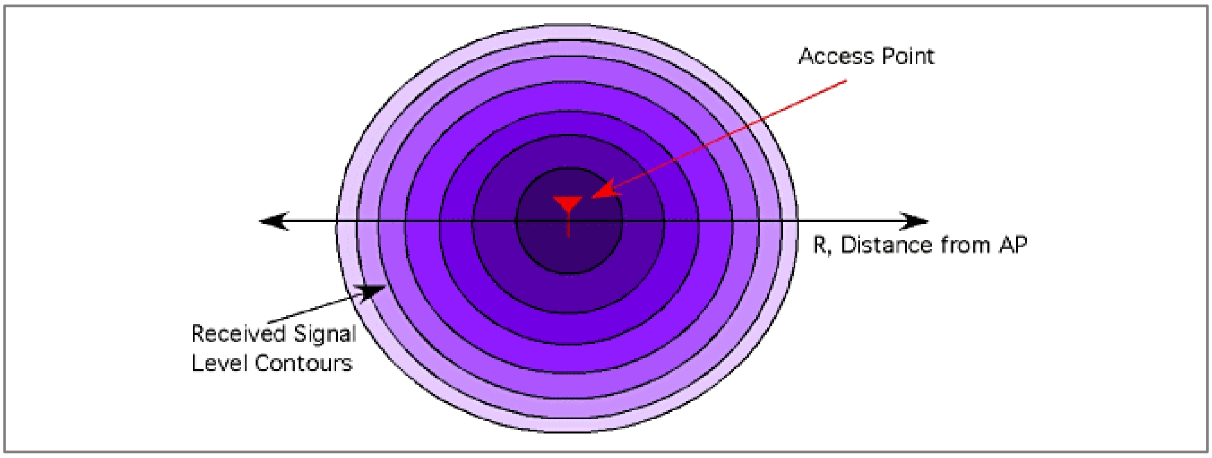


FIGURE 5 THE IDEA OF RSS (FUHR, 2008)

However, the ideal situation usually does not occur because of the influence of the context on the signal strength. The wave can bump into walls, windows, trees etc., these bumps cause the signal to weaken. In theory you would be able to compute the distance between location and target based on the signal strength, but the influence of materials can cause errors in the positioning accuracy. These errors will increase in areas where a lot of obstacles are present. Even a human body can influence electromagnetic signals, as radiation will partially be absorbed by the water in the body. When a lot of changes occur in the context of the signal, for example the arrival, arranging and departure of freight wagons, the RSS will not be predictable and it might be difficult to use it in such a way that accurate positioning can take place (Hennigens, 2012) (Zhang, Li, & Zhang, 2009).

2.1.3.2 ANGLE OF ARRIVAL (AOA)

Another parameter is the Angle of Arrival (AOA). This parameter provides information about the direction of the incoming signal. By using multiple antennas instead of one, it is possible to see which antenna received the signal first. This makes it possible to calculate the angle of arrival by the differences in arrival time of an incoming signal at the different antenna elements, as is visualised in Figure 6.

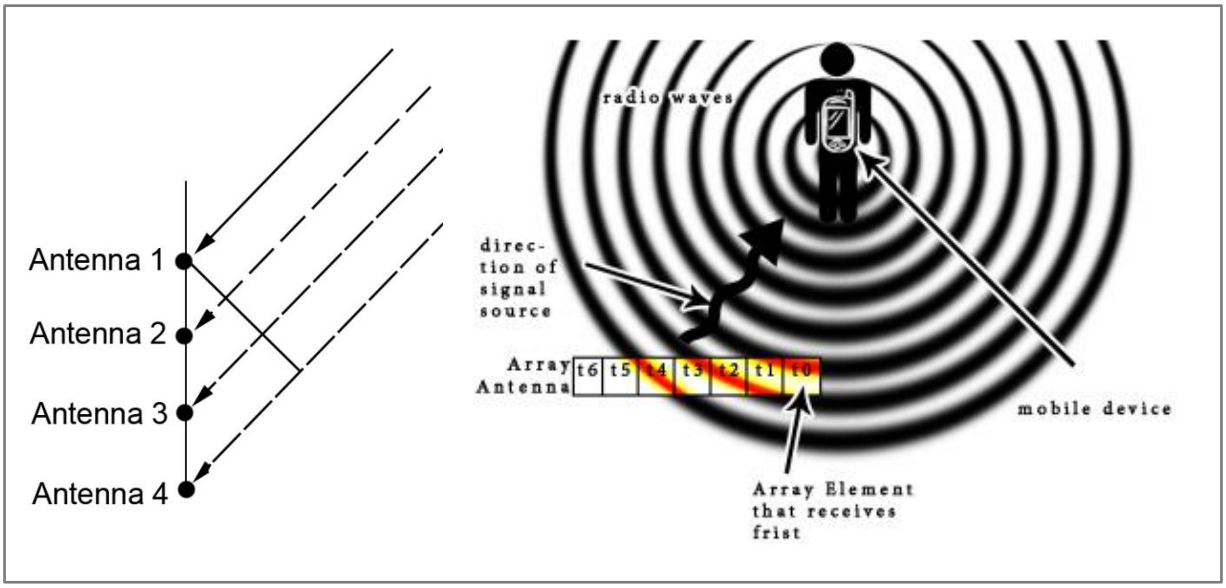


FIGURE 6 THE PRINCIPLE OF DETERMINING THE ANGLE OF ARRIVAL WITH A ULA (SAHINOGLU, GEZICI, & GUVENC, 2008) (HENNIGENS, 2012)

There are different arrays in which the antennas can be organised. The simplest and commonly used is the Uniform Linear Array (ULA), which can be seen in Figure 6. Other possibilities include Uniform Circular Arrays (UCA) and rectangular lattices. These are just different ways of arranging the antennas, the principle of determining the AOA will remain the same (Hennigens, 2012) (Sahinoglu, Gezici, & Guvenc, 2008).

2.1.3.3 TIME OF ARRIVAL (TOA)

The third parameter is the Time of Arrival (TOA). In this method the distance between the two devices is calculated using the estimated time of flight of the signal between the two devices. To determine the travel time, the time on the clocks of both devices has to be equal, or time information has to be exchanges via certain protocols. As different techniques determine the time of departure and arrival of the signal in slightly different ways, this will be elaborated in the description of the techniques. The accuracy of TOA measurements can be improved by increasing the Signal to Noise Ratio (SNR) and/or the effective signal bandwidth

(Sahinoglu, Gezici, & Guvenc, Ultra-wideband positioning systems, 2008). When the travel time is known, the distance can be calculated, because electromagnetic waves travel with the speed of light, 299.792.458 meter/second, the distance is defined by the speed of light times the travel time (Penrose, 2004) (Martin, Liu, Covington, Pesti, & Weber, 2010).

2.1.3.4 TIME DIFFERENCE OF ARRIVAL (TDOA)

The last parameter is Time Difference of Arrival (TDOA). If time synchronisation between the target and reference nodes is absent but there is synchronisation between two reference nodes, TDOA measurements can be obtained. The difference of arrival time of two signals travelling between the target node and two reference nodes is estimated, and this locates the target on a hyperbola which has the foci on the two reference nodes. By adding a second hyperbole that is determined by the TDOA of one of the original two nodes and a third one, the position of the device can be determined as can be seen in Figure 7.

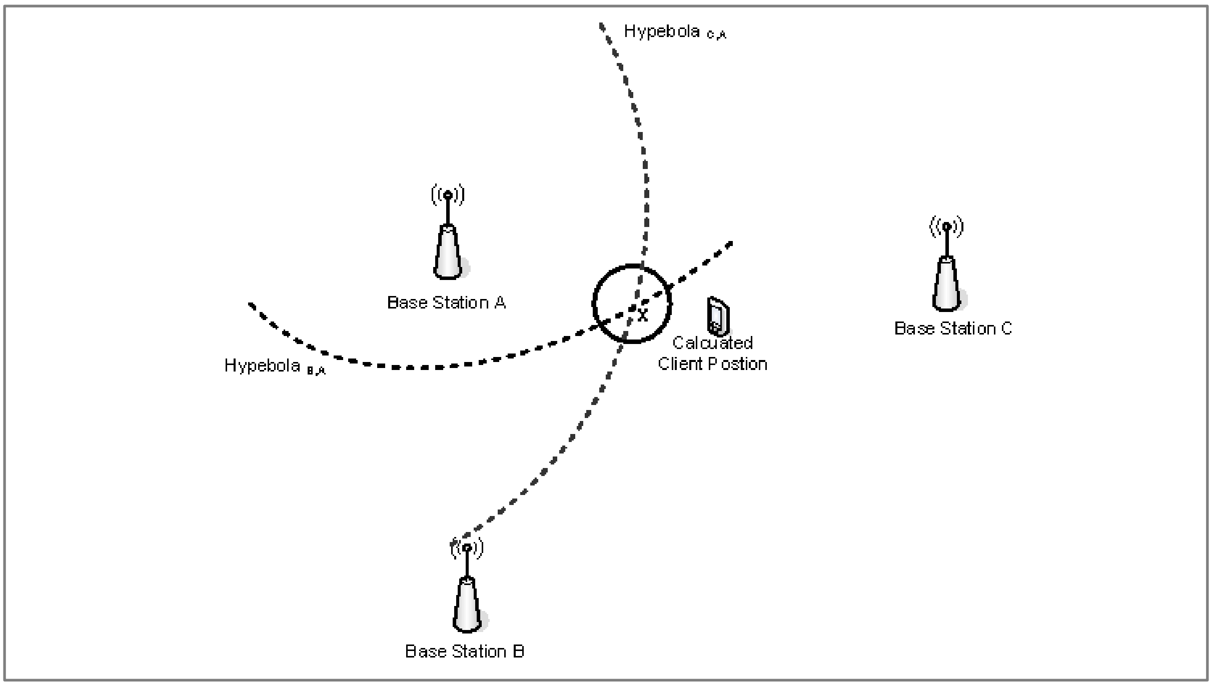


FIGURE 7 THE PRINCIPLE OF TDOA TO POSITION A DEVICE (CARLSEN & PETERSEN, 2010)

The TDOA measurement can be obtained in two ways. The first is to estimate TOA at each reference node and then to obtain the difference between the two estimates. The target and reference times are not synchronised, so the estimated TOA will not be correct, however, the two reference stations are synchronised, so the offset in TOA will be the same for both signals. The TDOA measurements in this case will, like the TOA measurements, increase in accuracy with bandwidth and SNR. Another way to determine the TDOA measurement is to perform cross-correlations of the received signals and calculate the delay corresponding to the largest cross-correlation value. This works well for single-path channels, but the performance degrades significantly over multipath channels or noise (Carlsen & Petersen, 2010) (Martin, Liu, Covington, Pesti, & Weber, 2010).

2.1.4 COMPUTATION METHODS

There are a couple of computation methods that are not technique dependent and used in multiple positioning and localisation techniques. These methods are cell identification, trilateration and triangulation.

2.1.4.1 CELL IDENTIFICATION

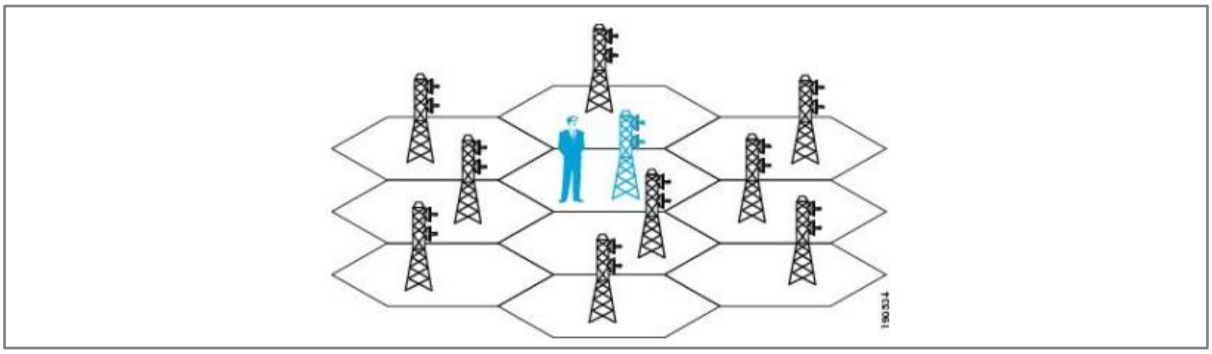


FIGURE 8 CELL IDENTITY POSITIONING (HENNIGENS, CURRENT APPROACHES OF WIFI POSITIONING, 2012)

For cell identification, only one station needs to receive the signal send out by the device or connect with it to perform localisation. There are two methods to determine the correct cell. The first is to use signal strength to determine to which station the device is closest and give it that cell identity. The second method is only applicable when the device connects to the station, the station to which the device is connected yields the result. Both methods are quite simple and exist of only a couple of queries, but unfortunately, the location that is determined is very inaccurate because the device can be anywhere within the cell it was matched to. A disadvantage when using the second method is that a device will not always connect to the closest access points, but sometimes use one that is further away. Because of this, the accuracy can be even lower than the cell size (Hennigens, Current approaches of WiFi positioning, 2012) (Martin, Liu, Covington, Pesti, & Weber, 2010).

2.1.4.2 TRILATERATION

Trilateration is a computation method that can be used when the distance between the two devices is known. As mentioned in paragraph 2.1.3, there are multiple ways in which the distance between the two nodes can be estimated. Trilateration is based on the fact that the moving device needs to be somewhere on the circle with a middle point of the reference station and a radius of the calculated distance. If you know this of two circles, they will have two intersection points, so two possible positions of the device. If a third circle is added, only one point will remain, as can be seen in Figure 9. In order to perform trilateration on a surface, so 2D, the distance from the moving node to at least 3 fixed stations has to be known. If trilateration in 3D, so including height, is required, at the distance to at least 4 stations needs to be present.

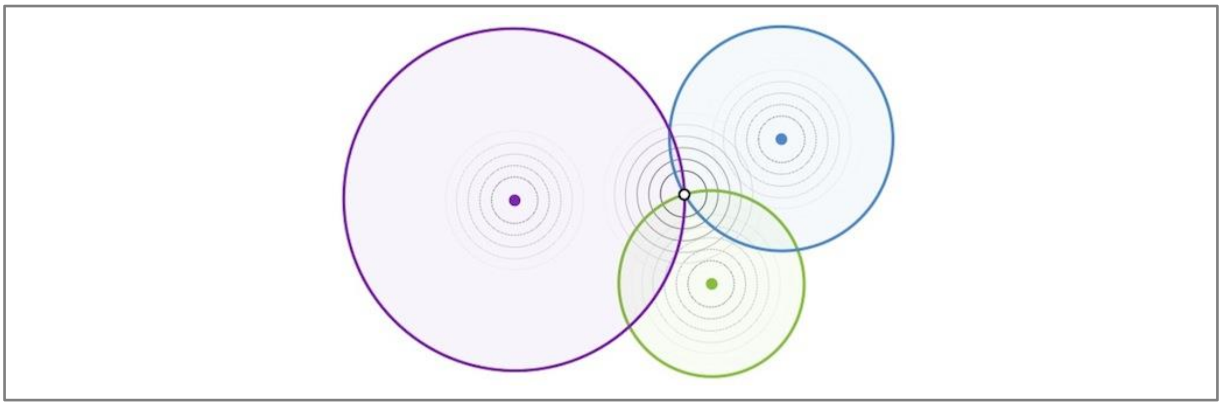


FIGURE 9 TRILATERATION (THOMAS, 2013)

Because the distance is only an estimation, an offset is added to each of the circles, resulting in a small area in which the device can be (Martin, Liu, Covington, Pesti, & Weber, 2010).

2.1.4.3 TRIANGULATION

Triangulation can be applied when the AOA for a signal from at least two reference stations and the locations of these reference stations are known. The lines defined by the reference stations and the angles intersect each other and the point that that intersection returns, is the location of the device. This is illustrated in Figure 10.

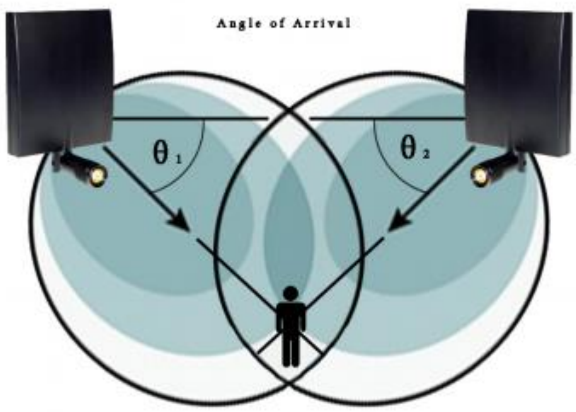


FIGURE 10 TRIANGULATION BASED ON THE ANGLE OF ARRIVAL (HENNIGENS, CURRENT APPROACHES OF WIFI POSITIONING, 2012)

2.1.5 MULTIPATH

A general error source that is present when using electromagnetic signals to determine locations is multipath. Multipath errors happens when a signal does not reach a receiver directly, but reflects from (several) surface(s) before it arrives at the receiving antenna. This means that the time it took to travel will be longer, the RSS will be lower and the AOA will not be correct. The concept of multipath is depicted in Figure 10.

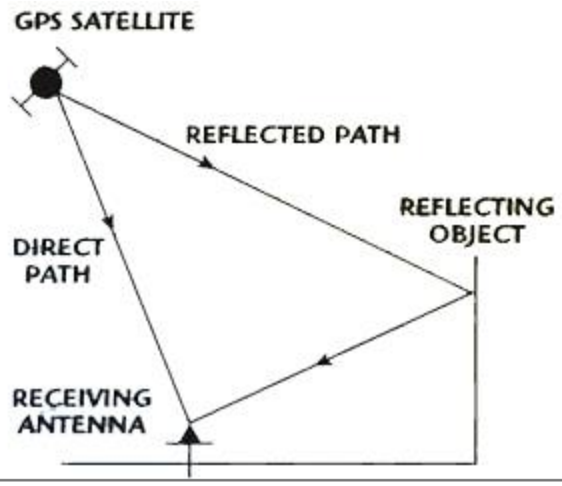


FIGURE 11 DIRECT PATH VERSUS MULTIPATH (DJEBOURI, 2014)

2.2 POSSIBLE LOCALISATION METHODS

In order to find a wagon on a marshalling yard, a localisation technique has to be used. To find the appropriate technique, a literature study has been done towards six localisation techniques. In order to properly compare these techniques, the performance parameters as developed by the Civil Aviation Community will be used. In this paragraph the performance parameters will be discussed, followed by a description of the different localisation techniques. In the next paragraph the choice of localisation method will be discussed. This paragraph will provide an answer to the second, third and fourth research question.

2.2.1 PERFORMANCE PARAMETERS

In 1998 in the Civil Aviation Community, a new air traffic control system (PANS-OPS) included a performance-based navigation system called Required Navigation Performance (RNP), where the performance for operation on the route is defined (Universal Avionics, 2013). The parameters of this system can also be used in the analysis of the performance of positioning systems, as navigation and positioning have similar performance requirements. The Required Navigation Performance method consists of four main parameters (European Space Agency, 2014).

Accuracy: the accuracy is determined by how much the measured position matches the true position. Accuracy is a statistical measure, so a statement of the uncertainty in a position should be included in the accuracy statement. In the literature study, accuracy will be described using levels, for example mm-cm-dm levels. This is because the description of accuracy differs a lot in the available literature.

Availability: the availability is the percentage of the time that the system is usable in the coverage area. Availability also includes signal availability. Signal availability is defined by the percentage of time the signals, transmitted by external sources, are available to use. In this measure, both the physical characteristics of the environment and the technical capabilities of the transmitter are important.

Continuity: this is the possibility of the system to perform its task without interruption during an operation.

Integrity: the last parameter represents how much the information supplied by the system can be trusted. It also includes the ability of the system to warn the users when the system should not be used.

2.2.2 POSSIBLE APPLICABLE TECHNIQUES

For each of the possible techniques, first the principle and technique will be described, followed by a description of the performance parameters and concluded by the suitability in this case.

2.2.2.1 GNSS/GPS

THE PRINCIPLE AND TECHNIQUE

Global Navigation Satellite Systems (GNSS) provide satellite based positioning, which is the determination of positions of observing sites on land or at sea, in air and in space by means of artificial satellites (Hofmann-Wellenhof, Lichtenegger, & Wasle, 2008). The best known GNSS system is the Global Positioning System (GPS), which was originally developed by the army of the United States. GPS has grown to be a positioning system used all over the world for all types of applications.

With GNSS, a device with a GNSS receiver can determine its position on the earth based on the signal received from the satellites. This calculation is done using TOA measurements. In order to know the TOA, satellites have very precise atomic clocks on board, but as the receiver usually does not have such a precise clock, the exact time difference is part of the calculation, as will be described later. In the almanac of the receiving device, the theoretical positions of the satellites are available. Corrections to these positions are send to the device as well.

The position of the device is calculated using trilateration, see Figure 12. The signal of at least four satellites is needed to get a 3D location, but more satellites will improve the accuracy of the position as it will eliminate some of the errors (Ogaja C. , 2011).

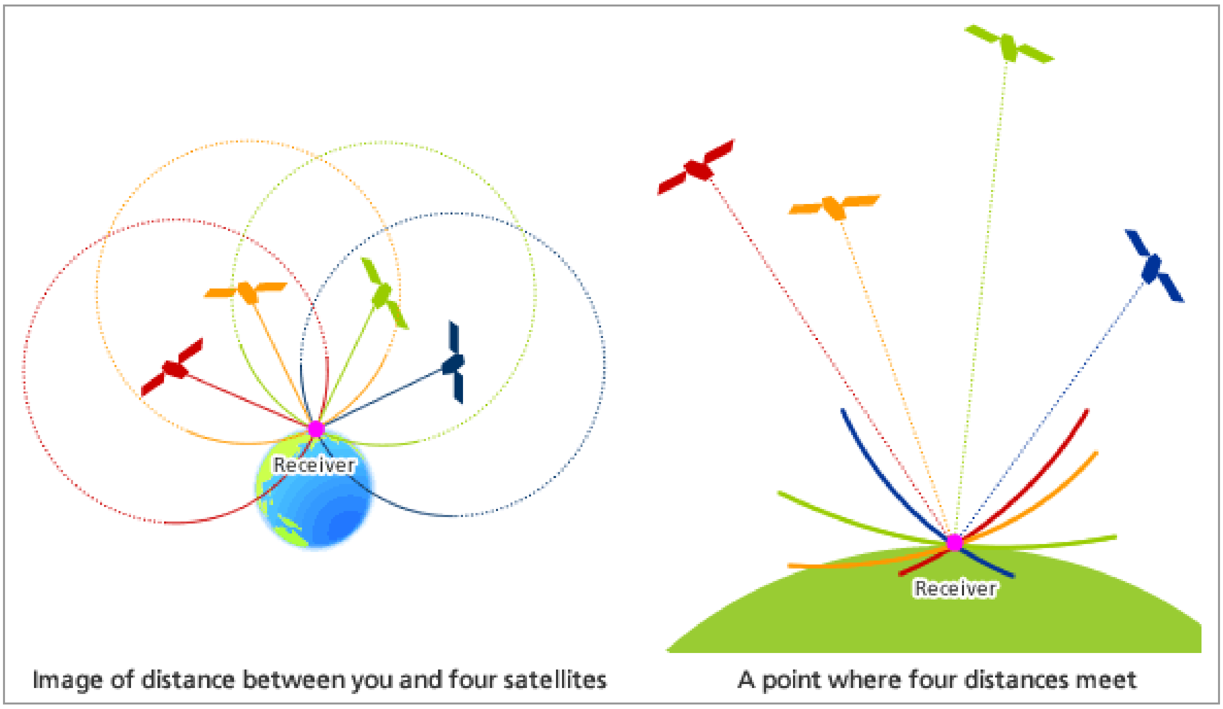


FIGURE 12 POSITIONING PRINCIPLE OF GPS (JAPAN AEROSPACE EXPLORATION AGENCY, 2003)

THE TECHNIQUE: PSEUDORANGE VS CARRIER PHASE

GNSS is based on TOA measurements. There are two ways to extract the instant of transmission from the signal.

Pseudo range GNSS

Pseudo range GNSS compares the pseudo-random-signal that it receives from the satellite with an identical signal it generates itself and measures the offset between the signals, as can be seen in Figure 13. This offset is directly related to the travel time, which can be determined using this offset. The satellite sends out this random signal every millisecond, and once the receiver recognises the signal, it starts generating the same signal (Warner & Johnston, 2003). The receiver then matches the signal and finds delays caused by the travel time. Though this method is quite easy, it does not yield results that are as accurate as carrier phase measurements do. The reason for this is that the signals can be out of phase, while still seeming to match. This is because the signal consists of 0 and 1, and the values can be the same, even though the signals are not completely in sync. This problem could lead to accuracy problems of up to the meter level.

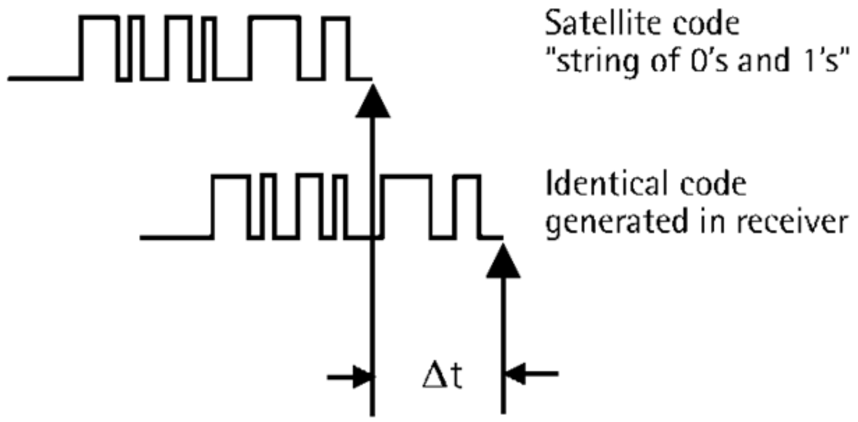


FIGURE 13 PSEUDO RANGE GNSS (WHAT WHEN HOW, UNKNOWN)

The GNSS observations are given by the observation equations. First, the observation equation for pseudo range GPS will be given and then it will be explained step by step.

$$P = \rho + c(dt - dT) + t_r + \varepsilon_P$$

where:

P	is the ionosphere-free pseudorange
ρ	is the geometrical range computed by iteration as explained next
dt	is the station receiver clock offset from the GPS time
dT	is the satellite clock offset from the GPS time
c	is the speed of light in vacuum
t_r	is the signal path delay due to the neutral atmosphere (troposphere)
ε_P	is the noise component, including multipath

Now a step-by-step explanation will follow, it all starts with the actual observation to the satellite s, which can be written as:

$$P = (T - T^S)c$$

Where T is the known reading of the receiver clock, T^S is the reading of the satellite clock when the signal was transmitted and c is the speed of light. This observation can be developed by setting the clock time T to the true receive time t plus a clock bias τ. The receiver clock bias exists because the clocks on the satellites and the clock in the receiver are not synchronised, there is an offset in time. While the satellite clock is an atomic clock, which is very expensive and very precise, the clock in a receiver is usually a lot less expensive, and thus less precise. Because of this offset, the unknown time offset will be the fourth variable in the position calculation, X, Y and Z of the receivers are the first three unknowns. The receiver clock offset is receiver specific, so it is the same for all satellites. Because of this fourth unknown variable, a signal from at least four satellites is necessary to determine position. The receiver clock error can be determined by adjusting the positions based on the fourth available satellite (Warner & Johnston, 2003) (Calais, Unknown). When this bias is added, this results in the following equations:

$$T = t + \tau$$

$$T^S = t^S + \tau^S$$

This gives the following new observation equation:

$$P^S = \big((t + \tau) - (t^S + \tau^S)\big)c$$

$$= (t - t^S)c + c\tau - c\tau^S$$

$$= \rho^S(t, t^S) + c\tau - c\tau^S$$

Where $\rho^S(t, t^S)$ is the range from the receiver to the satellite. This model is still simplified, influences from the atmosphere are not yet included in this equation. By iteration, the geometrical range can be computed via the following formula:

$$\rho^S(t, t^S) = \sqrt{(x^S(t^S) - x(t))^2 + (y^S(t^S) - y(t))^2 + (z^S(t^S) - z(t))^2}$$

Adding the error factors for signal path delay and the noise component, will give the final observation equation as described earlier, which can be written for every satellite separately (Kouba, 2003) (Blewitt, 1997).

Carrier phase GNSS

Carrier phase GNSS not only compares the pseudo range of the signal but also the carrier frequency. The carrier frequency is much higher than the signal frequency, and this can therefore result in a much higher accuracy. Because the carrier frequency is so much higher, the cycles so much shorter and the syncing of the signal received by the device and the signal the device self-produced can be much more accurate. The problem with this method is that all waves are uniform which makes it difficult to match the signals. The pseudo range measurements can be used to get the matching close and the carrier phase measurements to determine the time difference more exact. The accuracy that can be achieved this way is in the millimetre level (Henry, 2014)!

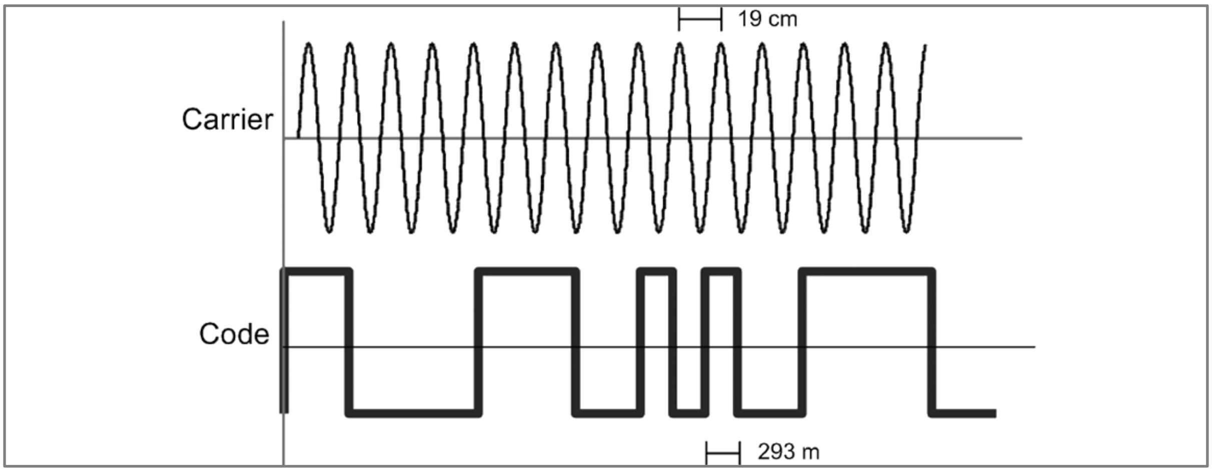


FIGURE 14 CARRIER PHASE IN COMPARISON WITH PSEUDO CODE

For carrier phase GNSS, a similar observation equation can be written to the one for pseudo range GNSS. This results in the following equation:

$$P = \rho + c(dt - dT) + t_r + N\lambda + \varepsilon_\varphi$$

with the additional parameters:

N	is the non-integer ambiguity of the carrier phase combination
$\lambda_1, \lambda_2, \lambda_3$	are the of the carrier-phase L1, L2 and combination wavelengths
ε_φ	is the measurement noise component

AVAILABILITY

GPS has 31 satellites in orbit around the earth, of which at least 24 are transmitting signals at 95 percent of the time. It is available everywhere in the Netherlands, so no new systems have to be installed. Only GPS receivers are necessary to determine locations. Differential GPS, where the received signal is compared to that of a known reference station, will also need infrastructure with a reference station or the use of DGPS networks that are offered by several companies. The availability will only increase in the future, as a third batch of GPS satellites will be launched into orbit in the near future (National Instruments, 2014). With the new European GNSS Gallileo, the Chinese Beidou and the Russian Glonass also developed or in development, the availability of GNSS will only increase.

ACCURACY

The accuracy of GPS depends greatly on the method that is used to determine the position. As mentioned before, the accuracy of pseudo range GPS is in the order of meters to tens of meters and that of carrier-phase GPS can decrease to the millimetre level. The accuracy can be determined theoretically based on a principle called the Position Dilution of Precision (DPOD) (Langley, 1990). DPOD has to do with the configuration of the satellites. If the satellites occupy a large volume in the sky, the accuracy is higher and when the satellites are closer together, the accuracy is lower. An ideal geometry is when four satellites or more are distributed evenly in the sky. The position dilution of precision is depicted in Figure 15.

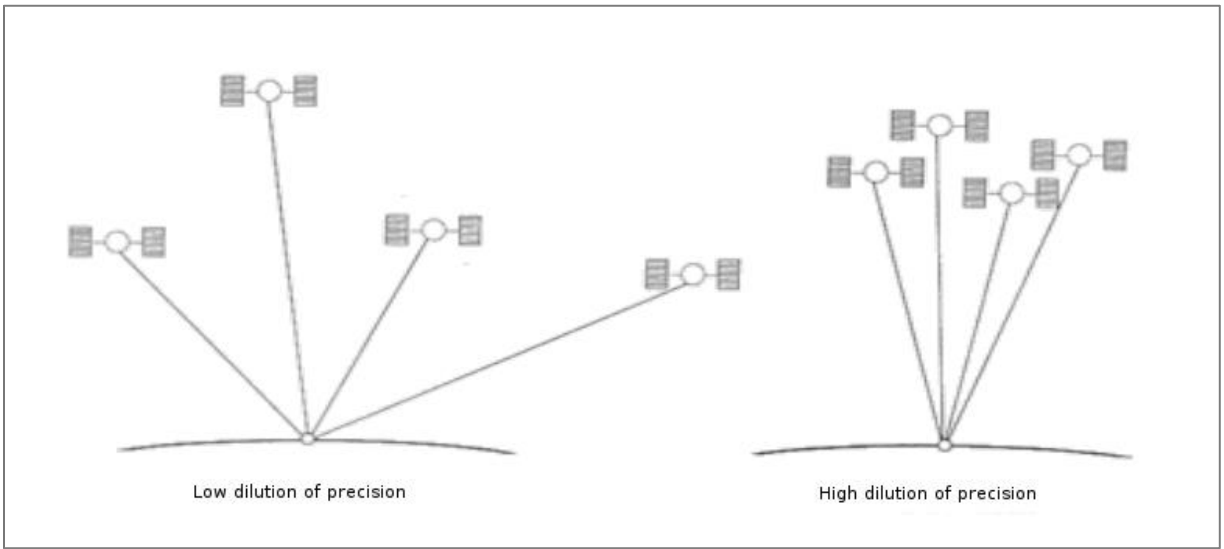


FIGURE 15 DILUTION OF PRECISION

When the visible satellites are distributed more evenly in the sky, the calculated position will be better. This is because the area, caused by the offsets of the system, of the actual position is larger when all satellites are closer together. This principle can be seen in Figure 16.

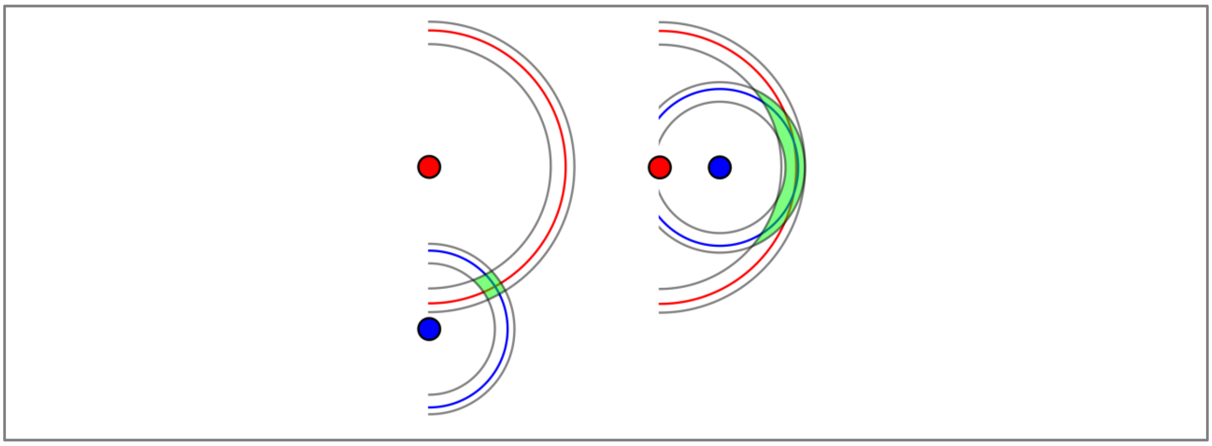


FIGURE 16 SATELLITES FURTHER AWAY FROM EACH OTHER (LEFT) VS CLOSER TOGETHER (RIGHT)

ERROR SOURCES IN GPS

Multipath can be a problem in GNSS positioning, especially in urban environments, but there are some other GNSS specific error sources, which will be shortly discussed here.

Orbital (ephemeris) errors

The theoretical orbits of satellites are known and stored in almanacs on the GPS receivers. But unfortunately these theoretical orbits are not always completely correct. This is monitored by the Master Control Station in Colorado. The control station also models the satellite orbits and this information is used by the GPS receivers to determine the location of the satellites. However, it is impossible to perfectly model the satellite orbits, which causes satellite ephemeris errors.

Satellite clock errors

This is the difference between the satellite clock time and the true GPS time, because even with the high-quality atomic clocks, satellite clocks errors are unavoidable.

Atmospheric errors

The ionosphere consists of electrically charged particles or ions. Because of the free ions in this layer, the GNSS signals do not travel on the speed of light in this region. The distances are calculated too long, because of the delay of the signal in this layer. The delay that this layer causes is dependent on time, season, and geographic location and has major influences from solar activity and the geomagnetic field. The troposphere causes a delay which is a function of the elevation and altitude of the receiver. This error is dependent on many factors like the atmospheric pressure, temperature and water vapour content.

IMPROVING GPS ACCURACY

One way to improve the accuracy of GNSS is to calculate the location a lot of times. If the GNSS receiver stays on the same spot for a certain amount of time, many results can be modelled and compared and the result will get closer to the true position.

Differential GPS positioning

Differential GPS (DGPS) positioning determines the position based on two receivers simultaneously. One of the two receivers is the base receiver and is set up at a reference station of which the coordinates are known. The other receiver, called the ranger, moves around or stands on a location from which the coordinates are not known. The base receiver will compare the position that is calculated based on GPS measurements with the known location to be able to use the errors in the measurement. If the ranger is less than 20 km from the base station this data can be used to improve the calculated position of the ranger.

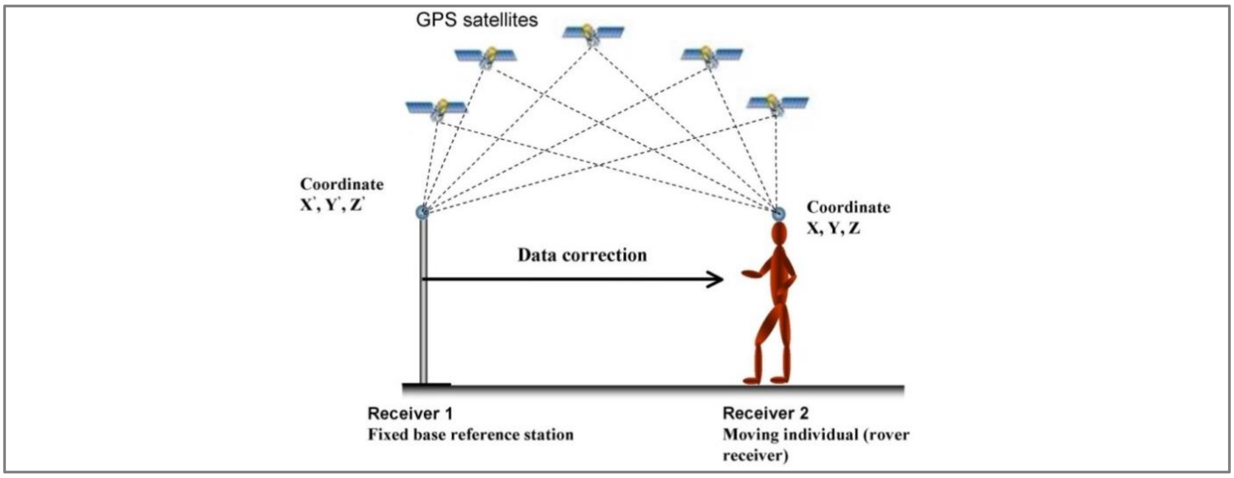


FIGURE 17 THE PRINCIPLE OF DGPS (TERRIER, 2005)

CONTINUITY

The US requires an availability of at least 24 operational GPS satellites for 95 percent of the time. To ensure this, there are 31 operational GPS satellites in orbit the last few years (National Coordination Office for Space-Based Positioning, Navigation and Timing, 2014). The continuity of GNSS as a whole will be larger because other satellite systems are also available, apart from GPS.

INTEGRITY

GPS has several checks and balances that are built into the system in order to ensure a fairly high level of integrity. This means that there are systems that will recognise failure in the positioning of GPS (Heng, XingXin Gao, Walter, & Enge, 2011). In order to monitor the integrity of the GPS constellation, a system called JPL is developed. This operates the only global ground network that is capable of inferring the satellite state in real time globally and to separate clock performance from orbit performance. This network tracks the satellites, each GPS satellite is tracked with at least as many as 10 tracking stations simultaneously. The average number of tracking stations tracking one satellite is 25 (California Institute of Technology, 2013). Unfortunately, GPS integrity failures have occasionally occurred even with several systems installed (Heng, XingXin Gao, Walter, & Enge, 2011).

Two issues that can influence the integrity of GPS are spoofing and jamming. Jamming is the process where the GPS signal is blocked by a countersignal, which will be noticed by receiver because the receiver will register the missing GPS signals. Spoofing, however, is much more difficult to recognise. With spoofing, a special device will send out a fake GPS signal. The receiver will read this signal and assume this is the correct satellite GPS signal, it will then ignore the real satellite signals and calculate position based on the spoofed signal (Warner & Johnston, 2003).

2.2.2.2 SUITABILITY OF GNSS

There are several advantages and disadvantages of using a GPS based system to track trains on marshalling yards.

An advantage of GPS is its availability, there is no need to set up a new network, as GPS is always available. This makes the implementation quite easy, as practically no infrastructure is needed on the marshalling yards.

A disadvantage of GPS is the inaccuracy of the system. The accuracy of GPS differs for the different methods, better methods are usually used in more expensive devices. One or the demands for the system is that it should be low cost. The most accurate systems will therefore not be an option. The accuracy of point positioning is in the meter to tens of meters level and the accuracy of DGPS is somewhere in the meter level. As the railway tracks lay 4.5 - 5.3 meters apart, both these methods might not generate a high enough accuracy. Another way to improve the accuracy, possibly even to a level in which the track can be determined, is to use multiple measurements. The more measurements you have, the more accurate the result will be. As the wagons will stay on the marshalling yard for a certain amount of time, this could make it possible to use cheaper GPS devices but still get the accuracy that is needed. The drawback of this method is that some time is needed to determine gather enough measurements, so if the location of a wagon is requested when it has just arrived on the marshalling yard, the chance of the position to be faulty will be larger.

Another disadvantage of GPS is the fact that satellites have to be visible from the receiver. This means that if the receivers would be placed at the bottom or on the side of a freight train, the train itself might block a lot of the available satellites. Devices should therefore always be placed as much on the top of the wagons as possible.

2.2.2.3 RADIO FREQUENCY IDENTIFICATION (RFID) & NEAR FIELD COMMUNICATION (NFC)

THE PRINCIPLE AND TECHNIQUE

Radio Frequency Identification (RFID) is a system where an electronic device is attached to an item and uses radio frequency or magnetic field variations to communicate (Glover & Bhatt, RFID Essentials, 2006). RFID systems typically consist of three components. The first component is a tag, which is attached to the item and is composed of a semi-conductor chip, an antenna and potentially a battery.

The second component is a reader, sometimes also called an interrogator. The reader has an antenna, an RF electronics module and a control electronics module and can recognise the presence of tags and read the information they send back or reflect. The last component is the controller, also known as middleware or host. This is the software on the system with which the reader communicates (Glover & Bhatt, 2006) (Hunt, Puglia, & Puglia, 2007).

Near Field Communication (NFC) is a new technique that is based on RFID. This technique is developed to exchange personal data in a secure manner. The system is basically the same as that of RFID, but certain security regulations have been implemented to make sure transactions are secure. In NFC, the tag and reader only couple when within a couple of centimetres from each other. A special feature of NFC is the fact that it can act in Pear-to-Pear (P2P) communication, or Reader/Writer Mode. In this mode, can be exchanged data efficiently with minimum delay (Yadav & Sharma, 2014).

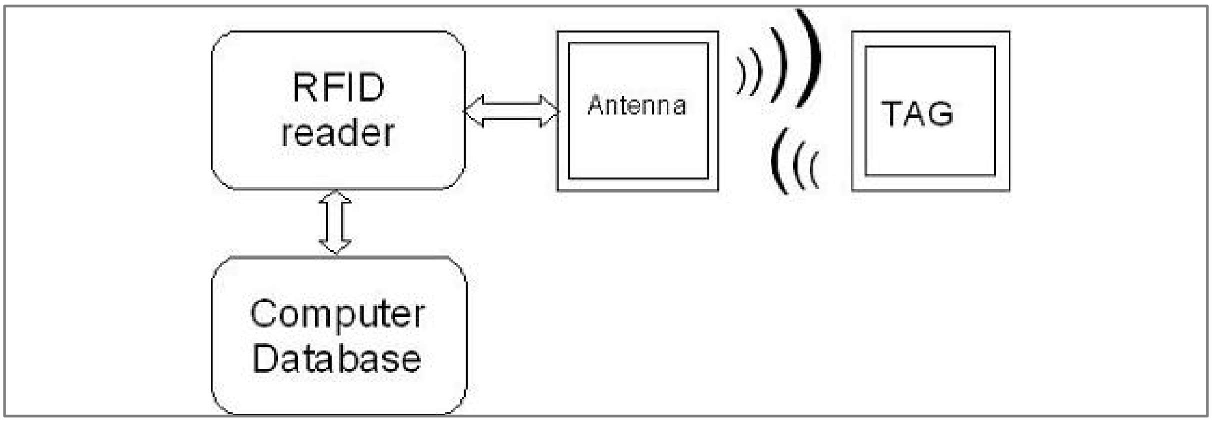


FIGURE 18 THE PRINCIPLE OF RFID COMMUNICATION (RFID PRODUCT, 2008)

ACTIVE VS PASSIVE

RFID tags can be passive or active. Passive tags do not have their own power supply, but get powered by the signal they receive, and use that small amount of power to alter the signal and reflect it. There are also passive tags where batteries are on board but are used for other applications or on-board electronics, for example to monitor other sensors on the devices.

Active RFID tags have their own power supply that they use to transmit a signal to the reader. Because the active tag will send its own signal, it can transmit information over a longer range and can exchange information with less powerful readers as the reader does not have to power the passive tag. Also, these types of tags usually also have larger memories. The technology behind active RFID tags is more complex and the tags are larger than the passive tags. Recent advances in chip technology do have resulted in a lower price for these tags, making it possible for them to compete with passive tags (Glover & Bhatt, 2006) (Hunt, Puglia, & Puglia, 2007).

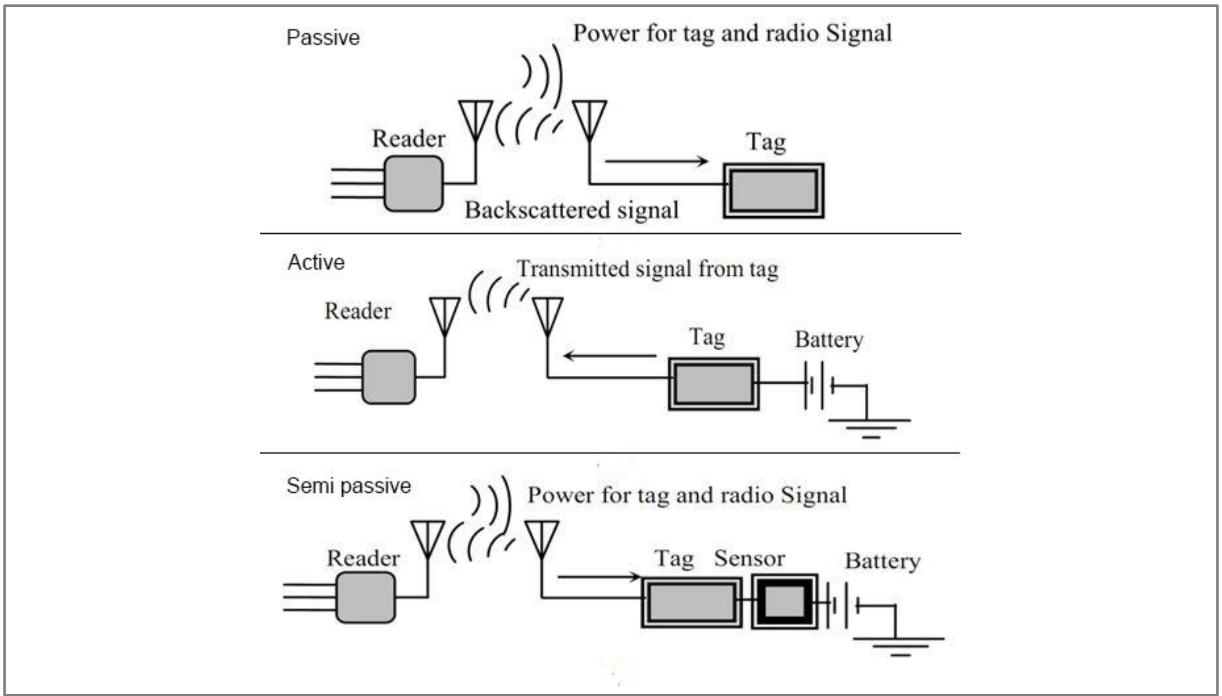


FIGURE 19 THE PRINCIPLE OF ACTIVE, PASSIVE AND SEMI-PASSIVE TAGS (DOBKIN, 2012)

Another type of tag is the semi-passive tag, these power themselves to make it possible for the reader to use all power for communication instead of sharing some with the tag. They do not, however, out their own signal (Glover & Bhatt, 2006).

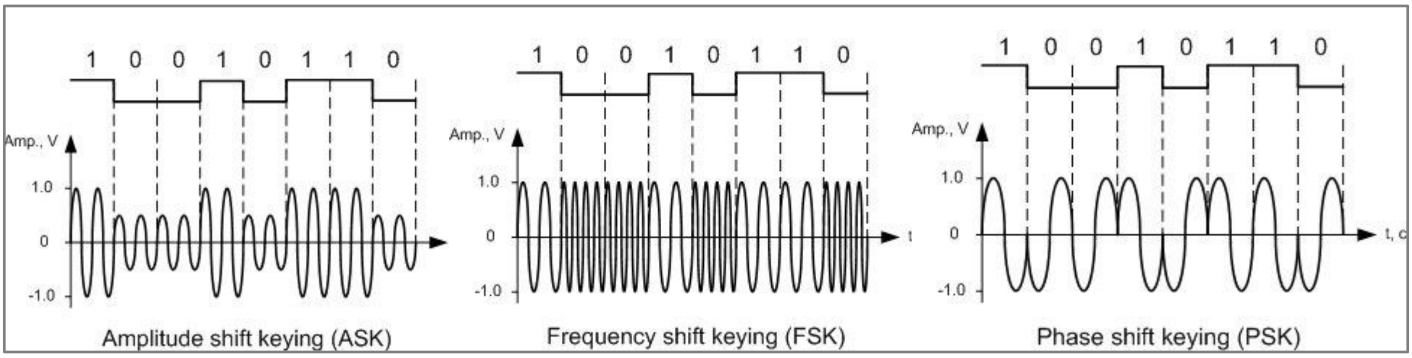


FIGURE 20 THE THREE DIFFERENT TYPES OF KEYING (GLOVER & BHATT, 2006)

The way of sending the data from the tag to the reader is called keying. There are three types of keying, all referring to a way in which the analogue carrier may be modulated to represent ones and zeros in a digital message, they are visualised in Figure 20 (Glover & Bhatt, RFID Essentials, 2006).

An RFID reader can communicate with more than one device at the same time. In order to make sure that the reader does not have difficulty in telling where one tag's response ends and another's begins, anti-collision tags know to wait their turn when they are responding to a reader. The principle of waiting for an answer is also applied when a tag and a reader have made a connection. There are two communication modes in which the devices exchange their information. In full-duplex (FDX) communication, the tag and the reader may talk at the same time. In half-duplex (HDX) communication, the reader and the tag will take turns in speaking (Glover & Bhatt, RFID Essentials, 2006).

AVAILABILITY

In order for RFID, or NFC, to be available, a network has to be set up. It can be used for localisation by subdividing the total area into smaller subareas. At the entrance and exit of such a subarea, a scanner can be placed, to scan every tag that goes in and out of that area. This way the tags that are in a certain area and the order they came in will be known. To implement this system, a network of scanners has to be set up in the subdivisions of the areas. For more precise positioning, this network has to be made denser. To make the system work, scanners have to be installed in the different areas the system needs to cover. Also everything that will be tracked will get a unique tag in order to recognise it coming in or going out of an area.

ACCURACY

The term accuracy is a strange with RFID and NFC, as you will place a scanner somewhere and scan who/what goes into an area and leaves that area. It is only a question of how well the tags are scanned, which is actually part of the integrity parameter. A parameter that is similar to accuracy is the range. Range says something about how far away the tag can be in order to still be scanned.

The frequency on which RFID operates is closely connected to the read range. The spectrum in which RFID operates is split into four parts. These are low frequency (LF), high frequency (HF), ultra-high frequency (UHF) and microwave. The useable frequencies are limited and are known as Industrial Scientific Medical (ISM). In Table 1 the different frequency ranges and ISM frequencies are visible.

TABLE 1 FREQUENCY RANGES (GLOVER & BHATT, RFID ESSENTIALS, 2006)

Name	Frequency range	ISM frequencies
Low frequency (LF)	30-300 KHz	<135 KHz
High frequency (HF)	3-30 MHz	6.78 MHz, 13.56 MHz, 27.125 MHz, 40.680 MHz
Ultra-high frequency (UHF)	300 MHz – 3 GHz	433.920 MHz, 869 MHz, 915 MHz
Microwave	>3 GHz	2.45 GHz, 5.8 GHz, 24.125 GHz

With each frequency, it is possible to estimate a maximum range for passive tags. Low frequency communication has a maximum range of about 50 centimetres, HF RFID of up to 3 meters and UHF of up to 9 meters. The last frequency, microwave, has the largest range, more than 10 meters (Hunt, Puglia, & Puglia, 2007) (Glover & Bhatt, 2006).

The different frequencies also have a couple of other characteristics. Lower frequencies have a slower data rate but do perform better near metals or liquids (Hunt, Puglia, & Puglia, 2007). There are solutions to reduce the effects of metal on the performance. This can be done by adding a spacer of about 5 mm or more, which mitigates the effects a little but not completely. Another solution is to design tags that are tuned to work specifically on metal. By taking into account the expected reflections both the tags as the readers can be optimized so that the metal interference can contribute to the RF field in a constructive way, but these adjustments are quite difficult to make. Another downside is that it will only work well over a fairly narrow frequency range, which will increase the production costs even more (Omni-ID, 2008).

The range of NFC, as mentioned before, is very small. This range is set to be only a couple of centimetres as a security measure.

CONTINUITY

The continuity of the system depends on the type of tags that are used. If the tags are made of a durable material, the continuity of the system should be good, as long as the scanners have power and are connected to a database.

INTEGRITY

For the integrity parameter it is important to make sure that all tags are scanned when passing the scanner. The scan time of RFID and NFC is very low. It takes less than 0.1 ms to scan a tag. Application will determine whether this is enough to scan all passing tags quickly enough (Yaqub & Shaikh, 2012). It is theoretically possible for a reader to communicate at a rate of 1000 tags per second, with an accuracy that exceeds 98%.

2.2.2.4 SUITABILITY OF RFID AND NFC

With respect to range, RFID would be suitable, as the range could be up to several meters. The most suitable frequency to use would be High Frequency, as the range of that frequency is up to 3 meters which is perfect with regards to distance between the shunting tracks. It might even be possible to use Low Frequency RFID, but that depends on where on the wagons the tags will be placed, and integrity has to remain intact, it has to be ensured that every tag is scanned. The range of NFC is way too small to be applicable in this case. The wagons would have to pass the scanner too close, which would make the risk of breaking the NFC reader or part of the train too large.

One disadvantage of RFID and NFC is the fact that measures should be taken in order to make it useable with metal around the tags and readers, seen as there is a lot of metal present at marshalling yards. This will increase the costs of the system. As the tags are very cheap, this might still not make them that expensive.

On the marshalling yard, readers would have to be placed at the point where the train has just passed all switches, because after that it will not be possible for the wagons to change tracks. If another reader is placed before the exiting switches, it is known when the wagons left the shunting track. Each wagon should get a RFID tag with an identification number, and by the order they are scanned at the entrance, it is possible to determine the order the wagons are in.

Applying this method on marshalling yards does mean that a lot of readers are necessary. On the emplacement the Kijfhoek in Rotterdam, about 90 readers would be necessary, if you place one at the beginning of each track and one at the exit (Wikipedia, 2014). The reading rate of RFID is fast enough to scan the wagons while they are driving onto the marshalling yard (Yaqub & Shaikh, 2012).

2.2.2.5 WI-FI & BLUETOOTH

Wireless Local Area Networks (WLAN) or Wi-Fi can nowadays be found in almost every building. You can stay connected to the internet wherever you go, and a lot of free Wi-Fi access points are popping up in public spaces. Even though Wi-Fi originally was not meant as a positioning technology, it can be used for that (Lloret, Tomas, Canovas, & Bellver, 2011). Wi-Fi (or WLAN or IEEE 802.11) is an industrial standard for wireless data transmission. Wi-Fi uses electromagnetic waves to transport information. There are two frequencies used by Wi-Fi, these are 2.4 GHz and 5 GHz. Wi-Fi positioning techniques work based on the parameter RSS, see paragraph 2.1.3.1 (Hennigens, 2012) (Zhang, Li, & Zhang, 2009). Another characteristic of Wi-Fi is that both active and passive positioning is possible as explained in paragraph 2.1.2. Also a hybrid combination of passive and active positioning is possible, where the devices receive a signal from the access point, like in active positioning. Instead of calculating the location on the device itself, the information about the received signals is send to a database, on which the location can then be calculated, just like with passive positioning (Hennigens, 2012).

Bluetooth positioning is quite similar to Wi-Fi positioning. Bluetooth exists since the 1990s, and the first mobile phones with Bluetooth came on the market in 2000. It is a wireless communications technology that is simple and secure. It is present everywhere in a lot of devices, ranging from mobile phones and computers to medical devices and home entertainment products. Bluetooth has a range from up to 30 meters and can transfer data with a rate of 721 kbps (Yadav & Sharma, 2014). Bluetooth operates on a frequency band from 2.4 to 2.485 GHz, so in the same spectrum as Wi-Fi. It is meant to set up a connection between two devices to transfer data between those two devices. The objective of Bluetooth is to replace cabling with wireless transmission of all information. The difference with Wi-Fi is that Bluetooth enables one to send information from device to device by pairing two devices, whereas Wi-Fi enables the device to connect to the World Wide Web. The two techniques do not replace each other, but are complementary to each other (Bluetooth, 2014).

POSITIONING METHODS

The positioning methods as described in paragraph 2.1.4, can be used to describe the different methods for Wi-Fi positioning and Bluetooth positioning. An easy method is the cell identity, but this will yield a localised area and not a real position it will be very inaccurate (Zhang, Li, & Zhang, 2009). Trilateration is also possible, based on RSS values, but a lot of context factors influence the signal strength, and therefore this method is not always reliable. Also the device needs to be seen by at least three access points in order for this method to work (Hennigens, 2012). In the Synthesis Project, part of the master of science Geomatics, some research was done to find out the feasibility of this method, and this research showed that it is very difficult to achieve localisation using Wi-Fi trilateration (Pronk, Theunisse, Jonker, Chen, & Willems, 2014) (Oliveti, Mulder, Abdhilakh Missier, Zervakis, & Wu, 2014).

Triangulation based on the AOA is also possible with Wi-Fi signals. For this, hardware needs to be modified as special antennas are mandatory. This takes effort to do, especially as antennas like this are not present in standard Wi-Fi devices and specially made devices are needed. Triangulation is only possible with a passive receiver. AOA with Bluetooth is also being developed at this moment. Special antenna's and receivers need to be used in order for this to work, but the systems that have been developed should be quite accurate and would be very useful for indoor localisation.

TOA and TDOA cannot be used for localisation, as Wi-Fi and Bluetooth were not developed for this purpose and the precise time measurements necessary for TOA and TDOA are not available.

The last localisation method is based on pattern recognition and is called fingerprinting positioning. Fingerprinting consists of two phases, in the training phase the area in which the positioning has to take place is mapped out and by collecting the signal strength of access points at different points. If a mobile device with an unknown position goes into the area, the signal strengths this device receives will be compared with the recorded signal strengths and a location can be estimated based on the recorded signal strengths (Lloret, Tomas, Canovas, & Bellver, 2011). The accuracy of the different methods depends very much on the situation. Fingerprinting was also researched during the Synthesis Project of the MSc Geomatics in 2014. The conclusion of this study was that fingerprinting is a much more suitable method for Wi-Fi localisation than other available methods. This is mainly because static surroundings (e.g. walls) become much less significant as they are also there during the training phase, therefore less attention has to be paid to the placement of the scanners (Oliveti, Mulder, Abdhilakh Missier, Zervakis, & Wu, 2014).

AVAILABILITY

The availability of Wi-Fi localisation depends on the location of the system and the used method. If Wi-Fi is already available, it might be possible to use that infrastructure to do the positioning. This is most easily achievable if fingerprinting is used, but could also be possible to use the other active modes. For passive Wi-Fi positioning, the Wi-Fi access points need to have the ability to read the signals send by the devices and process these to a database. Not all standard access points have this ability, so in that case either new access points need to be used or special Wi-Fi scanners need to be placed alongside the existing system.

For Bluetooth, special scanners or transmitters need to be installed in order to make localisation possible.

ACCURACY

Accuracy of Wi-Fi and Bluetooth localisation depends on the method that is used. Cell ID is not a very accurate method and greatly depends on the density of the Wi-Fi network. Fingerprinting can be quite accurate, but there are also a lot of context factors there, for example the result can be different when there are a lot of people in the room or when changes are made in the surroundings of the device or the access points. In theory, RSS and AOA would yield the best result. However, due to the many external factors that can influence the signal, this method does not yield the theoretical accuracy. Wi-Fi and Bluetooth localisation could very well be used to provide results like heat maps, because you can quite easily get an idea of the number of people in a certain area, as most people have Wi-Fi and Bluetooth enabled devices and this method scans for this. However, accurate localisation or positioning using Wi-Fi or Bluetooth is still in the development phase.

CONTINUITY

The continuity of Wi-Fi and Bluetooth localisation depends greatly on whether or not the access points are working and whether the devices are actually sending out connection requests or not. A problem encountered while working on the Synthesis Project, was that mobile devices do not send out connection requests as often and regularly as expected and that the frequency of requests differs per brand and per device. This makes it difficult to always know exactly where a device is, as you are dependent on the frequency of the requests send out by the device.

INTEGRITY

The integrity of Wi-Fi localisation is greatly dependent on the number of access points. The larger the number of access points, the higher the probability of correct detection. This is especially the case for the fingerprinting technology. Integrity monitoring can help in increasing the probability because it can recognise a faulty access point location based on the redundancy of the other measurements (Garcia Castano, Svensson, & Ekstrom, 2004).

Not much research has been done towards Bluetooth localisation integrity, but seen as the system is quite similar to Wi-Fi localisation, the integrity will probably be similar as well (Garcia Castano, Svensson, & Ekstrom, 2004).

2.2.2.6 SUITABILITY OF WIFI AND BLUETOOTH LOCALISATION

Wi-Fi and Bluetooth positioning, though interesting, are both not suitable for positioning trains on marshalling yards.

Both methods are greatly dependent on the received signal strength, which is influenced by the surroundings of the devices. There are usually a lot of trains on a marshalling yard, those steel objects will greatly influence the signal strength. Wagons will also arrive, arrange an depart, so the RSS pattern will also change accordingly. There will also be a lot of multipath because all trains will reflect the signal. This results in received signal strengths that will not be representative for the distance between the access point and the receiver that is placed on the wagons.

Furthermore, recent experiences during the Geomatics Synthesis Project 2014 showed that calculating a distance from signal strength depends on many factors and that it is not possible to accurately determine this at all, not even outside without wagons. An alternative to trying to calculate distance from signal strength is to use the method fingerprinting.

However, as the configuration of the freight wagons is constantly changing on marshalling yards, it will be impossible map of the RSSs.

Another reason not to use Wi-Fi and Bluetooth is that the devices that would be on the train need quite a lot of power to send out the signals and this is currently not available in the freight wagons.

The last disadvantage is that this method is most convenient when a WLAN network is already in place, for example in office buildings. On a marshalling yard this would mean that a Wi-Fi network would have to be set up especially for the tracking of the wagons. Because Wi-Fi is actually meant for internet communication and not for positioning, it will be a waste of money to put a network in place that you are not going to use for what it is made for.

ULTRA WIDEBAND (UWB)

Ultra-wideband (UWB) is a broadcasting technique that is being explored more and more lately. It has been in development since the 1960s, but is only being implemented now because the level of technology didn't make implementation possible. The idea of UWB is to send information over a very broad bandwidth, at least 500 MHz (Sahinoglu, Gezici, & Guvenc, 2008). UWB sends packages of data in pulses over the different parts of the bandwidth it is using, instead of sending all information over one wavelength. The pulses that UWB sends can be compared to heartbeats, as can been seen in Figure 21.

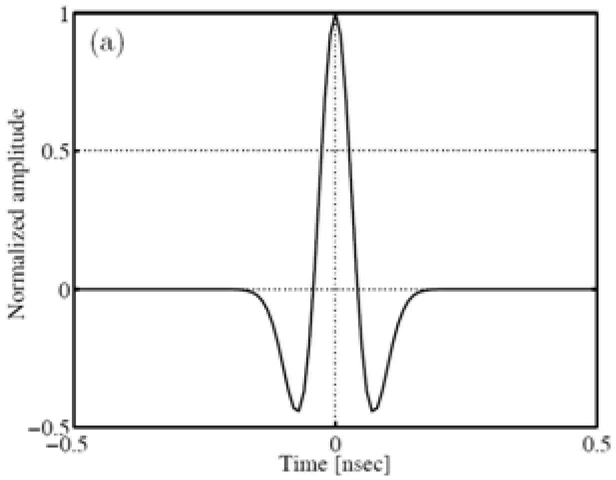


FIGURE 21 FORM OF A UWB PULSE

UWB sends these pulses with different amplitudes and different frequencies within its bandwidth. It can send data very fast, as it is not bound to one narrow wavelength (Aiello & Batra, 2006). Because it sends pulses over different frequencies, errors, for example from multipath, can be more easily detected because they will occur in a different way at different frequencies (Herrmann, No Date).

The fact that it uses a very broad bandwidth also poses challenges. Wireless communication is becoming very important in people's daily lives and it is very important that wireless technologies can exist together and do not interfere too much with each other. Because of the large bandwidth the technique would interfere with a large part of the other wireless techniques, public authorities regulated the bandwidths UWB is allowed to use and the power with which it can use these bandwidths (Sahinoglu, Gezici, & Guvenc, 2008).

UWB positioning can be executed both directly and indirectly. Direct positioning is performed directly from the signals travelling between the nodes, and indirect positioning is a two-step approach in which certain parameters are extracted from the signals first, after which the position can be determined based on those parameters. The complexity of the second way, indirect positioning, is much lower and the performances are usually very close, so it is more practical to use indirect positioning. The parameters that will be used to determine position are RSS, AOA, TOA and TDOA. Triangulation and trilateration are both used to determine the location based on UWB measurements (Sahinoglu, Gezici, & Guvenc, 2008).

AVAILABILITY

To make UWB available, a new system will have to be installed. The system will consist of beacons that are placed strategically in the area in which the positioning has to take place, and the position of these beacons has to be known. Everything that has to be tracked will be equipped with a device that can read and/or transmit UWB signals. The installation of this system requires quite a lot of time. There are also not many people and companies working with UWB, so in that way the availability is also low.

ACCURACY

A very important factor in the accuracy of the system is the positioning technique that is used. RSS can be used in combination with trilateration to determine a position, but RSS measurements will not provide very accurate range estimates, because of inaccuracies in both the RSS measurements and the quantification of distance versus signal strength. The standard deviation of this error cannot be made smaller than one meter for distances larger than 6 meters, so the accuracy of this method lies in the meter level (Sahinoglu, Gezici, & Guvenc, 2008).

Another parameter that can be used is the AOA which can be used to determine position using triangulation. The accuracy of this method is much better than that of the RSS method, it lies in the decimetre level.

TOA can be also used as a parameter. As we already saw in paragraph 2.1.3.3, the increase in the effective signal bandwidth can cause the accuracy of TOA measurements to be improved, and UWB has a very large bandwidth, which allows highly accurate distance estimations (Sahinoglu, Gezici, & Guvenc, 2008). The theoretical accuracy of this method with a reasonable SNR value can be around a couple of centimetres and higher bandwidths can lead to even better distance estimation. TDOA can also be used to determine position with UWB, in case the clocks of the receiver and the transmitters are not synchronic. This method will yield a comparable accuracy.

CONTINUITY

The continuity of UWB is comparable to that of Wi-Fi positioning and Bluetooth positioning. The beacons need power and a connection to the database. The tags might also need power and could require a connection to the database, dependent on whether you will use passive or active positioning. These power supplies and connections have to work in order to make sure continuity is guaranteed.

INTEGRITY

A study conducted at the Karlsruher Institut für Technologie showed that the integrity of UWB is comparable to that of Wi-Fi and Bluetooth localisation. If not enough scanners are visible, the estimated location will become less good and integrity drops. This research group combined UWB with Inertial Sensor Based Systems (INS) and developed an integrity algorithm that greatly improved the integrity of the estimated localisation.

2.2.2.8 SUITABILITY OF ULTRA WIDEBAND POSITIONING

Ultra wideband positioning, even though comparable to Wi-Fi and Bluetooth localisation, is a lot more suitable for marshalling yards. This is because of its capabilities with triangulation, the accuracy of ultra wideband is a lot better, as it is much easier to detect errors and filter them out when pulses are send in this way than if it is just a regular EM wave.

To implement the system a new infrastructure needs to be implemented on the marshalling yard. It is still uncertain how ultra wideband behaves when surrounded by the amount of steel that is present on marshalling yard. As it has a lot of special capabilities with respect to several errors, it might be able to handle it better than other systems, but this has not yet been tested so this could be a limiting factor.

The mobile devices in the train will also need power in order for the localisation to be possible. It is unclear how much power is necessary, but this is also a factor that could be limiting.

A disadvantage is that the method is still in the development stage. There is still a lot that is not certain and even though the method might seem promising, it would be a risk to use it in this project, as there is not time to change to a different system later.

2.3 CHOICE OF LOCALISATION METHOD

TABLE 2 COMPARISON OF TECHNIQUES BASED ON PERFORMANCE PARAMETERS

Method	Availability	Accuracy	Continuity	Integrity
GPS point positioning: pseudo range	Easily available with simple GPS sensor	5-20m	Quite continuous, not much off-time	After first fix, the method is trustable
GPS point positioning: carrier phase	Difficult because of special scanners that are needed	Sub-cm	Same	Same
DGPS: pseudo range	Quite easily available with GPS sensor and computer	5cm-10m	Same	Same
DGPS: carrier phase	Difficult because need for special scanners	Sub-cm	Same	Same
RFID	A lot of scanners will be needed to be able to locate all wagons	Precise	RFID should be fast enough to scan the wagons while driving	No real indication is given on whether the scanners will always work
NFC	Not possible to place scanners so close to the wagons	Precise	Fast enough, so continuity should not be a problem	Same as with RFID
Wi-Fi	Not available on marshalling yards, new	Decimetre	A lot of influence by structures on the marshalling	A lot of influence by structures makes it less

	system		yard	trustable
Bluetooth	Not available, new system, range small	Decimetre	Same	Same
Ultra wideband	Not available yet, new system	Centimetre	Good, different bandwidths is less influence of surroundings	Not really known yet, has to be tested

In Table 2 a comparison has been made of all techniques. This comparison is made based on the performance parameters. Only a couple of methods for positioning trains on marshalling yards remain, these are RFID, GPS/GNSS and Ultra Wideband.

There are couple of other performance parameters that can be used to help making this choice, these are yield, consistency, overhead, power consumption, latency and roll out and operating costs. Two of these are very useful in this comparison, these are power consumption and roll-out and operating costs. Also the parameter overhead is interesting to look at.

Power consumption is about the amount of battery resources that the devices that have to be localised will use in order to be localised. For both GPS and UWB, the power consumption is in general high, whereas for RFID, the power consumption is very low, especially when using passive tags. In their current set up the GPS devices of Undagrid are very power efficient, which will be elaborated later (Martin, Liu, Covington, Pesti, & Weber, 2010).

The roll-out and operating costs are the costs to install the infrastructure and databases. There are also costs for the databases to be maintained, and costs to control the units. UWB is, according to Martin et al. (2010) still very expensive. However, the technique is still under development, and will probably become a lot cheaper in the coming years as it will be applied more. GPS/GNSS receivers are becoming increasingly cheaper. The costs for GPS/GNSS are also dependent on the method that is used to determine the location. The costs for RFID are very low, the tags cost almost nothing and the readers are also not very expensive. In this case a lot of readers would also be required because a reader should be placed at the end and the beginning of each shunting track (Martin, Liu, Covington, Pesti, & Weber, 2010).

After extended comparison and deliberation with my supervisors from both the TU Delft and CGI, the decision has been made to use GPS as a method of localisation. GPS is a method with a lot of possibilities and has been used and researched a lot. This method is also very suitable in this case, as it connects best to the other part of the system that is being developed at this moment. Because GPS/GNSS has been chosen as a method, the system of Undagrid can be used for testing, which is also an advantage. This system and the company are already known at CGI, which is an advantage in the development of the rest of the system as a lot is known about Undagrid and how the devices work.

2.3.1 OTHER RESEARCH ON GPS AND TRAIN LOCALISATION

To get more insight in the academic field on this subject, research towards lane identification with help of GPS has been investigated. This research showed that, with help of real time precise predicted orbits, clock offsets and ionosphere maps, it was possible to get lane identification with errors smaller than 1.1 meters for 95% of the time. The Root Mean Square Error was about 50 centimetres. Of course this is a different situation than the one in this research, as this extra information will not be used, but still it showed that it is possible to locate something quite precisely when using low-cost GPS equipment (de Bakker, Knoop, van Arem, & Tiberius, 2013).

Another connected research field is the research towards the behaviour of non-moving GPS devices. Most research is done towards navigating and finding moving devices, which means that most algorithms focus on improving positioning of a moving device. Positioning of a stationary device often yields less accurate results because no improvement is added to the position in the algorithms. More research is at this moment being executed towards this field of GPS positioning.

This project actually is a combination of the two that are mentioned above, lane identification is also in issue determining on which track the wagons is located. As wagons are usually not moving when they are on a marshalling yard, research towards non-moving GPS is also important. This research will be extended during the project, as research will be done towards the influence of using previous GPS/GNSS measurements to improve the resulting location.

3 SOFTWARE

3.1 MARSHALLING YARD CHARACTERISTICS

Localisation in this project is subject to several typical characteristics that are determined by the case: a freight wagon on a marshalling yard. Three of these characteristics have been very important during this project, and will be discussed here.

3.1.1 STATIONARY VEHICLES

Wagons on a marshalling yard are stationary most of the time. The purpose of a marshalling yard is twofold. It is often used as a place for the train and the driver to stay a while. This can be because the driver needs rest, because more load is expected or because the train has to wait before it can leave again. During this time the train will stay on the same spot and will not move. Another purpose is to rearrange the train or combine several trains to form a new train. In this case some driving will occur, but again, the wagons will also be stationary for a large part of the time.

In this project, the stationary character of the wagons creates an opportunity to investigate the possibilities of stationary GNSS. It is possible to use the history, in other words the previous measurements, in order to improve the accuracy and precision of the location that is calculated. The amount of improvement of this measure will be investigated using this project. Techniques to do this are based on statistical analysis of a large number of values. Even though in this system the goal is not to analyse the values, but to calculate an improved value of the location, techniques of statistical analysis of datasets can still be used. There are many ways to calculate a single value out of multiple measurements, but a lot of them are very elaborate and require the results to be plotted to a normal distribution (Rousseuw & Croux, 2012). While these are very good methods when doing accurate statistical analysis of the results, they are not very useful to calculate a better result for our GPS measurements in real time, as they would yield abundant data and would slow the process of getting a single location down significantly. There are two methods that can be implemented with less influence on the calculation time of the system, but which can still improve the resulting location significantly, these are the average and the median. Both methods were implemented during this project, with the possibility of changing from one to the other to see what influence they would have on the calculated location. This part of the project is implemented to see whether real time application of these methods would improve the result of the system and make it more probable that the wagon is located on the correct track. Both methods will be explained below (Dixon, 1953).

The first statistical method, with which the results could be improved, consists of taking the mean, or average, of the previous results and the current measurement. To take the mean, all longitude (or y) values will be added together and divided by the number of measurements to get the mean longitude (or y) value. The same goes for the latitude (or x) values. The underlying principle of applying this statistical method is that when you calculate the mean of multiple measurements, as many measurements as possible, the resulting value will get closer and closer to the real value (Rousseuw & Croux, 2012). For GPS measurements in the open field, this will result in the average measurement which is located in the middle of the original cloud of raw data.

An issue with the mean of measurements is that measurements of GPS locations can have large outliers, results that are calculated a hundred meters from where the device actually is. This can be due to all sorts of error sources, as was explained previously, but can have a pretty large influence on the calculation of the average. This problem can partly be solved by removing the outliers from the measurements. However, measurements that are on the edge of the scatter cloud are still quite far away and will still have a rather large impact on the result. This issue can be solved by calculating the median of the coordinates instead of the mean.

To calculate the median, the values of the latitude and longitude will be put in order from smallest to largest, the median is the middle value. Because the values will first be ordered from smallest to largest, the outliers will automatically be either at the top of the list or at the bottom. The only influence on the result they will have, is on the amount of measurements, and thus on where the middle will be exactly. But as the measurements more towards the middle will lie closer to each other, this will much less significantly influence the end result of the calculations (Rousseuw & Croux, 2012). After calculating the median of the different measurements, the result will lie in the densest middle part of the raw data.

In an old, but still relevant, paper on statistical analysis (Dixon, 1953), research is described about when it would be more suitable to use the average versus the median to come to an appropriate solution. The conclusion of this research is that the mean calculation is best used in cases of small sample sizes with smaller biases. When larger sample sizes are used, the median is the better method. As the sample size of this project will very quickly become large, the median was the more desirable of the two methods before the tests were done, also because this one is less affected by the large outliers (Dixon, 1953). However, the tests will show whether this is actually the case, which is why both methods were implemented in order to compare the two.

This is a very interesting subject, as not much research has been done about improving the result of stationary GPS sensor based on previously gathered data about the location. In this project, one of the results will be an answer to the question whether or not it is favourable to use this previously gathered data.

3.1.2 RAIL NETWORK

The most interesting and most useable characteristic of a marshalling yard is the fact that the wagons have to be on a rail. There is a very small chance that a train might have been derailed, but this is outside the scope of this project, as it would ask for a very different system to find derailed wagons than to find wagons on a rail. Also the chance of derailment on a marshalling yard is quite small, as the speeds are very low.

As a train's location will always be somewhere on a certain track, this gives an extra dimension to the project. The location of a coach cannot be anywhere on the planet, it has to be somewhere on the rail network of the marshalling yard. You could even see this rail network as an extra reference system that can be given to the found location. It also gives a twist to the project, as the result no longer has to generate a specific position, but instead can give a location on a certain track.

The rail network does not only cause an extra reference system to be present in this project, it also gives another input to the system. A train cannot move from one track to the other without driving all the way around the track. If a train remains stationary, it will also remain on the same track and cannot suddenly appear on the track next to it.

The characteristics of the rail network on a marshalling yard are a very interesting part of this case. This is because the characteristics of the rail network can help improve the accuracy of the result. A technique called map matching, which is also used a lot in car navigation, can be used for this purpose. Map matching calculations use the position or location as calculated in the previous steps and add data from a road or rail network to this to improve the positioning and localisation output. The purpose is to identify the right network segment on which the device is lies. The resulting accuracy depends both on the quality of the point location and on the quality of the network map (Quddus, Ochieng, & Noland, 2007).

Map matching can happen at different levels, there is point-to-point matching, where each position is matched to the closest point on the network by computing the minimum distances to the different points (Bernstein & Kornhauser, 1996).This is a fast and easy to implement method, but its accuracy is very sensitive to the way the network is designed. If more points are added to the arcs, the resulting position will be more accurate than, for example, when an arc only has beginning and end nodes (Quddus, Ochieng, & Noland, 2007).

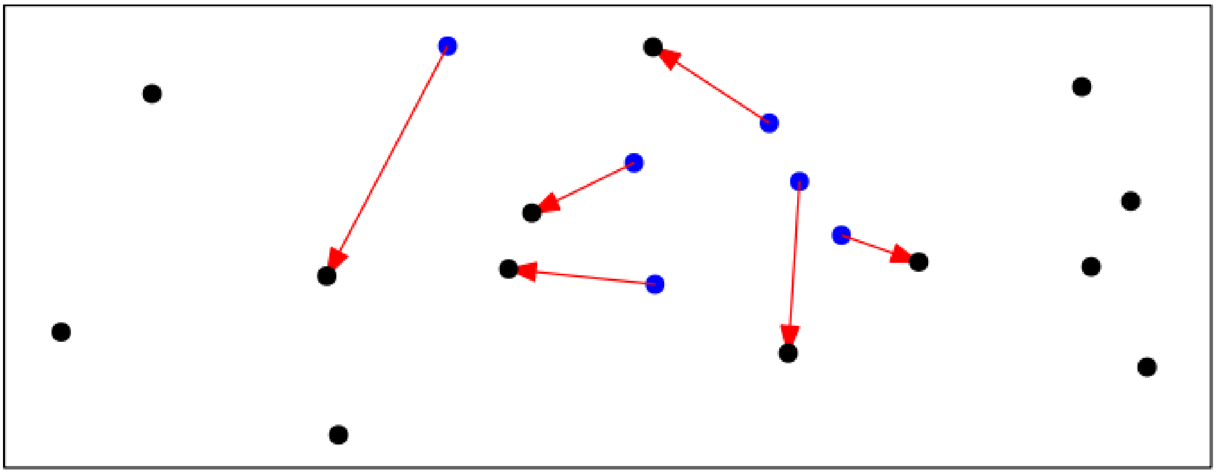


FIGURE 22 POINT-TO-POINT MATCHING (ROTE, 2010)

A second map matching technique is curve-to-curve matching, which matches two curves to each other. This could be used when the input is a line that has been walked and that line is matched to lines in the network. In this case the position is given as a point and this point will have to be matched, so this method is not relevant here. In this project, point-to-curve matching will be executed. Point-to-curve matching finds the closest curve in the network and matches the point to that curve. Each curve comprises of linear line. The line segment with the smallest distance from the point is selected as the one on which the vehicle is present, and the point will be moved to that line as is visualised in Figure 23 (Quddus, Ochieng, & Noland, 2007). This method will yield a better result than point-to-point matching, especially in cases where the network curve only consists of a couple of points.

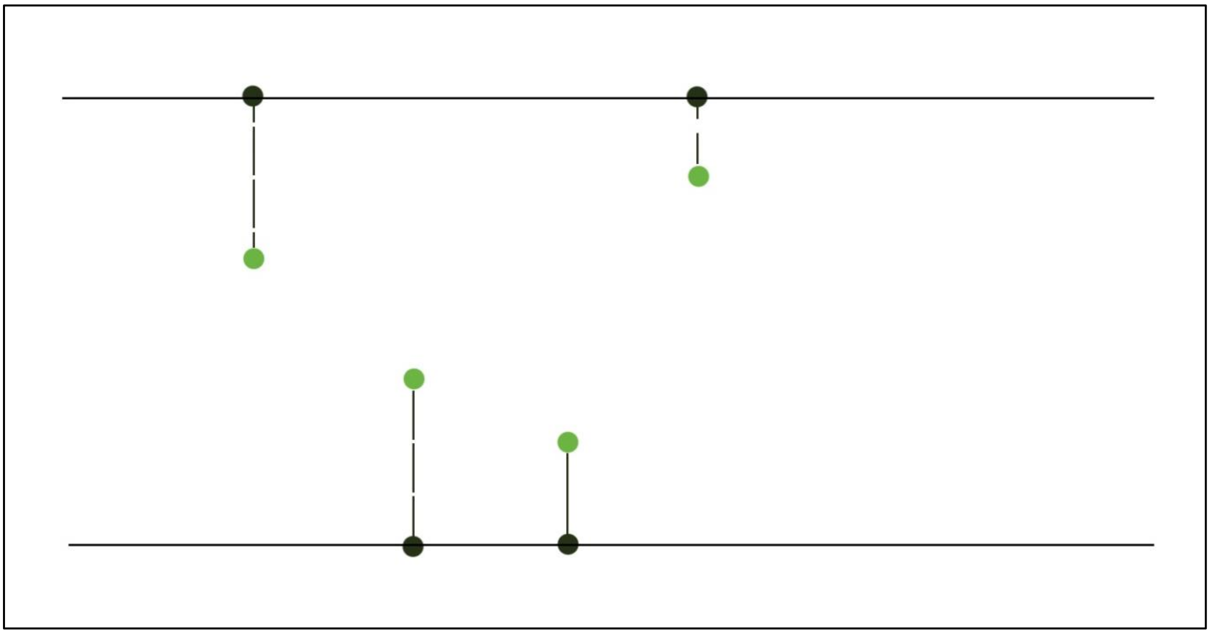


FIGURE 23 POINT-TO-CURVE MATCHING (OWN IMAGE)

So, by using the point-to-curve matching to match the point to the rail network, the inaccuracy from the sensors will be reduced and a specific rail number will be given to the result. Important to notice is that the point position of the device is actually not that important, but the localisation of the train on a certain part of a certain track is very important, and that value is found in this step. By snapping the wagon location to the network, the location is better defined for the purpose, and it will be easier to find the wagon.

Apart from just knowing that the train is on a network, it is also known that the train has not moved, this provides us with ‘historical’ information about the tracks the wagons were matched to earlier. By using this information, it might be possible to more accurately determine the track that the wagon is actually on. This could be done by taking all previous matched tracks into account and calculating the percentage that each of those was matched. The track with the highest percentage could be chosen as the correct track and the point could be snapped to that particular track. The percentage will be given as an attribute for analysis purposes. Because it would not be favourable if a correct the measurement was originally matched to the correct track, but this changed to an incorrect track because that one has a higher percentage due to earlier errors, a certain threshold percentage could be set. This threshold percentage would have to be found by trial and error. Only if the percentage for a certain track is higher than a certain percentage, will the point be snapped to this track instead of the original one. Tests and analysis will show whether something with the percentage has to be done or whether it is better to just use the track the device is currently matched to, as the previous results are also already incorporated in the previous step.

The network that will be used to perform the map matching is part of the Basisbeheerkaart (BBK), a network of all rail elements of ProRail in the Netherlands. This map contains a lot of information, of which a large part will not be relevant for this project. The relevant part is the so called spoorassen (track axis), for which data is also available on the marshalling yard. This part of the network will be filtered out, after which the different marshalling yards will be filtered out of the remaining network. The reason to take only part of the BBK is that if the whole network was taken, this would make the system very slow, as it would have to load a lot of redundant information. A polygon will also be created around the different marshalling yards of the Netherlands, in order to detect and remove outliers from the data. Outliers are the measurements that are outside this polygon that represent the marshalling yard.

FIGURE 24 THE WHOLE RAIL NETWORK OF THE NETHERLANDS (PRORAIL, UNKNOWN)

In Figure 24, the whole network of the Netherlands can be seen as it is present in the BBK, and in Figure 25 is an image of just the marshalling yard de Kijfhoek in Moerdijk. The BBK is originally created using photogrammetry and is updated at least once every three year, but this will be improved when techniques make it possible to process changes directly. The average relative precision of the BBK is 0.56m (ProRail, Unknown).

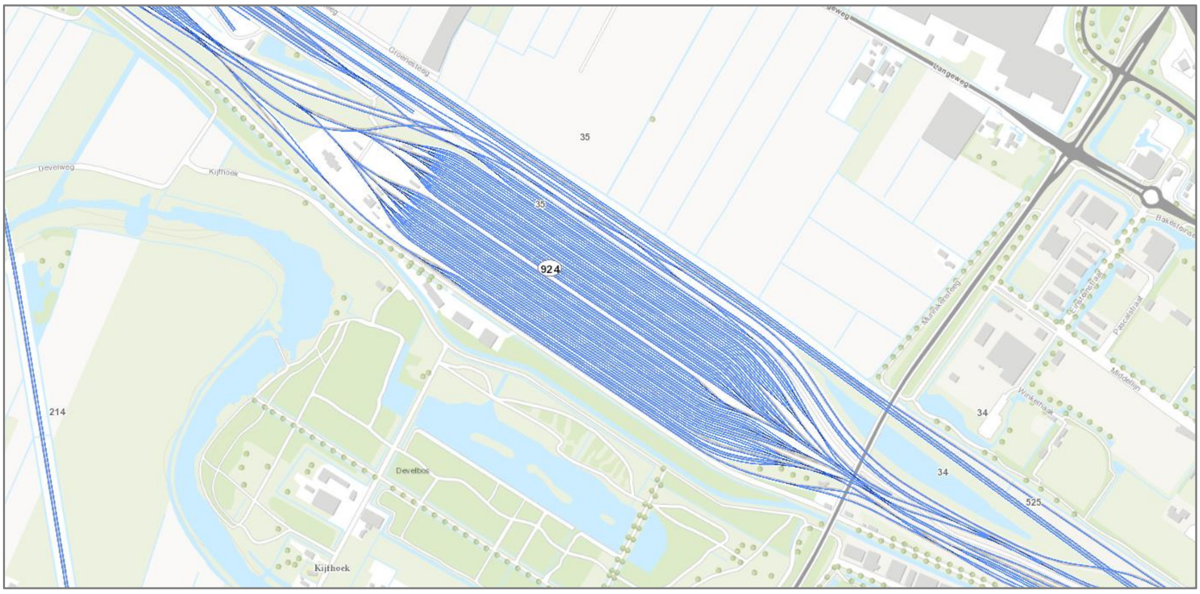


FIGURE 25 MARSHALLING YARD DE KIJFHOEK, MOERDIJK (PRORAIL, UNKNOWN)

The tracks on a marshalling yard are a lot of straight lines, that lie about 5-6 meters apart. The fact that there are this many lines that are only a couple of meters apart makes localisation on marshalling yards harder than when the train is not on a marshalling yard. While the train is not on a marshalling yard, there are usually two train tracks present, one for each driving direction. This makes it quite easy to determine which track the train is on, especially while the train is driving. To determine the track, the driving direction can be calculated after which the correct track can be chosen as there will only be one possible track left. Map matching is then performed to this track. On a marshalling yard this is a lot more problematic, as many tracks are available. An advantage of the marshalling yard, as mentioned before, is that previous results can be taken into account to improve the result and make it possible to determine the track after all, which is not possible when the train is moving.

3.1.3 ORDER OF THE WAGONS

The last characteristic of localisation on marshalling yards is the fixed order of the wagons. As wagons on a marshalling yard are usually not forming a train, it has to be made clear that when the order of the wagons is mentioned, this is about the wagons that are located on the same track. This order cannot change unless several manoeuvres have taken place, for which several wagons will have to be driven around the marshalling yard. Therefore it is safe to assume that the order of the wagons will remain the same as long as the train has not moved. This fact can be used during the development phase as a way to test the system. The distance between two devices cannot change because the order does not change, so the accuracy of the system can be measured based the deviation of the measured distances versus the real distances. After the development phase, this information can be used to determine the order of the wagons on the marshalling yard based on the measurements that are made.

To calculate the order it is necessary to get all wagons that were snapped to the same track as the current wagon, calculate the distance and bearing to each of them and by comparing these, it would be possible to know the order of the wagons.

However, this calculator heavily depends on the fact that the correct track has been given to each of the wagons, because if all wagons on a marshalling yard have a device like this, they can be in the same mesh network despite being on different tracks. This implies that it is necessary to first see the accuracy of the map matching process.

3.1.4 COMPARABLE PROJECTS

Even though there have not been many projects which are truly comparable, there are sectors in industry where localisation has been done and researches can be used to gain more knowledge about the drawbacks and possibilities. For this part of the academic field, two sections of industry were researched, the shipping industry and public transport industry.

In the shipping industry the focus is on security and the centralisation of information. At this moment, there is a very large amount of different documents about the location and load of the different containers, and this could be made more efficient. Technology is used more and more. The technologies that are used most at this moment are RFID and GNSS, and combinations of those two are also researched and implemented. Similarly to what was concluded after the literature study of chapter 2, all techniques have their advantages and disadvantages. RFID is more precise but requires more manual work in the case of shipping industry and will not yield automatically a position like GNSS, but GNSS is less accurate.

Other methods are investigated, but combinations are a very logical step as well (Berger, et al., 2012).

Research towards tracking in public transport focusses on tracking as a way to improve the user experience of public transport, directly (e.g. apps) and indirectly (by making transportation more efficient). Different techniques are used in this field of industry, but the most used techniques are based on GNSS/GPS in some way. There is also research towards also implementing other methods. Kane, Verma & Jain (2008) researched a method that works when tracks are more or less fixed, for example with trains, trams and buses. A distance meter is used to measure the distance the vehicle has travelled and can communicate this via gateways that are positioned on tactical places on the route. This way the train, bus or tram can be roughly localised.

This project investigates a new, but related field of tracking. Implementing tracking on marshalling yards has, according to the literature found about it, not yet been researched and therefore will supplement the different researches that are already been done in this field.

3.2 GEOEVENT EXTENSION FOR ARCGIS FOR SERVER

In order to process all measurements real time, the GeoEvent Extension for ArcGIS for Server is used. This is a tool developed by Esri that enables easy configuration of real time data. The GeoEvent Extension can connect to many types of streaming data feeds and by this, GIS applications can be transformed into decision applications. Connectors for common data streams are available, which include in-vehicle GPS devices, mobile devices and social media providers. Additional connectors can also be found online.

After a connection to a certain data stream is made, so-called events will be read by the connector, and these events can serve as an input to the next step, which is to process the data. In the processing phase, important events, locations and thresholds for specific operations can be detected and focussed on without interruption. Multiple streams of data can flow in continuously through filters and processing steps that can be defined. There are a couple of standard processors, for example GeoFences that can be used to see if a certain vehicle enters a dangerous zone.

The processed event can be delivered to an output in real time. It is possible to automatically and simultaneously send alerts to key personnel, to update the map, to append the databases and to interact with other enterprise systems, for example when locations change or specified criteria are met. Alerts can be send through multiple channels, such as emails, texts and instant messages (Esri, 2014).

So far, the GeoEvent Extension may seem limited in its possibilities. However, using the SDK that was developed with the tool, custom connectors and processors can be made. Connectors can be both input and output. An input is responsible for producing a series of GeoEvent objects and an output is responsible for consuming them. The input receives raw data from an outside source and will translate this into events. A connector, both input as output, consists of two components: a transport and an adapter (Esri, 2014).

The transport is responsible for generating raw events in their native format. The transport enables the flow of inbound events, and is also the part of the system that is stopped and started when the user stops or starts the input. The adapter translates the raw events into GeoEvent objects, these do not have start/stop methods as they are event driven (Esri, 2014).

Writing custom processors is also possible. Processors work similarly to transports, they are event driven, which means they only do something when a GeoEvent is provided by an input connector. There are standard processors, but for a specific project it could be necessary to build custom processors. Custom processors have to be build according to the rules as defined by the SDK and have to be built in Java (Esri, 2014).

A last important part of the GeoEvent Extension is the GeoEvent Definition. Every event that is processed has a reference to a certain GeoEvent Definition. The purpose of this definition is to provide metadata for the GeoEvent. The definition includes a list of the fields that can potentially be in the GeoEvent, this could be names, types etc. GeoEvent Definition can be generated automatically, but can also be created by the user or an automatically generated definition can be edited by the user. The most important part of a GeoEvent Definition is the Globally Unique Identifier (GUID). This is one value that must be different from any other GeoEvent Definition and must never change (Esri, 2014).

For this project, a custom connector and some custom processors are written and implemented. All components, being input, processors and output, will be thoroughly discussed in the next chapter. An overview of the GeoEvent service can be found in Figure 26.

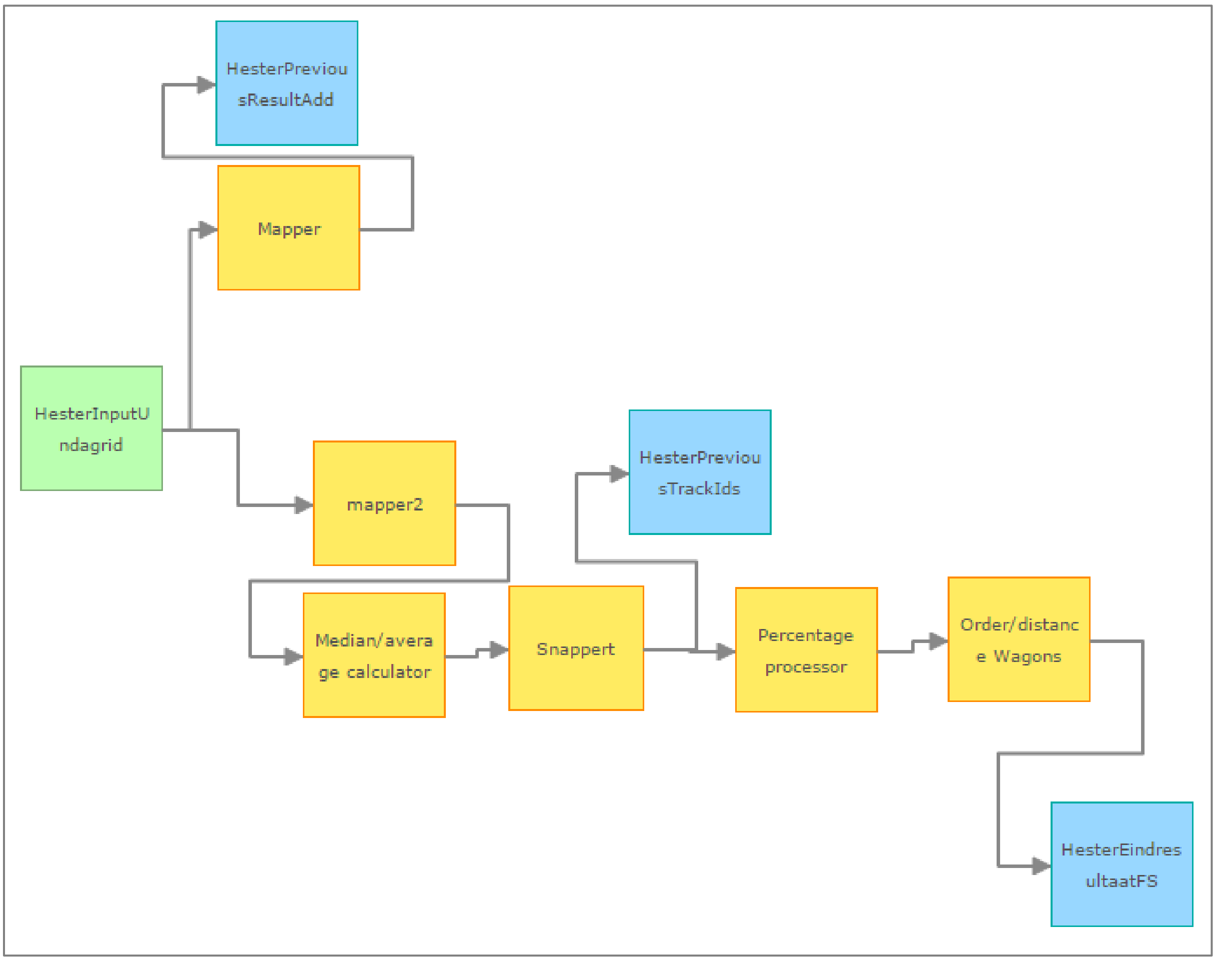


FIGURE 26 OVERVIEW OF THE GEOEVENT SERVICE

The green box represents the input from the Undagrid server, this box contains a custom connector that receives and reads the GeoEvents. This box is followed by two field mappers. Field mappers are standard processors that translate the attributes of one GeoEvent to another, field mappers can be used if one stage of processing does not align with the schema of the next stage of the process.

After the top mapper, an output called Previous Results can be found, this output creates a database to store the previous measurements that are used in the median or average calculator. The median/average processor, which can be seen in the bottom branch, uses the previous measurements to define the average or median of all locations. The snap or map matching processor snaps the result to a certain track in the network, which will also be saved in a database, similar to the previous results database. The next processor is the percentage processor. This processor takes the tracks the device has been snapped to and calculates how often the device has been snapped to each track, the percentage of the track the result is currently matched to is given as an output for analysis purposes. This percentage could show something about the accuracy or performance of the system, the higher the percentage the more reliable the results are. The last processor calculates the distance and bearing between the different devices on a certain track. This gives the order of the wagons as an output. The final blue box is the output, the location and all attributes form the output to a Feature Service in ArcGIS for server. This can then be used by ArcGIS online and the Operations Dashboard to show the result.

Another output that has been added often in the development phase, but is not necessary once the system is finished, is the output to a JSON file. In order to see what happens in every separate step and analyse this, the intermediate steps were saved to a JSON file in the development phase of the system. This enabled better debugging and analysing of the system and its results.

4 IMPLEMENTATION

4.1 HARDWARE DESCRIPTION

During this project, the GPS localisation system from a company called Undagrid was used. Undagrid is a small start-up company that provides localisation systems to different parties. They especially aim to develop systems that use little power, in order to make them useable on places where grid power is not available.

The system that is used during this project works with a number of nodes with a GPS receiver in each node, this receiver is the uBlox MAX7Q. This receiver does not only receive GPS information, but also Glonass information. It is technically a GNSS receiver and also receives corrections from the European correction system EGNOS. Each box calculates its own location and the best result is sent. Best in this case is the one with the smallest horizontal dilution of precision, see paragraph 2.2.2.1. It calculates the position during a small amount of time and sends the result to a gateway via a mesh network, which will be explained later. When the GPS node is not calculating or sending data, it is in low power mode in order to save energy consumption.

The system does not only save power by turning itself on low power mode when not sending or calculating data, it has another trick to use as little power as possible. In the nodes, there is an accelerometer present. This is the LIS3DH chip, which is chosen because of price and energy consumption. When this accelerometer detects movement, the GPS module in the node will be turned off, it will not calculate its position. When the accelerometer stops detecting movement, it will calculate its location. The standard settings for the system is to calculate the location only once, but for this project a calculation should occur about every minute when the wagon is standing still in order to gather previous measurements. Once the location is calculated, the devices will go in power saving mode.

Unfortunately, this version of the devices could not be used for the tests, as the company did not have the time to change the settings so that instead of calculating the location only once when it was standing still, it would calculate it every minute or so. Because of this, Undagrid offered us their old demo set, which calculates the location every minute anyway. This did not change a lot, apart from calculating the location more often, it basically works in the same way.

The devices still communicate to each other via a mesh network, the data is still send to the server in the same format. However, there are two aspects that did change because of the switch in demo set. First of all another GPS receiver was used in these nodes, being the Ublox UC530M, which is still a low cost GPS receiver, and is much comparable with the other receiver, it is just an older version of it.

The largest difference between the two systems is that no accelerometer was present in the demo set that was used in the end. This means that no movement was detected and while this was not a problem in testing the performance of the system, it did mean that the accelerometer data could not be used to determine the timestamp at which the wagon would have come to a stop. This timestamp is important when the system is implemented on trains as they are moving and stopping and only the previous location data since the last time it stopped is needed. However, the system could still be designed and developed for this project, but it has been kept in mind that if the system were to be developed in reality, this condition would have to be added to it.

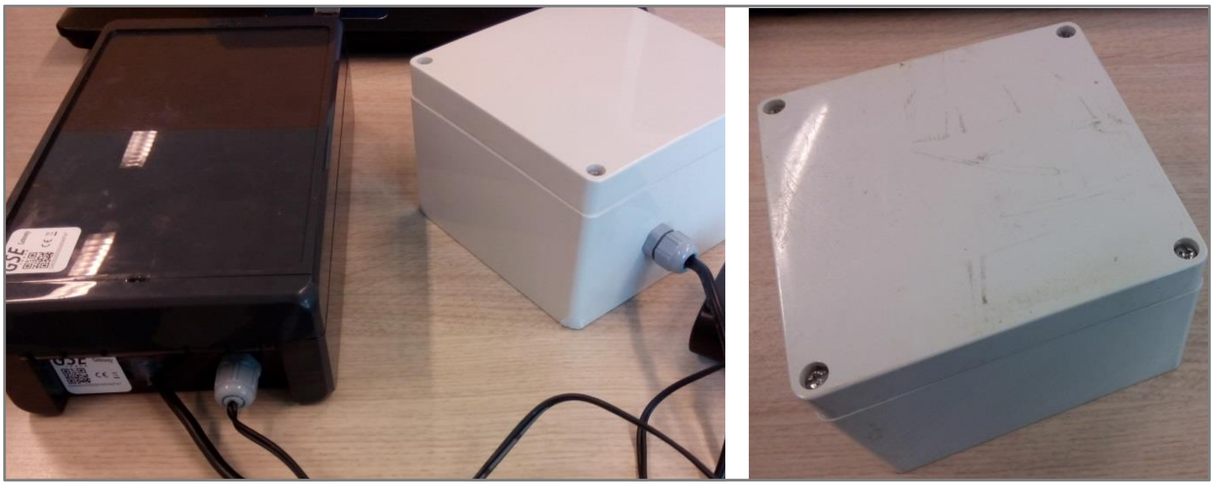


FIGURE 27 THE GATEWAY (LEFT) AND A SINGLE UNDAGRID GPS NODE

4.1.1 WIRELESS MESH NETWORKS

In order to transport the positioning data from the different devices to the server, a mesh network is used. Wireless Mesh Networks (WMN) are network configurations that communicate wirelessly via different hops, see Figure 28. The way a WMN works is very well comparable to the way wired networks communicate. Typical frequencies on which WMNs operate are 2.4 GHz and 5 GHz (Gerkis, 2006).

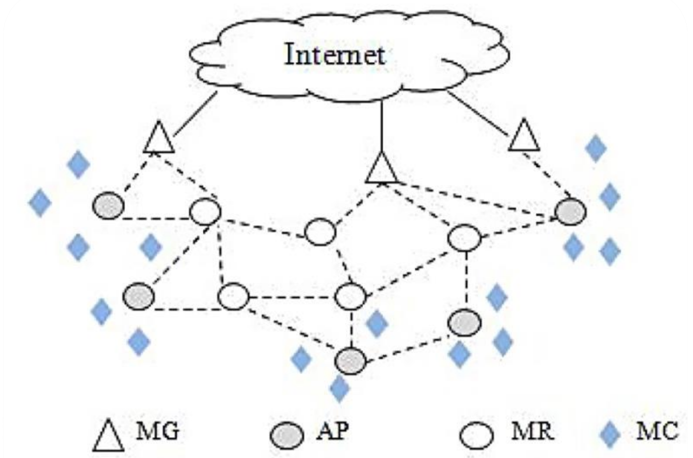


FIGURE 28 PRINCIPLE OF WIRELESS MESH NETWORKS (BENYAMINA, HAFID, & GENDREAU, 2012)

At this moment, a large part of the research towards WMNs is done towards ways to reduce the power consumption. A research of Nagabhushan et al. (2013) categorized the power saving approaches into five categories: routing types, protocol approach, energy conversing metrics, topology control scheme/design and path selection. An extensive description of these power saving approaches can be found in the research report, they will be described only shortly in this section. Routing type approaches depend on effective and efficient decision making. Protocols will also affect energy consumption if the service is reliable. Energy conversing metrics work by selecting routes based on the battery capacity or by avoiding hops to be used too much, which will cause bigger battery drain. Topology control scheme works by arranging the nodes in such a way that the energy consumption stays low and finally path selection which will select paths in such a way that nodes with less battery capacity are used less (Nagabhushan, Shiva Prakash, & Kinkin, 2013).

This project extends the research towards mesh networks with low power consumptions by testing a system that has implemented a power saving mesh network. The mesh network of the Undagrid devices works on 2.4 GHz with a proprietary protocol, which means that it communicates via a protocol that was especially designed for this system and no communication with devices outside this system is possible. The reason no standard protocol is used, is that the devices use a special kind of power saving mode, which falls in the protocol approach. The communication of the devices among each other works by a heartbeat rhythm of communication, which makes it possible for the devices to sleep for the largest amount of time.

The cycle of the heartbeat on which the mesh network communicates in this case has a pre-set cycle of a couple of milliseconds. If a node is on its own, it will start this cycle autonomously on a random point in time. There are two options for the next node, it can either start its own cycle, probably with an offset compared to the first one, or it can only listen to see if other nodes are available. The nodes do automatically listen periodically, in order to find each other when they are out of sync. Not only do the nodes randomly listen every so often, they also send out a small data message apart from their regular cycle, which contains enough information to find be able to synchronise. If a device hears another device, the network with the lowest network time will join the one with the higher network time.

Though a very interesting topic, the battery usage of this system has not been researched extensively during this project, as this lies outside the scope of this graduation project. The system has been used and no issues with the batteries have been noticed. The gateway itself was powered by a portable power bank during the tests, and did not consume much power. After two hours of testing, more than 75 % of the 4000 mAh was still available. So even though it lies outside the scope and other research would have to be done towards the efficiency of this system, something can be said about the power consumption, which is that it is probably quite low.

4.2 SOFTWARE

4.2.1 INPUT FROM UNDAGRID SERVER

As described in paragraph 4.1, the Undagrid devices send their location to a gateway, which is connected to the internet and sends the data to the Undagrid server. The data that is gathered can be extracted via a REST endpoint. This extraction was supposed to be possible by a push mechanism as well as by a pull mechanism, but the push mechanism was not working during this project. Fortunately, it is possible to build a connector for the GeoEvent processor that pulls data from an API endpoint. The transport that was developed requests the Undagrid server every minute, as the devices calculate their location every 60 seconds. This way, all data will be imported into the GeoEvent extension, which is necessary for the system to work correctly. The source code for this input transport was based on existing code with just a few parameters changed and can be found in appendix B. The adapter that is used is a standard generic JSON adapter. This can be used as the data the Undagrid server delivers is in the form of a JSON.

After the connector, two field mappers are used to translate the attributes, as the output of the Undagrid server is in the form of a JSON file with several subfields. Subfields cannot be processed in custom processors in the GeoEvent SDK, for this reason field mappers are used to flatten the hierarchical structure.

4.2.2 PROCESSORS DESCRIPTION

Four custom processors have been developed in order to process all data correctly for this project. These four processors will be discussed in this paragraph, except for the field mappers, which have been discussed in the previous paragraph.

4.2.2.1 GEOFENCE

The first processor that is executed in the GeoEvent extension is a standard processor. This processor uses a GeoFence to check whether the current point is inside or outside this polygon. If the result is outside the polygon, it will be considered and outlier and will not be processed any further.

4.2.2.2 AVERAGE/MEDIAN PROCESSOR

The first processor is the median or average calculator. In order to calculate the median or average of the present measurements and the previous measurements, the previous results have to be saved somewhere. This is done in a feature service, because it is quite easy to save values to and get values from a feature service with the GeoEvent processor.

```
String url = "http://cgi.godenzonenvanatlas.nl/arcgis/rest/services/HesterTrainTracking/PreviousResults/FeatureServer/0/query?where=sensorId+%3D+%27"
+nodeName +
"%27&objectIds=&time=&geometry=&geometryType=esriGeometryEnvelope&inSR=&spatialRel=esriSpatialRelIntersects&relationParam=&outFields=*&returnGeometry="
+ "true&maxAllowableOffset=&geometryPrecision=&outSR=&gdbVersion=&returnDistinctValues=false&returnIdsOnly=false&returnCountOnly=false&orderByFields=&"
+ "groupByFieldsForStatistics=&outStatistics=&returnZ=false&returnM=false&f=pjson";

String response = getData(url);

ArrayList<Point> points = getPoints(response);
```

FIGURE 29 THE REQUEST FOR THE PREVIOUS RESULTS

When the previous GPS measurements have been extracted from the Feature Service for that specific device, either the average or the median method is invoked. If the average method is invoked, all x coordinates are added up and divided by the number of coordinates, and the same goes for the y coordinates. The median calculation works by ordering all x coordinates and getting the value of the middle one, and also for the y coordinates. The resulting x and y coordinates are given to the rest of the GeoEvent.

```
public Point getAvarage(ArrayList<Point> points){
    double total_x = 0;
    double total_y = 0;
    for (int i=0;i<points.size(); i++){
        double x = points.get(i).getX();
        total_x = total_x + x;
        double y = points.get(i).getY();
        total_y = total_y + y;
    }

    double AvX = total_x / (points.size());
    double AvY = total_y / (points.size());
    Point FinPoint = new Point(AvX, AvY);
    return FinPoint;
}
private Point getMedian(ArrayList<Point> points) {

    ArrayList<Double> xpoints= new ArrayList<Double>();
    ArrayList<Double> ypoints= new ArrayList<Double>();

    for (int i=0; i < points.size(); i ++){
        xpoints.add(points.get(i).getX());
        ypoints.add(points.get(i).getY());
    }
    Collections.sort(xpoints);
    Collections.sort(ypoints);

    double medianX = getMiddle(xpoints);
    double medianY = getMiddle(ypoints);

    Point medianPoint = new Point(medianX, medianY);
    return medianPoint;
}
```

FIGURE 30 MEDIAN AND AVERAGE CALCULATOR

In Figure 30 a fragment of the code of the processor is shown where the average and median methods are given. The whole code can be found in appendix B.

4.2.2.3 MAP-MATCHING PROCESSOR

The second processor will find the rail that is closest to the calculated point, and will then also find the closest point on that rail. As mentioned in paragraph 3.1.2, the rail network and its location are quite well known in the Netherlands. From this rail network, the rails in a certain marshalling yard can be extracted based on GEOCODE or name. Around the rail network of a marshalling yard, buffers have been added of 3 meters, and these buffers are used to determine to which track the calculated point has to be snapped. This is done by performing and intersection between the buffer feature service and the point. The result of this intersection is a list of all IDs of the intersecting buffers. This ID corresponds with the IDs of the tracks themselves, so these can be queried afterwards. If the point does not lie within the buffer, a track number of -1 is appended and the snapping will not occur, instead, the coordinates will remain the same as before the snap calculator. The reason to use buffers is to limit the amount of possible railway tracks the processor has to snap to. If all railway tracks were taken into account, the calculation to find the closest line segment would take a very long time, resulting in a slow process.

The intersection method that has been used on the buffer and the point is a build-in method in the ArcGIS REST point, which constructs a set-theoretic intersection between an array of geometries and a geometry (Esri, 2015). In set theory, intersection is defined as follows: the intersection of sets x and y is the set consisting of those objects that are members of both x and y, this can be denoted as $x \cap y$. The way this works under the hood is by checking for each of the available geometries (in this case polygons), if the point is inside or outside this geometry (Devlin, 1993)

When the rail ID is known, the rail can also be extracted from the feature service. A method created by Esri is invoked to get the closest point on the track. The track that has been found using the intersect method and the current point are the input, and closest point on the line will be the output. The method is part of Esris GeometryEngine class and is called nearestCoordinate.

4.2.2.4 PERCENTAGE ON RAIL PROCESSOR

The third processor that has been written is a processor that calculates the percentage a certain track has been assigned to a certain device. This processor works by taking the previous tracks numbers to which the specific device was matched previously from the feature service (database). It counts the number of times each of the tracks that have been appended, occur in the database and calculates a percentage for each track.

After calculating the percentages, the percentage of the track the current event is matched to will be given as an output. This will serve as an indication of the reliability of the result. The higher the percentage the higher the chance that the rail is actually on this track and that enough previous results have been taken into account that this result can be seen as quite sure. During the analysis phase of the project, it has been investigated whether it would be better not to snap the result to the closest track directly, but to use the track it was closest to most often. However, during the analysis the conclusion came that this would not be beneficial for two reasons. First of all, it was concluded that the result would not benefit from it at all, because at the start of a session, the map matching might occur wrong quite often, and the chance of undoing a correct match was too large. Secondly, the previous results are already taken into account in an earlier algorithm, and taking it into account again only yields extra computation time and unnecessary steps.

```java
ArrayList<Integer[]> percentageTrack = percentageCalculator(trackIds);
int largest = 0;
int resultTrack=trackId;
for (int i=0; i<percentageTrack.size(); i++){
    if (percentageTrack.get(i)[0]>largest){
        largest = percentageTrack.get(i)[0];
        resultTrack = percentageTrack.get(i)[1];
    }
    else {
        continue;
    }
}
```

FIGURE 31 PERCENTAGE CALCULATOR

4.2.2.5 ORDER OF WAGONS PROCESSOR

In this processor, the order of the wagons is calculated. To find the wagons and sensors of which the order should be calculated, all coordinates from sensors that are matched to the same track are extracted from the server. Two parameters are used to determine the distance between the current sensor and the sensors on the same track. The first is the distance, this sensor will already say something about the order, it will show which wagons are close and which ones are further away.

However, it will not yet calculate the order exactly, as a direction would be needed for that. The second parameter is the bearing, which does exactly that. The bearing calculates the angle to the different sensors. For this purpose, it does not really matter whether this angle is compared to the x or y axis, it only matters that the devices on one side of the sensor have an opposite angle from the devices on the other side of the current sensor (Sheppard & Soule, 2008). Using this information, the order of the wagons can be determined.

In Figure 32, the code that determines the distance and the bearing is given. As the RD new coordinate reference system is used in this system, the computations for both distance and bearing were pretty straightforward. The distances are compared to each other and the bearings determine on which side of the sensor they are.

```java
private double getBearingRD(Point point1, Point point2) {
    double x1 = point1.getX();
    double x2 = point2.getX();
    double dx = x1-x2;
    double distance = getDistanceRD(point1,point2);
    double result = Math.asin(dx/distance);
    double resultDegree = Math.toDegrees(result);
    return resultDegree;
}

private double getDistanceRD(Point point1, Point point2){
    double x1 = point1.getX();
    double x2 = point2.getX();
    double y1 = point1.getY();
    double y2 = point2.getY();
    double dx = x1-x2;
    double dy = y1-y2;
    double distance = Math.sqrt(Math.pow(dx, 2)+Math.pow(dy, 2));
    return distance;
}
```

FIGURE 32 CALCULATING DISTANCE & BEARING

During the test, this method could be implemented, but a result could not yet be given to the GeoEvent extension as the current version of this extension does not support an output of arrays. If the system would be implemented, the newest version of the GeoEvent extension would be used and this would not be a problem.

4.2.3 OUTPUT VISUALISATION

The output of the GeoEvent Service writes the event and all attributes to a feature service in ArcGIS for server. This feature service can be used to visualise the result on a map, by importing it into ArcGIS online. ArcGIS online has a function to update the results every couple of seconds, this time can be set in the attributes. This means that the events will be visible in real time on the map. This map can also serve as an input for the Operations Dashboard.

The Operations Dashboard is another tool developed by Esri that enables the user to bring together a common view of the systems and resources that are managed and to monitor real time feeds for large scale events or day-to-day operations. This can be installed on desktop or tablet devices and can thus be used to distribute the system to the different stakeholders involved.

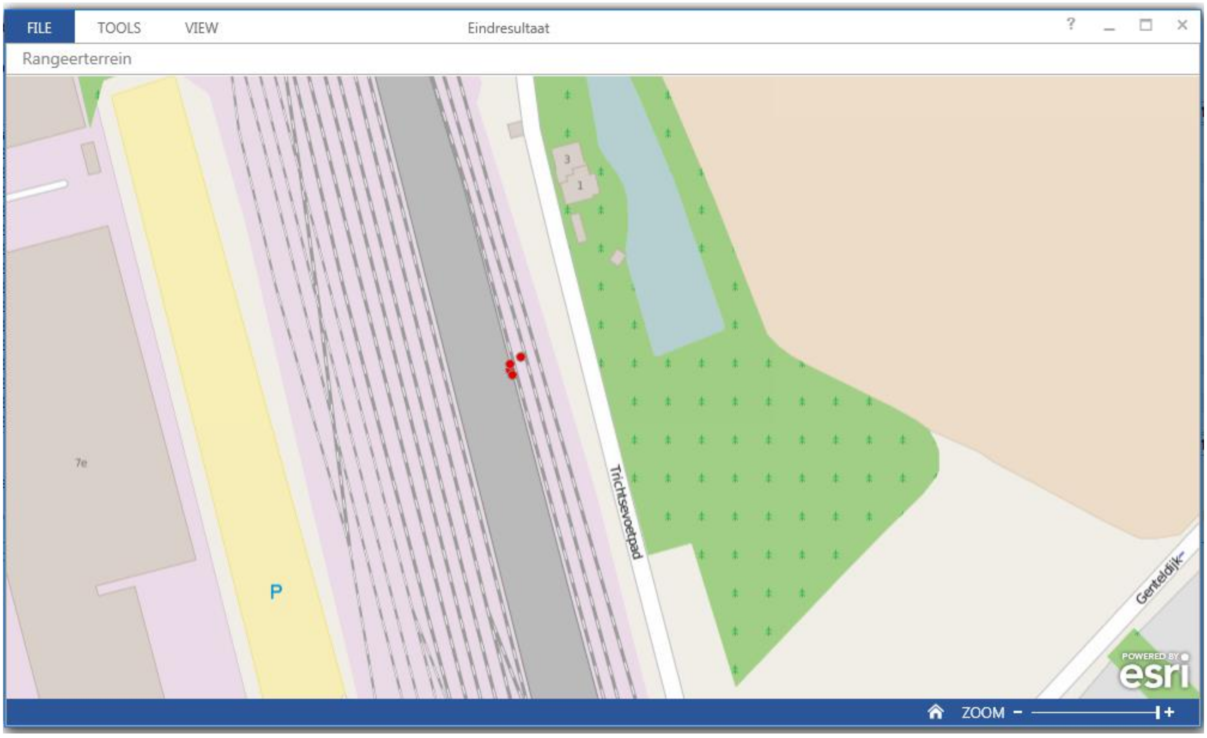


FIGURE 33 THE RESULT IN THE OPERATIONS DASHBOARD

The two outputs as mentioned above are not the only possible outputs. First of all, outputs to files (JSON, CSV) are used a lot in the development phase to debug and analyse the system. But there are more possibilities that are more interesting. For example alerts could be sent to different stakeholders if trains start moving when they are not supposed to, when different loads are next to each other against regulations etc. These outputs are not implemented at the moment, but it is quite easy to implement these using the GeoEvent extension when all necessary data is present. This, however, was not in the scope of the project and these outputs were therefore not implemented (yet).

5 TESTS AND ANALYSIS

In week 18 and 19, the complete system was tested to see whether it worked and how it performed. In this paragraph the original plan and the executed plan will be described, followed by a description of the test results and a conclusion based on the tests results.

5.1 TESTING SCHEME

5.1.1 ORIGINAL SCHEME

The best way to test the system is to test it on a real marshalling yard. In order to achieve this, there was contact with the Dutch railway company ProRail and a transporter called Locon. Locon is a small transporter active in Belgium, the Netherlands and Luxemburg. One of the regular transports of Locon is between marshalling yard de Kijfhoek in Moerdijk (NL) and the marshalling yard in Vlissingen (NL). As this train drives back and forth over the same track a couple of times a week, and stops at the same marshalling yards during those transports, this would be a perfect case to test the system by attaching Undagrids GPS boxes to the train for about a week. The train will be at the marshalling yard in Moerdijk a couple of times for about a night, in which data can be collected and the system can be tested real time.

There would be two options to collect the data, first of all the gateway could be placed in the locomotive. This would mean that data would always be collected, not only when the train is on a marshalling yard. In order to only analyse the data for the case of this project, when the train is standing still, a feature has to be built in to the system to see whether the train is moving and how long it has been standing on the same location. The final system Undagrid developed has an accelerometer, which could be used to do this, but this system was not available during this project. If the gateway was in the locomotive, the gathered data could also be used to test the system for when the train is driving, even though this is outside the scope of this project.

Another option is to place the gateway on the marshalling yard. This would be better and easier for this purpose, as the data is only collected when the train is on the marshalling yard, the accelerometer would not be necessary because there is not much data collected while the train is still driving. This method could also be easier because the power supply for the gateway could be better available on the marshalling yard than on the locomotive.

Unfortunately, contact with ProRail and Locon was very slow, resulting in a very limited timespan to organise the test once contact with Locon was made. Both parties realised that the timespan would be too short to test it before the end of this project, so the decision was made to test the system in a plan B test design that had been made on forehand.

5.1.2 ACTUAL SCHEME

Plan B to test the system was to as accurately as possible recreate a marshalling yard and test the system that way. The original idea was to do this on a large parking space, which would represent a marshalling yard. The parking space in question would be the P+R parking in Delft, behind the central station. However, when developing this plan, an aspect of the BBK was discovered that asked for a change in the test scheme. A very important step in the process of localising trains on marshalling yards was to snap the location of the trains to the rail network as an extra reference system. As the dataset of ProRail, which contains the rail network for the whole Netherlands also contains a line for the edge of the platform, the choice was made to test the system on some train stations in the Netherlands. This way, the edge of the platform could simulate a track on a marshalling yard and could be duplicated at a distance of 5.5 meters to represent several tracks. In these tests, the Undagrid devices were placed as close to the edge of the platform as was safe to do, so they were placed on that ‘rail’ on the simulated marshalling yard.

Three tests were performed on the third, fourth and ninth of May on the station of Geldermalsen, a small town in the centre of the Netherlands. During these tests, the devices were situated in pairs, with a distance of 10 meters between each pair of devices. The edge of the platform was used as the line in the rail network and the simulated tracks were placed approximately 5.5 meters from each other. In Figure 34, a top view of the station is shown, in order to see the context of the test.

FIGURE 34 APPROXIMATE LOCATION OF THE DEVICES

Something to take into account when looking at the pictures is that the buildings on the left have almost all been demolished, and the parts that are still high are approximately 10 meters high and far enough away to not play any role in the visibility of the sky. The placement of the devices during the tests can be found in Figure 35.

FIGURE 35 LOCATIONS OF THE DEVICES DURING ALL TESTS AT THE STATION OF GELDERMALSEN

In Figure 36, a picture is shown on how the devices were located on the station, with an actual freight train in the background.

FIGURE 36 TESTING AT GELDERMALSEN STATION (OWN PICTURE)

The test on the 3rd of May took place from 21.15 until 22.09 and on the 4th of May from 19.17 until 19.48. On the ninth of May, the longest test took place from 20.30 until 22.26.

5.1.3 RD VERSUS WGS84

As described in paragraph 3.1.2, the BBK data from ProRail was used as an input to the map matching algorithm. This data is in the RD new reference system, but the data from the Undagrid devices is in WGS84. During the first analysis, which was performed in the previous stage of the project before the P4 presentation, the BBK data was transformed from RD new to WGS84 using the build in transformation in ArcMap. This transformation (Amersfoort_To_WGS_1984_2008_MB) was one of the most recent and most accurate transformations, according to the documentation.

Because of the disappointing results that were gathered during that test, one of the parameters that was investigated was whether this transformation was as accurate as promised and whether the inevitable distortions played part in the disappointing results. The Dutch RD reference system is very different from WGS84. The first one is national system that is based on a grid laid out, it is a Cartesian system, with its origin somewhere in the north of France. WGS84 is a worldwide system which uses degrees to determine position, called a polar system. Because of the fact that it is worldwide, it will pose less accurate results in the Netherlands. This has to do with changes in the earth, caused by tectonic plates, which causes errors in WGS84, but do not have an influence on the result when using RD new. For this reason, RD new is a better reference system to use when just using Dutch data, as it will be the most accurate reference system available for the Netherlands. Another reason to use the RD system is because calculating distances is a lot easier in this reference system. This is because this is not a polar system (degrees) but a Cartesian system (x and y). Using PCTrans 4.2.10, the program for geodetic and hydrographic calculations as provided by the Dutch Ministry of Defence (Minesterie van Defensie, 2013), the GPS data was transformed to RD new using RDNAPTRANS transformations.

5.2 ANALYSIS OF RAW DATA

Raw GPS data will always be in the form of a scattered point cloud, as every point in itself will have an inaccuracy of about 5-20 meters. This will result in a picture on the map that can be seen in Figure 37.

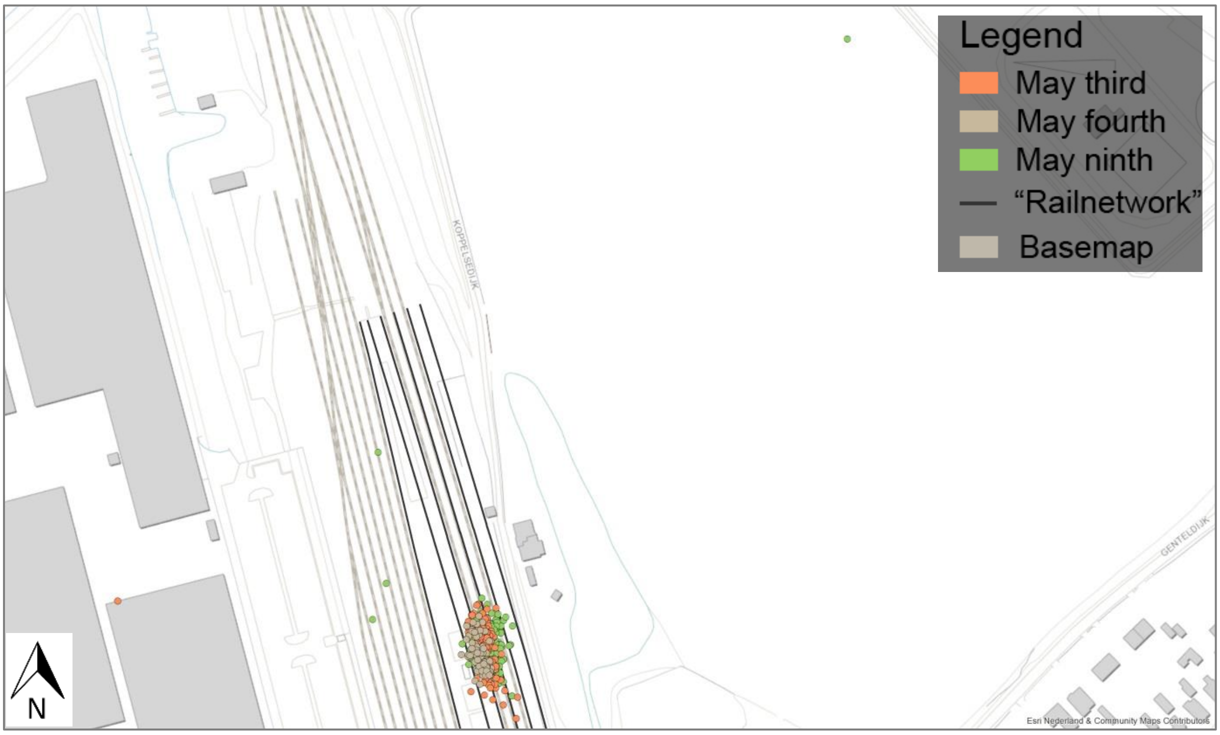


FIGURE 37 RAW DATA WITH OUTLIERS

In this figure, some outliers are visible, these are points that, through for example some error in the GPS measurements, are located very far away from the actual location of the devices. These outliers should not be taken into account when processing and analysing the data, and can be removed easily, both afterwards as well as in real time.

In order to remove outliers real time, GeoFences can be created around the marshalling yard. The GeoEvent Extension can check, using a build in processor, whether the current point is inside or outside the GeoFence. If it is inside the GeoFence, the result is processed, if not, it is discarded and the next event will be processed.

In this analysis, the outliers have been removed in a similar way, by checking whether they were inside a polygon, representing the simulated marshalling yard. The outliers were removed, which resulted in the data as visible in Figure 38.

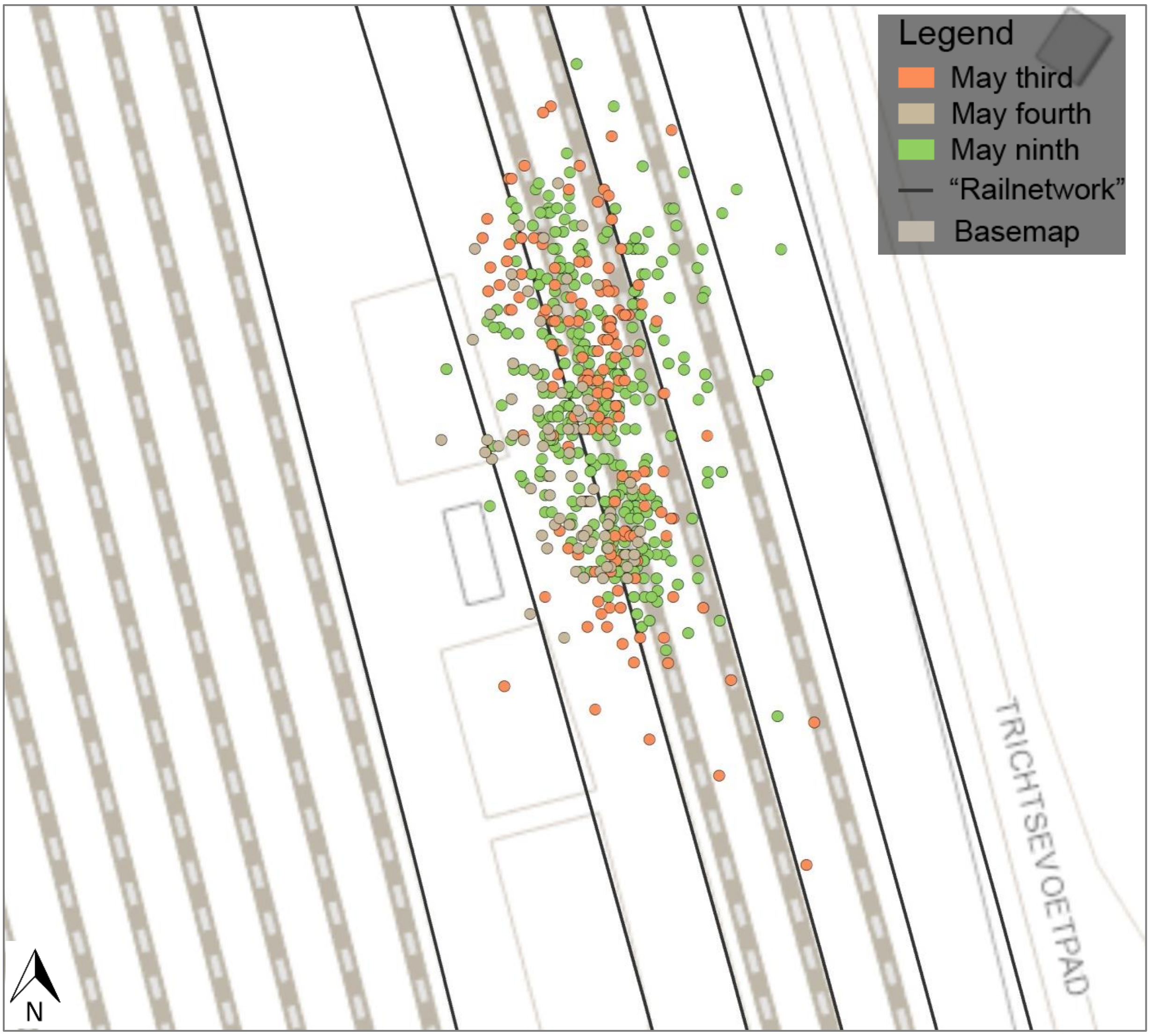


FIGURE 38 RAW DATA WITHOUT OUTLIERS

This image shows just what would be expected, scattered points around the positions of the sensors on the platform. The reason for the point clouds to be a bit oval shaped is that it is data for all five sensors that were positioned with two intervals of 10 meters on the edge of the platform.

During this analysis, one of the parameters that is used to analyse the performance is the distance between the devices. As the devices were placed on set locations with known distances in between them, the distances calculated from the GPS measurements can give an indication of the error in the GPS measurements. The devices were placed at different distances from each other, two at the same location (10 centimetres apart) and then a distance of 10 meters to the next one. Because of these varying distances, not the actual distance, but the variance from the distance is used to analyse the performance.

The distance between the sensors was also calculated for the raw data, to give the basis information to be able to see whether the median and average calculation improves this variance in distance. The variance in distances can be found in Figure 39.

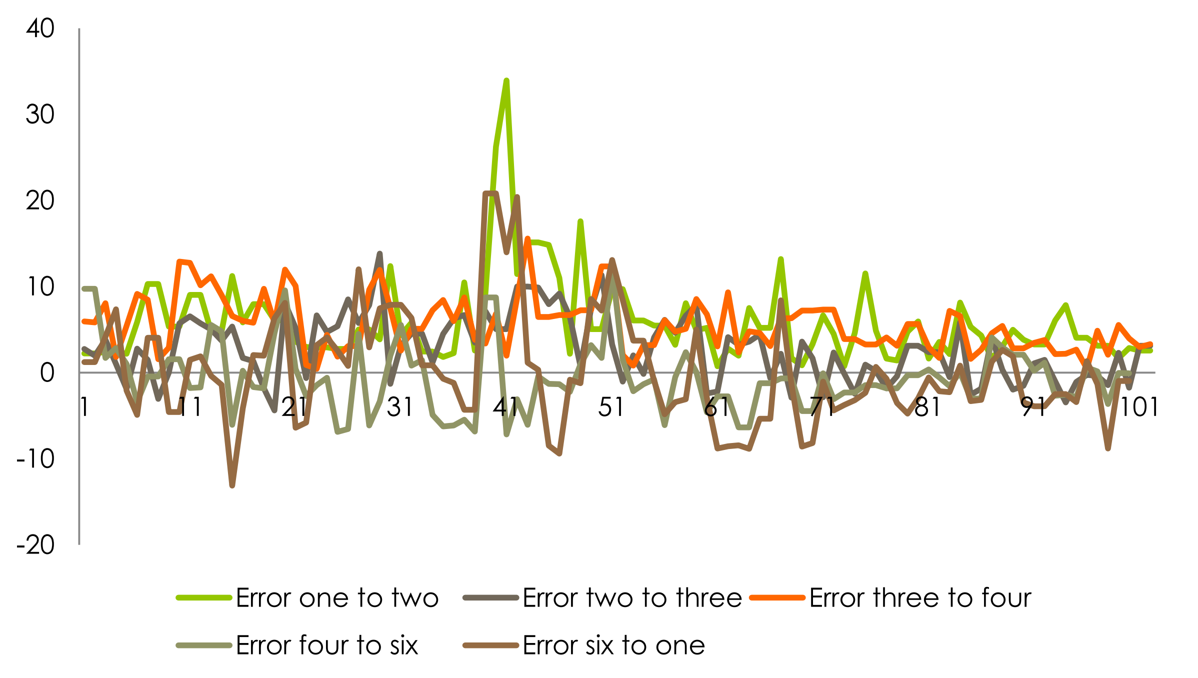


FIGURE 39 VARIANCES IN DISTANCES OF RAW DATA 9TH OF MAY 2015, Y-AXIS: DISTANCE (METERS), X-AXIS: TIME SINCE FIRST 20 RESULTS (MINUTES)

The graph of Figure 39 shows the variances in distances of the data as collected on the ninth of May 2015, very similar graphs have been created for the tests as performed on the third and fourth of May, the graphs of these tests can be found in appendix D. In the graph it is clearly visible that no pattern is visible from the variances in distances of the raw data. The distances do not really get smaller over time, they are just all over the place. This is exactly the kind of result that would be expected when calculating distances between raw GPS data, and the aim of this project is to improve these variances in distance by taking the previous measurements into account. To correctly take into account these previous measurements, some data should first be collected, as a median or an average of too small a number is not representative. For this reason, the first 20 results have not been taken into account during the analysis of both the median as well as the average calculation. These first 20 minutes are the calibration minutes of the system, in which the GPS sensors can locate the satellites and the starting values can be saved to the database in order to be able to perform an average or median calculation after 20 minutes. In the next paragraph, the performance of the average calculation of the previous results will be analysed, and in the paragraph after that the same will be done but then by taking the median.

5.3 ANALYSIS OF AVERAGE CALCULATION

For every test, both the average and the median calculation have been done to detect the difference, compare them and choose the method to take the previous results into account that can be used in the final system. In this paragraph, the results and analysis of the average calculation will be discussed. The average is calculated using the algorithm described in paragraph 4.2.2.2. First of all, the results of the average calculation of all sensors can be seen in Figure 40.

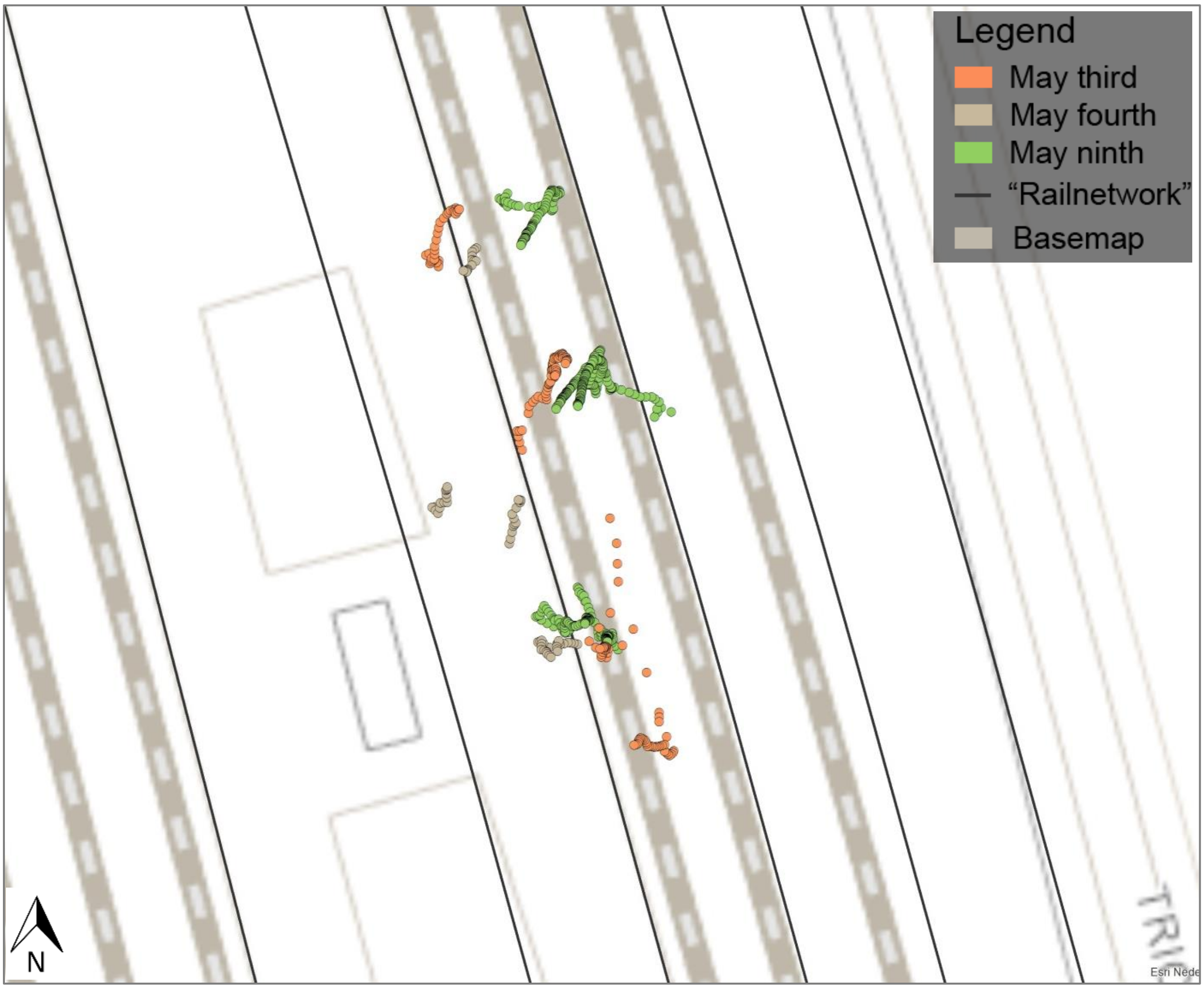


FIGURE 40 AVERAGE CALCULATION ALL AVERAGES

The map above shows a very different picture compared to the raw data. The scattered cloud of points is gone and instead snakelike configurations of points can be observed. Each snake of course is the averaged data from one sensor. In the map representation, it is already clearly visible that the averaging calculation has a large influence on the localisation of the different points. The values are much more concentrated around the actual location of the sensors. The result of the averaging calculator can much more closely be observed when looking at the different sensors separately for one test. To show and discuss the results, the test of the third of May will be discussed initially, but the conclusions that can be drawn are supported by the average calculations done in the other two tests.

FIGURE 41 AVERAGE RESULT OF TEST THIRD OF MAY

In this image, the result of the average calculation becomes clearer than in the previous one. The resulting locations form snakelike patterns on the map, where the result gets closer and closer to the actual result. Every separate sensor has its own pattern of all average results. The behaviour of the average calculation in time can be seen more clearly in Figure 42, where the results are given a colour according to their timestamp, ordered from red to green, with green being the most recent results.

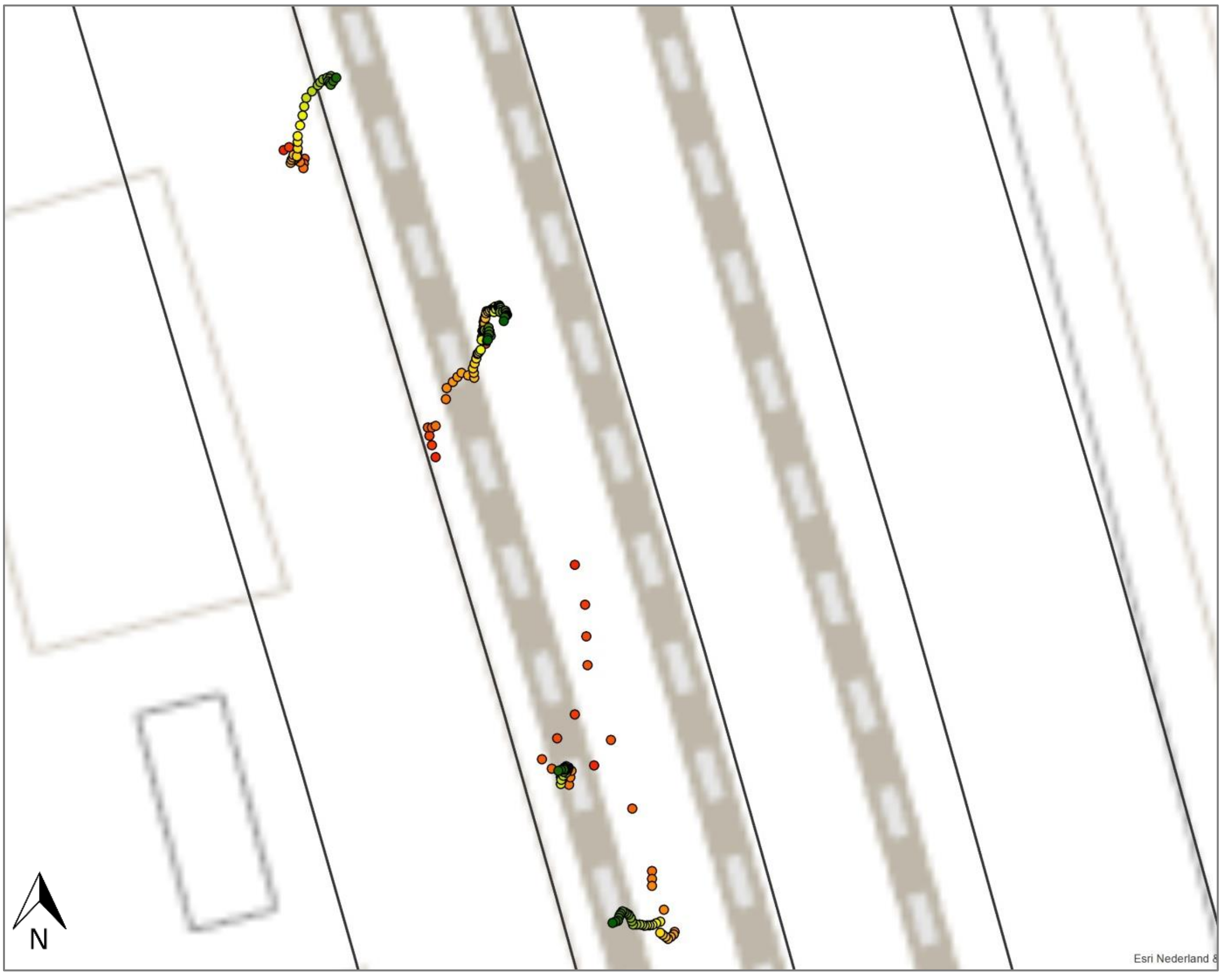


FIGURE 42 AVERAGE OF TEST OF THE THIRD OF MAY BY TIME

The development of the result of the average calculation over time is clearly visible in this image. A disadvantage of the average calculator in comparison to the median calculator should be that the average calculator is more affected by outliers, as explained in paragraph 3.1.1. The outliers have been removed, as was explained, based on whether they are inside or outside the simulated marshalling yard, there will be no really large outliers that can affect the result. Inside the simulated marshalling yard, however, there might still be measurements that are smaller outliers, measurements that are further away from the actual location compared to the rest of the results. These “outliers” will still influence the result and are the cause of the snakelike patterns that are formed by the average calculation result. After enough measurements, however, one small outlier will not influence the result that much, because there are enough results that one outlier does not have that large an influence. Based on the visualised data, it would seem that the more results, the better the result will be.

To get a more accurate analysis on the result and improvement that the average calculation has on the location, the variance in distances has also been calculated and visualised in a graph.

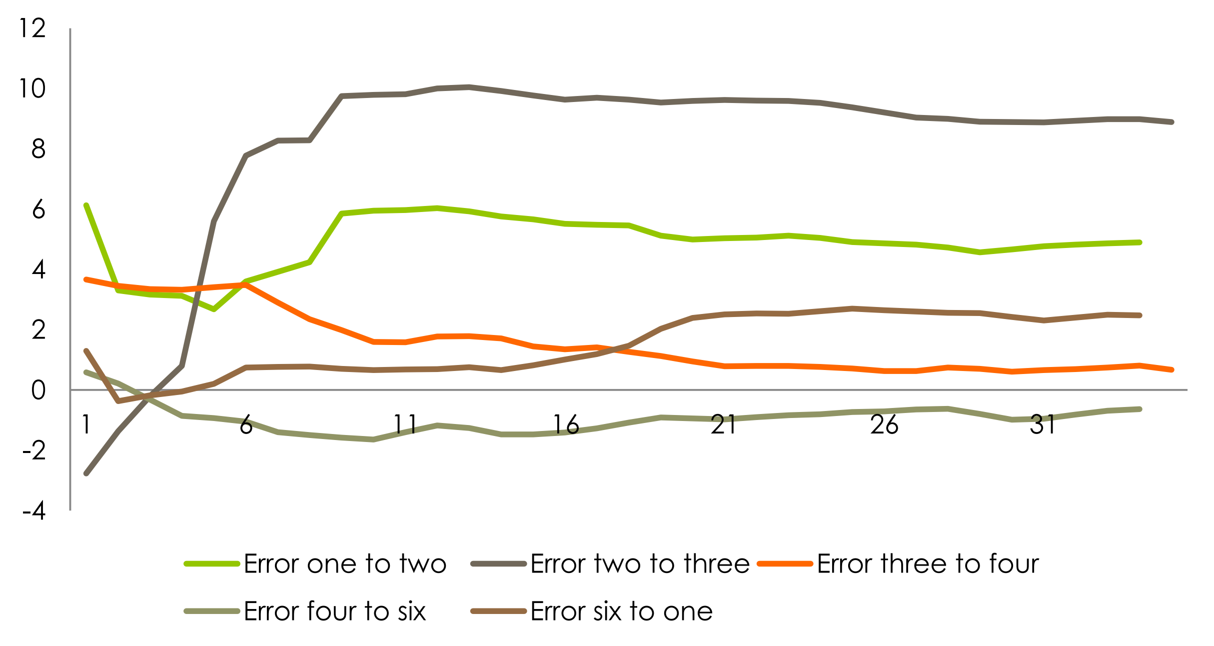


FIGURE 43 VARIANCES IN DISTANCES THIRD OF MAY, Y-AXIS: DISTANCE (METERS), X-AXIS: TIME SINCE FIRST 20 RESULTS (MINUTES)

In the figure it is visible that the average results follow very distinct pattern compared to the variances in distance in raw data. A strange phenomenon in the variance happens in the error of sensor two to three, this is probably an example of a couple of small outliers in the data that causes the result to change quite abruptly. Even though 20 previous measurements per sensor were already stored in the database before the average calculation started, this is not enough to make sure that the result doesn´t change anymore after a couple of small outliers. But overall, the variances in distance become a lot smaller with the average calculator then they were just using the raw data.

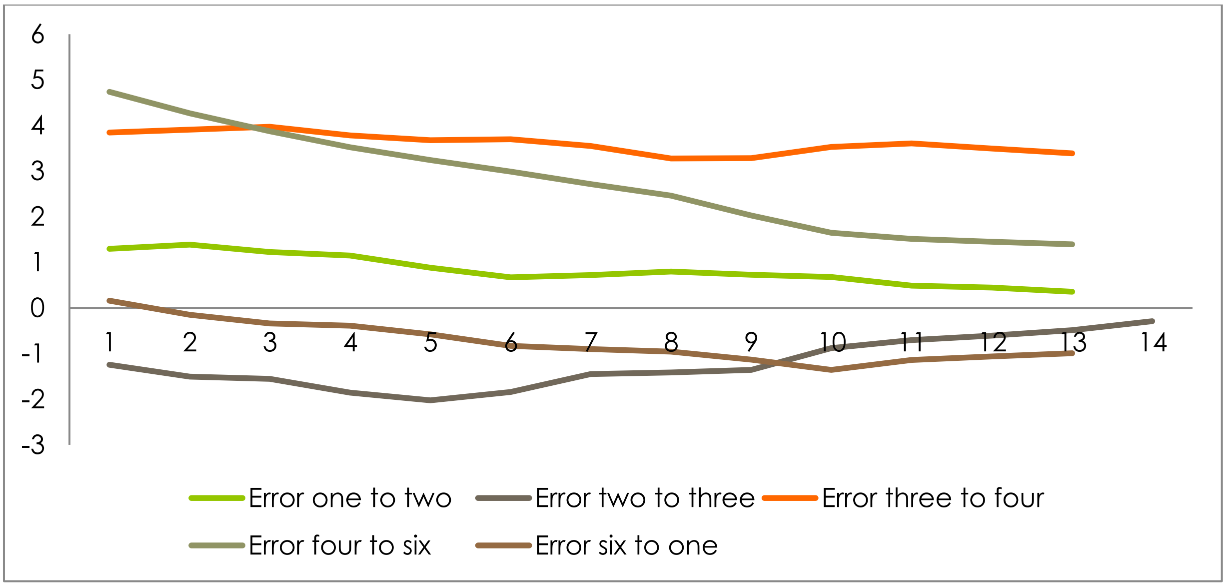


FIGURE 44 VARIANCES IN DISTANCE TEST FOURTH OF MAY, Y-AXIS: DISTANCE (METERS), X-AXIS: TIME SINCE FIRST 20 RESULTS (MINUTES)

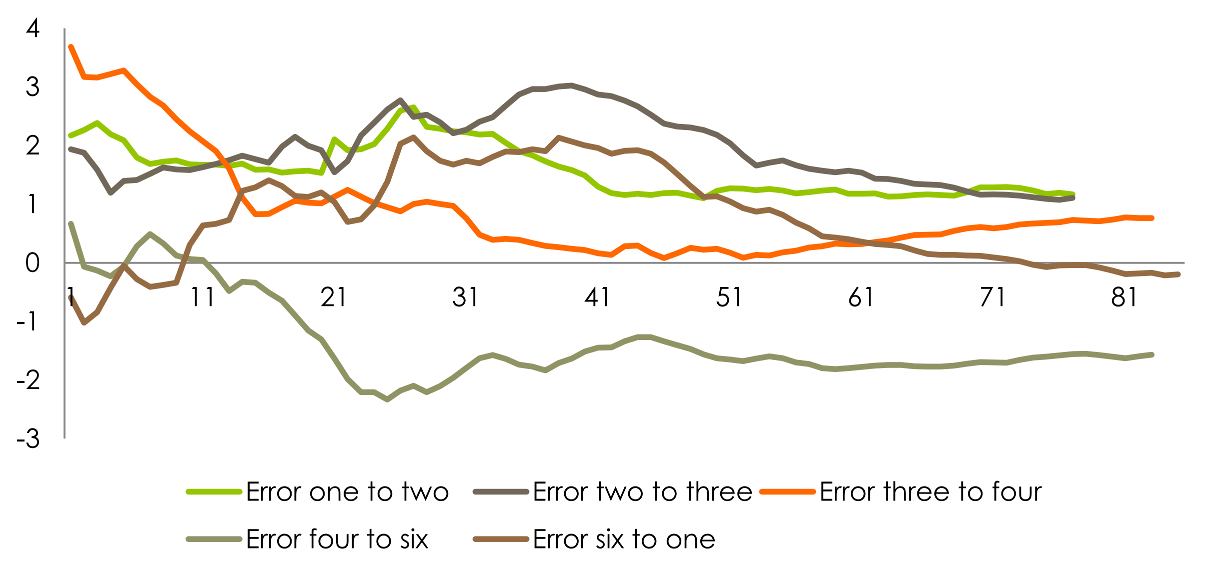


FIGURE 45 VARIANCES IN DISTANCE TEST NINTH OF MAY, Y-AXIS: DISTANCE (METERS), X-AXIS: TIME SINCE FIRST 20 RESULTS (MINUTES)

In both other graphs, the results of the average calculation can be seen as well. In both pictures, a pattern is clearly visible, that shows that the variances are quite small and have a small tendency to move more and more to the zero value during the test. Especially the test of the ninth of May is interesting to analyse, as this was the longest test with the most measurements and therefore most results. It is clearly visible, that while the results still vary in the beginning of the measurements, the more measurements are added, the more the results stabilise and get to zero. From the 61 measurements, so after an hour, the resulting variances in distances are all below two meters, so significantly more accurate then when raw GPS data would be used.

The next step in the analysis of the average calculator is to see the behaviour in combination with the map matching algorithm. The map matching algorithm, as explained in 4.2.2.3, will take the resulting point location and look for the closest point on the closest rail. As a wagon will always be on somewhere on the rail network, this is an important measure to take into account. The idea is that with the improvement of location accuracy, the measurement will be matched to the correct rail most of the time, and that after a certain amount of time, the result will always be matched to the correct rail. In order to most accurately incorporate this time aspect, it is most suitable to look at the results of the last test, on the ninth of May, as this was the longest test and will therefore show the influence of time on the improvement of the result the best. In Figure 46, the matched results have been visualised on the map.

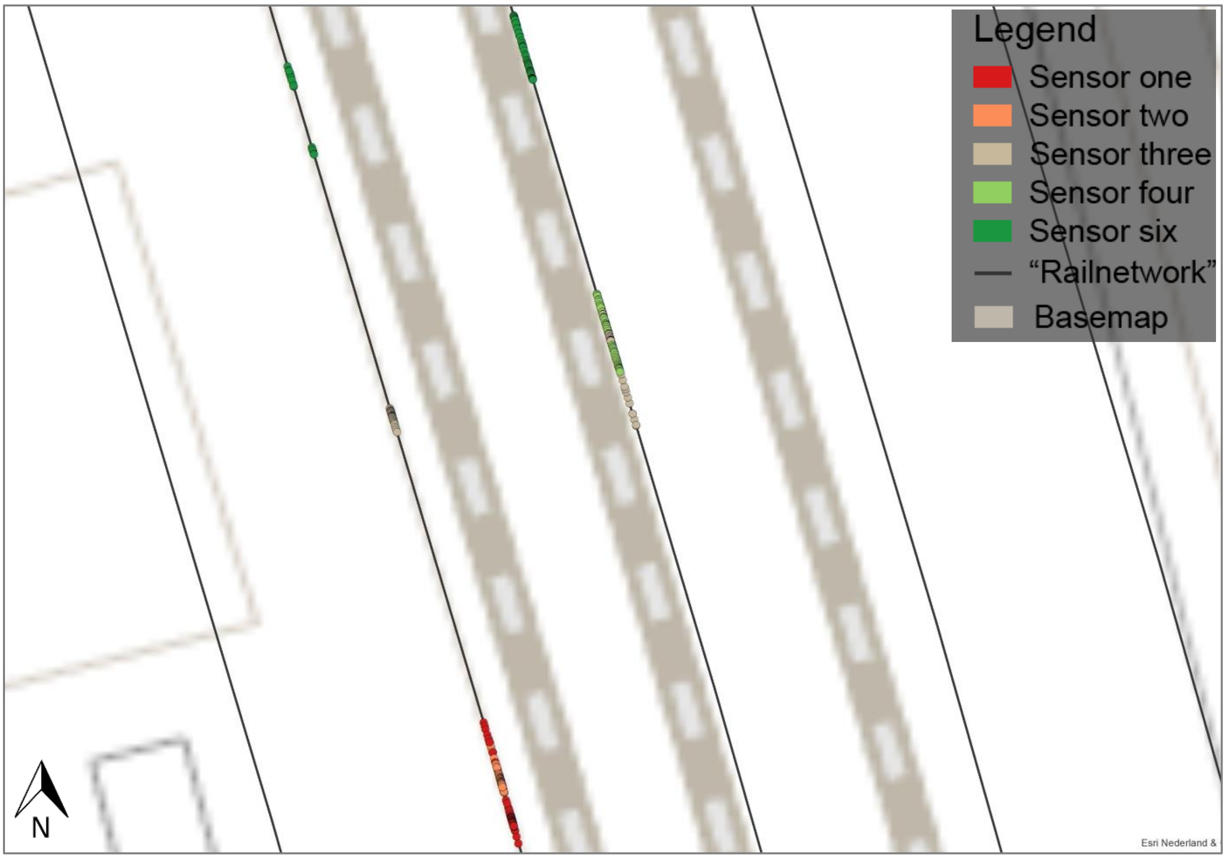


FIGURE 46 MATCHED RESULTS OF TEST NINTH OF MAY

As is visible, most results are matched to the correct track (the second track from the left), but also a large part of the results have been mapped to the track next to it. This has to do with the basic accuracy of GPS, which is bigger than the distance between the two rails. So on the first glance, the result is quite good, but not spectacular as still a large part of the results is mapped to an incorrect track. It will get more interesting, however, if the time aspect is taken into account. When the results are given a colour according to the timestamp (red to green), the resulting map gives a lot of information, as can be seen in Figure 47.

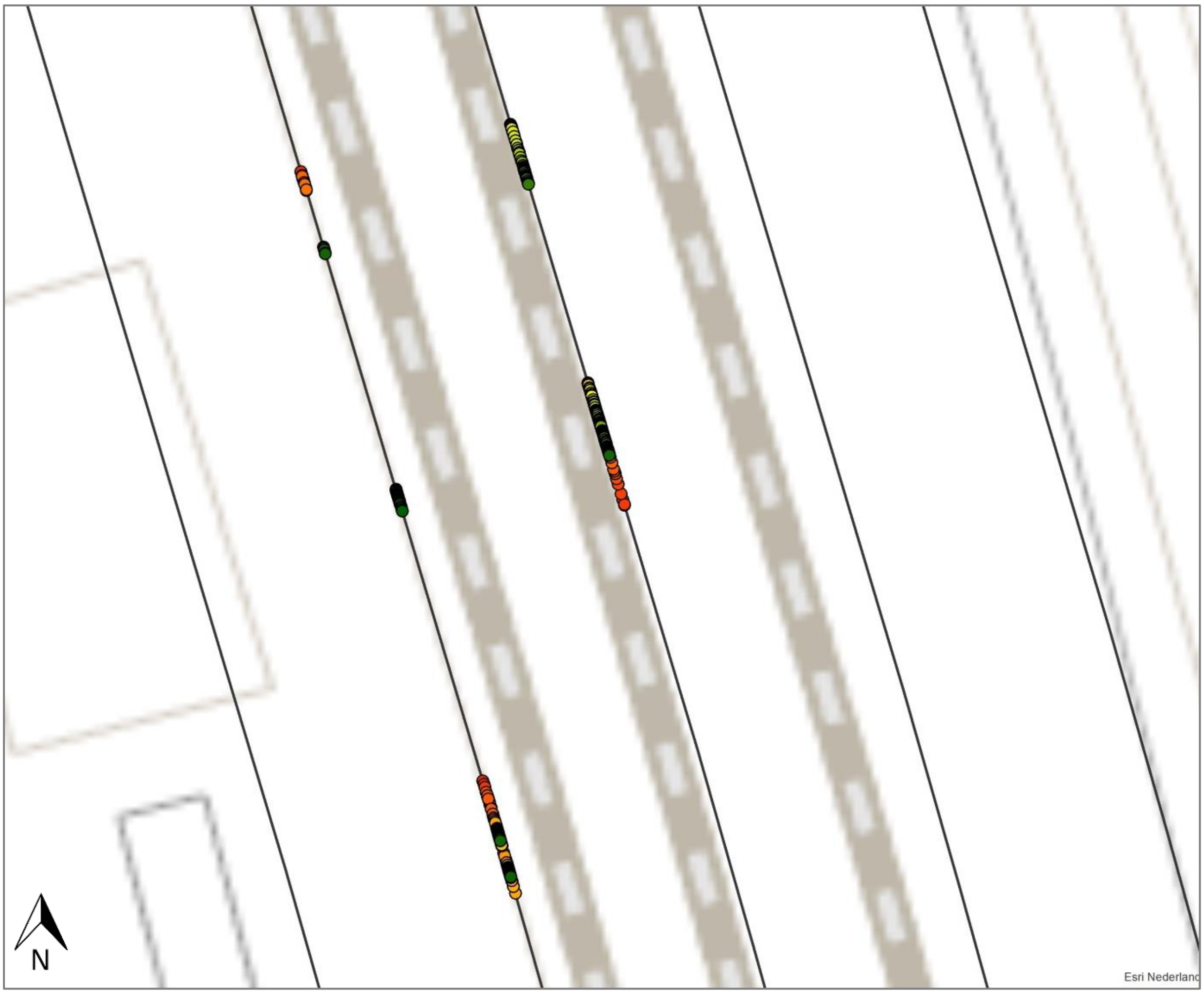
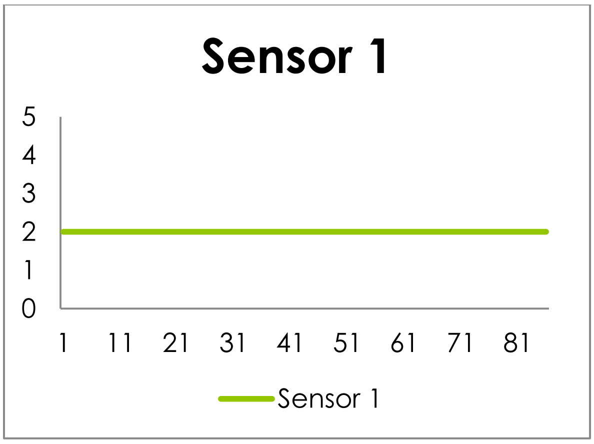
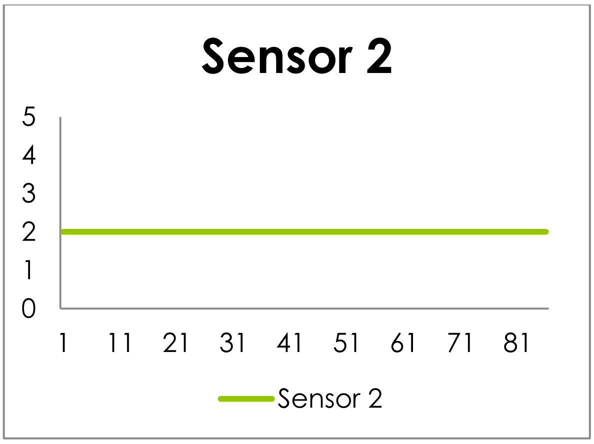


FIGURE 47 RESULTS MATCHING COLOURED BY TIME

On this image, it is can be seen that even from the sensors of which a lot of the results are on the incorrect track, the most green coloured results are on the correct track. This means that when more time has passed, the results are matched to the correct track more often, and in the end it seems that almost all results are matched to the correct track. This can better be visualised in a graph, like in Figure 48.





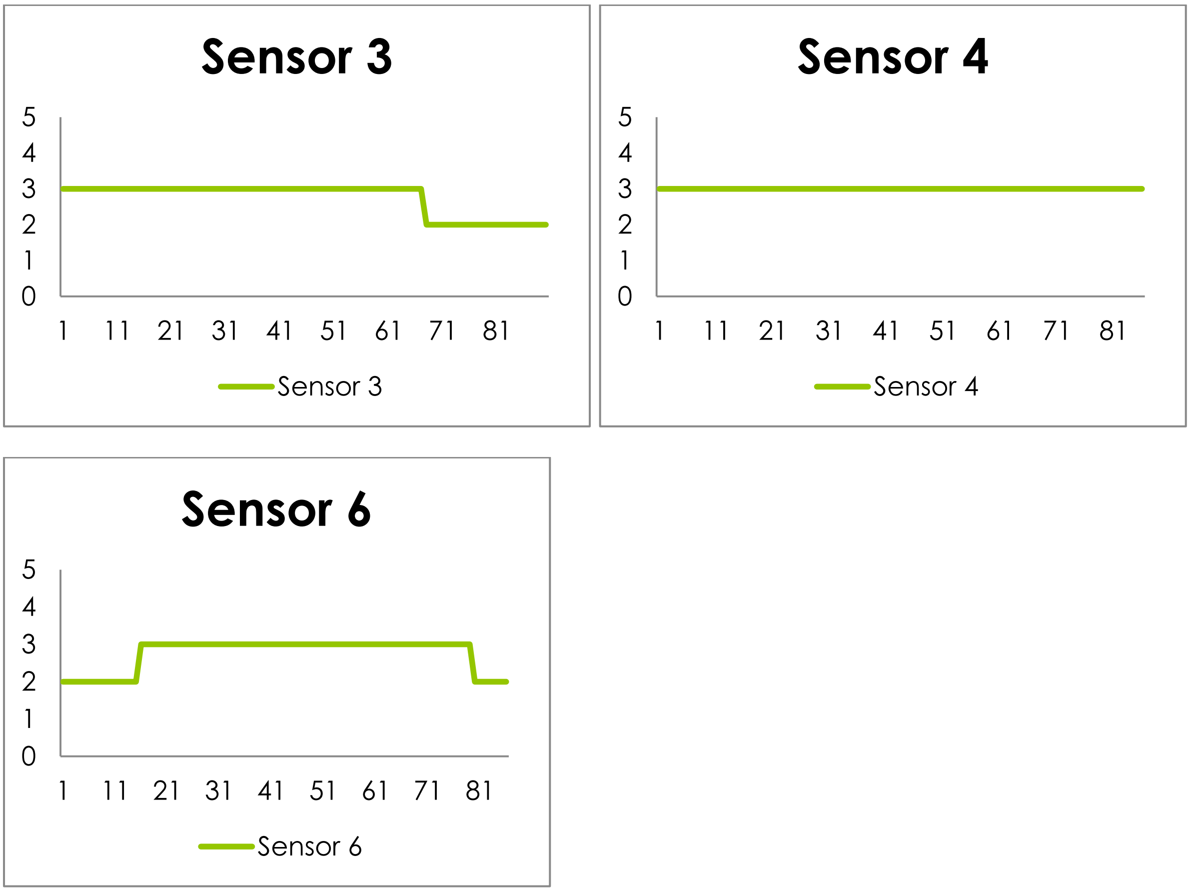


FIGURE 48 GRAPHS OF MATCHED RAIL, Y-AXIS: DISTANCE (METERS), X-AXIS: TIME SINCE FIRST 20 RESULTS (MINUTES)

In this image, a similar pattern is visible that was already noticeable in the map as shown in Figure 47. The results are matched to the correct track (track two) or the track next to it (track three). After a little over an hour, the result of four of the five sensors is matched to the correct track. Only sensor four, for some reason, is not matched to the correct track at that point. This is an interesting graph as it shows the development of the result over time.

Based on all data that was gathered and analysed in all three tests, between 51% and 88 % of the results was matched to the correct track, and in all cases the result was matched to the correct track or one of the tracks located immediately next to it. So using the average calculator, a little over half of the results will be matched correctly and all of them will be matched on a scale of the correct track or one of its neighbours. The results of the matching will get better when more results are used, resulting in almost all sensors matched correctly after five quarters of measuring data.

This result is already quite good, as in most cases, the possible location of the wagon can be minimised to one of three tracks. However, the hypothesis was that using the median calculator, the results would be better because (small) outliers would have a smaller effect on the result. This will be analysed in the next paragraph.

5.4 ANALYSIS OF THE MEDIAN CALCULATION

As described in paragraph 4.2.2.2, the median calculation takes the middle of all previous measurement and the current measurement. It is expected that using this as a location will significantly improve the result of the location of the train and of the track to which the train is mapped.

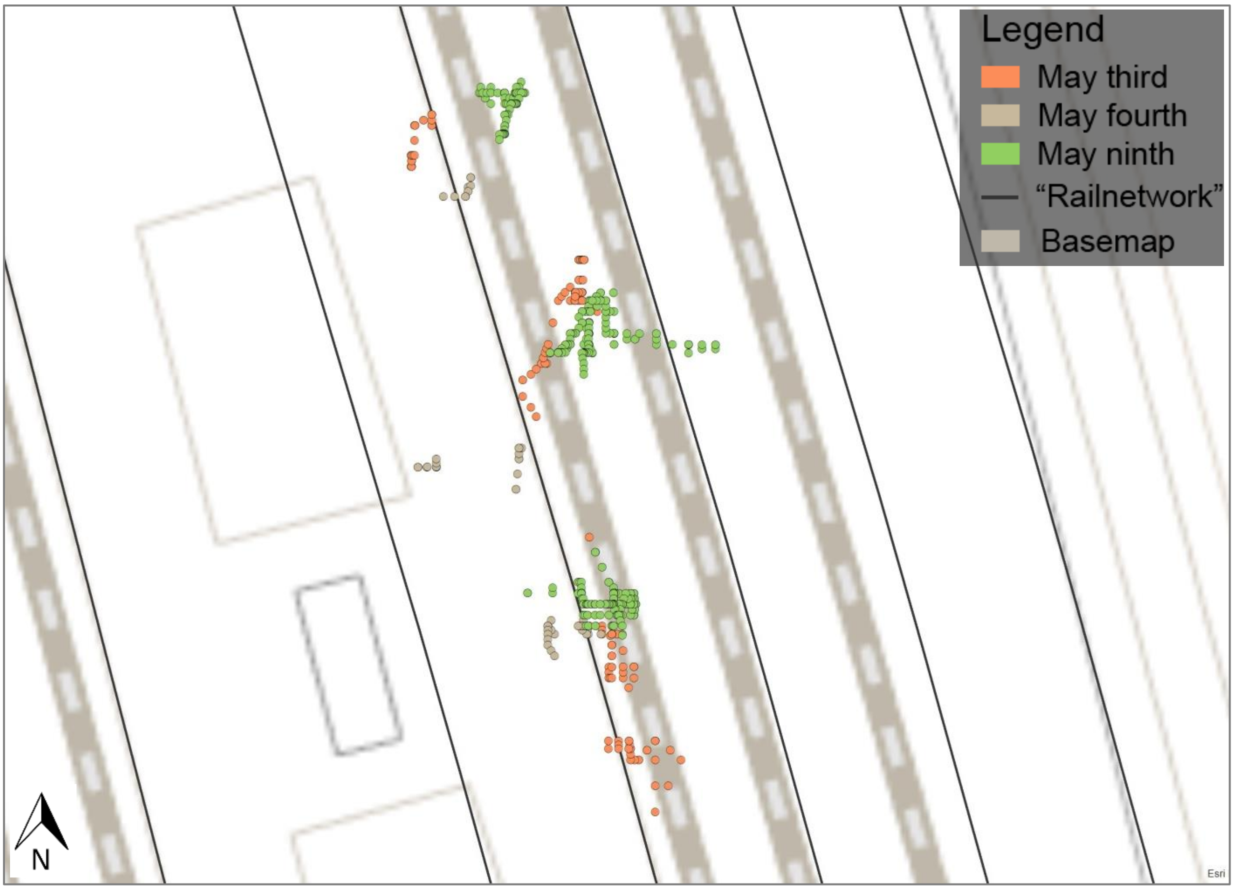


FIGURE 49 ALL MEDIAN RESULTS

The first observation that can be made with respect to this map is that the pattern of the median results is quite different from the pattern of the average results. Where the average result has a distinct pattern in the snakelike figures that appear on the map, the median calculation does not really have such a pattern. The only thing that is visible in the median results is that it seems to sometimes form a sort of raster, where the results are straight next or above each other. This has to do with the way the median calculation works. It takes the median of the x coordinate and the median of the y coordinate separately, which means that sometimes only one of the two parameters "jumps" to the next value. This may sound strange, but can very easily be explained based on a list of numbers representing the x and y coordinates, as can be seen in Table 3. These numbers are randomly chosen, but can be used to illustrate what happens with that so-called jump. When the x values and y values are ordered and the median value is taken, this will result in a coordinate of (3, 12). This will be the result of the median calculation. If a point is added with a new measurement with the coordinates (4,13), the median value for the x coordinate will remain 3, but the median value for the y coordinate will change, and will become 12.5 because of the newly added value. This is what causes the grid like structure in the median result.

TABLE 3 EXAMPLE X AND Y

X	Y
2	11
3	12
3	13
4	13
3	14
2	10
6	13
1	10
9	12

Similarly to the analysis of the average calculator, it is easier and clearer to analyse the results based on one of the tests. To make the two analyses comparable, the analysis will initially be done on the test of the third of May, with interesting data and results of the other two tests to back everything up. The first step is to just show the results of the median calculation for the different sensors for the third of May in Figure 50.

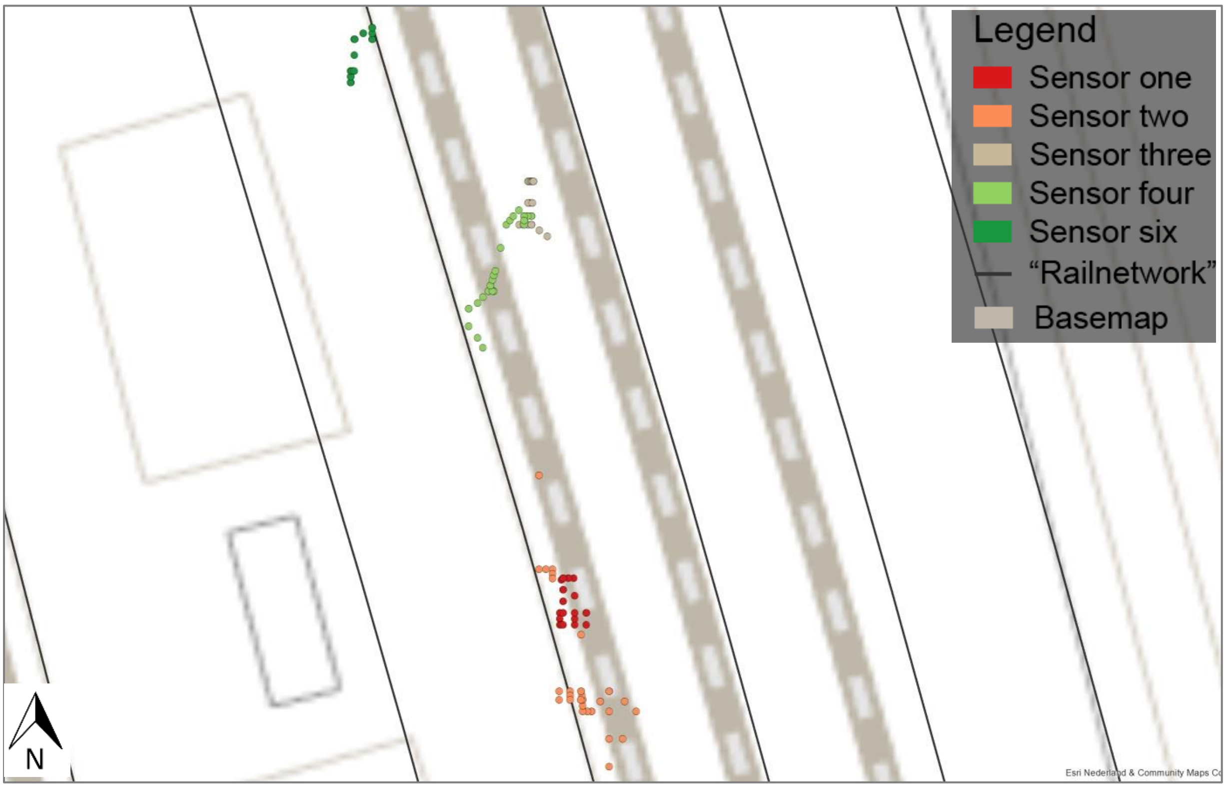


FIGURE 50 MEDIAN RESULTS OF TEST THIRD OF MAY

The fact that the sensors were located in pairs of two (one and two and three and four together) is clearly visible in the image. It can also be concluded that the median calculation has a large influence on the results and the resulting locations are very different from the original point cloud. What is interesting is that the locations of the different results are grouped denser than they were in the average calculation. In order to properly analyse the results, it is important to look at the calculated locations over time, to see how they develop. The result of the measurements with a colour according to the timestamp can be seen in Figure 51.

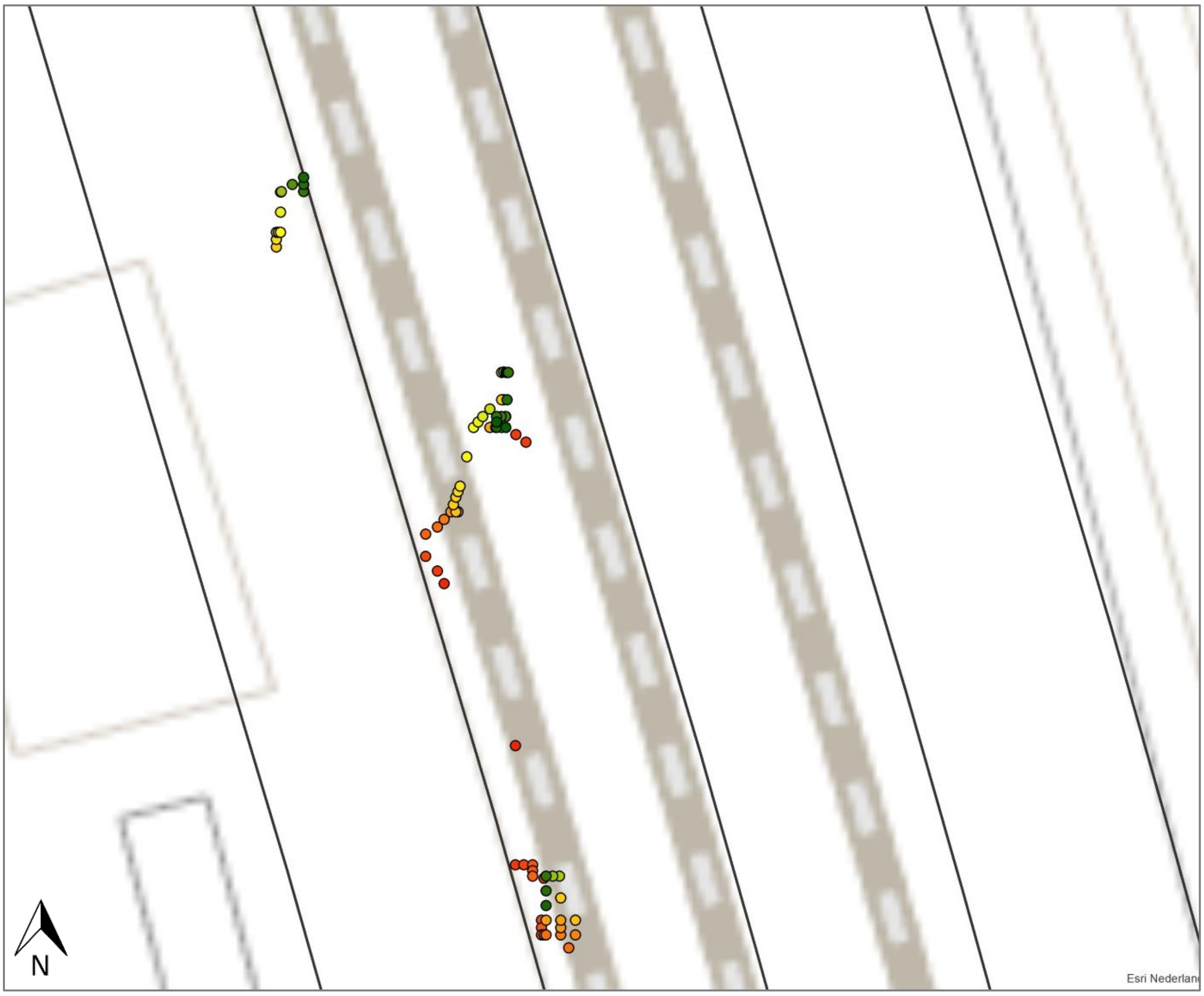


FIGURE 51 MEDIAN RESULTS COLOURED BY TIME

In this image, it can be seen that the results get more grouped together after more time has passed. Where the red and yellow dots can still lie quite a bit apart, the green results are really grouped together. There is no distinct influence of an outlier in the data for a large amount of time like there was in the average result, only sensor three seems to behave a little more like the average calculator, but this is probably because the original measurements of sensor three are quite spread out, of which the influence was also visible in the average analysis. This can be more clearly analysed when not just looking at a map, but when the variances in distances are taken into account.

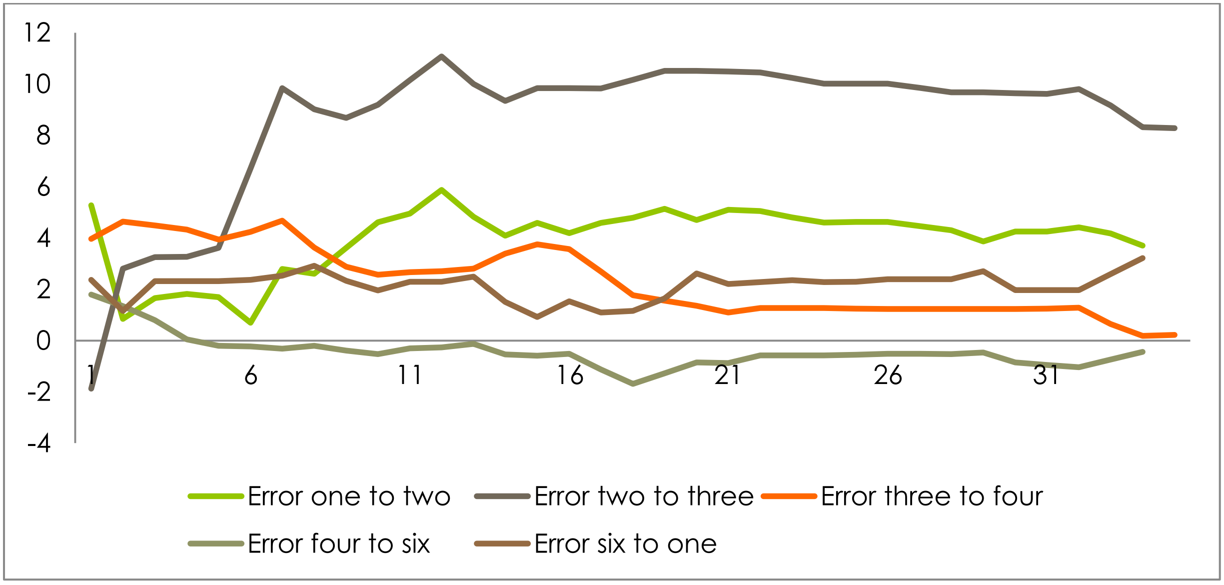


FIGURE 52 VARIANCES IN DISTANCE TEST OF THIRD OF MAY, Y-AXIS: DISTANCE (METERS), X-AXIS: TIME SINCE FIRST 20 RESULTS (MINUTES)

Figure 52 shows the variances in distance over time, this graph is very different from the graph of the variances of distance of the raw data. A clear pattern is visible, and the results get quite good over time when more data is added. After around 20 measurements, the results have pretty much stabilised and a lot of data with very large errors would have to be added in order to make the result change again. To analyse the result and the influence of time on the result even more, the results of the longest test, the test on the ninth of May, have been depicted below.

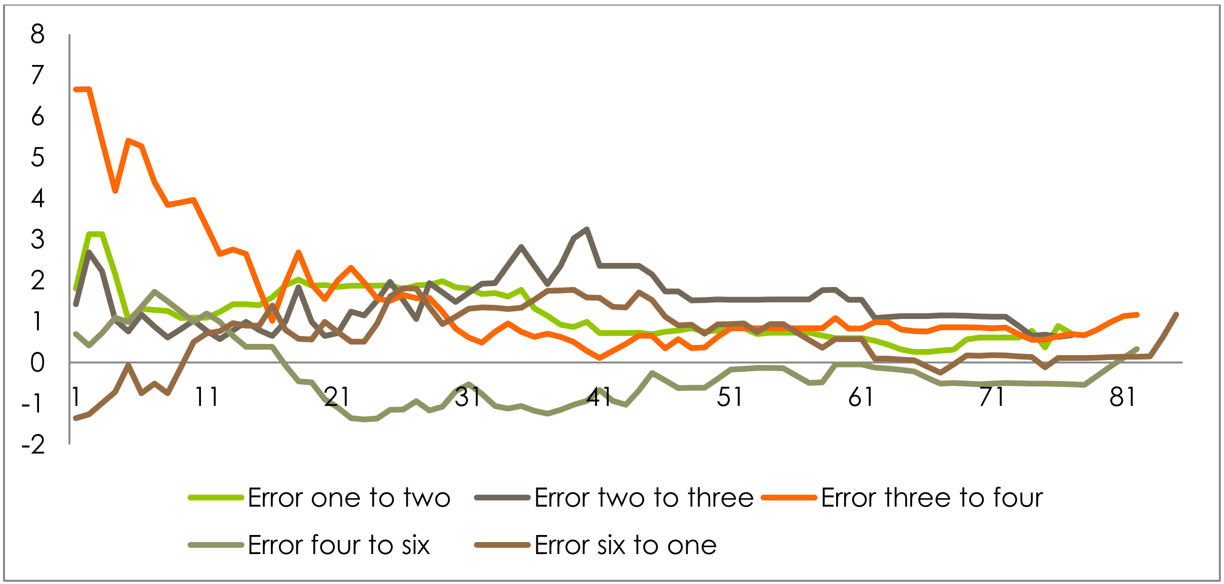


FIGURE 53 VARIANCES IN DISTANCE TEST NINTH OF MAY, Y-AXIS: DISTANCE (METERS), X-AXIS: TIME SINCE FIRST 20 RESULTS (MINUTES)

In this image it is very clearly visible that the more results are added to the database, the more accurate the results will be. After about 45 measurements all variances in distance are below 2 meters, and after about 60 measurements, they are even around only one meter. A very important characteristic of taking this many measurements into account is that when more measurements are added to the database, a single measurement, or even a couple of faulty measurements, will have less and less influence on the end result of the calculation. This is also clearly visible in the graph, where the results can vary quite a bit in the beginning of the graph with only the initial 20 measurements added to the database, where the results vary quite a bit compared to the variance in the results at the end of the graph. At the end of the graph it are really lines that are visible that are sometimes almost straight, whereas in the beginning there are sometimes single peaks where the result changes suddenly.

The next step in the analysis of the median calculator is to see how the map matching happens and develops over time. This is a very important step in the analysis, because the rail network is an important parameter in the localisation of lost train wagons. The first step is to see the results on the map, as depicted in Figure 54. Just like in the analysis of the average calculator, the test of the ninth of May is analysed because the influence of time can be observed a lot better with this test as it was the longest test that was done during this project.

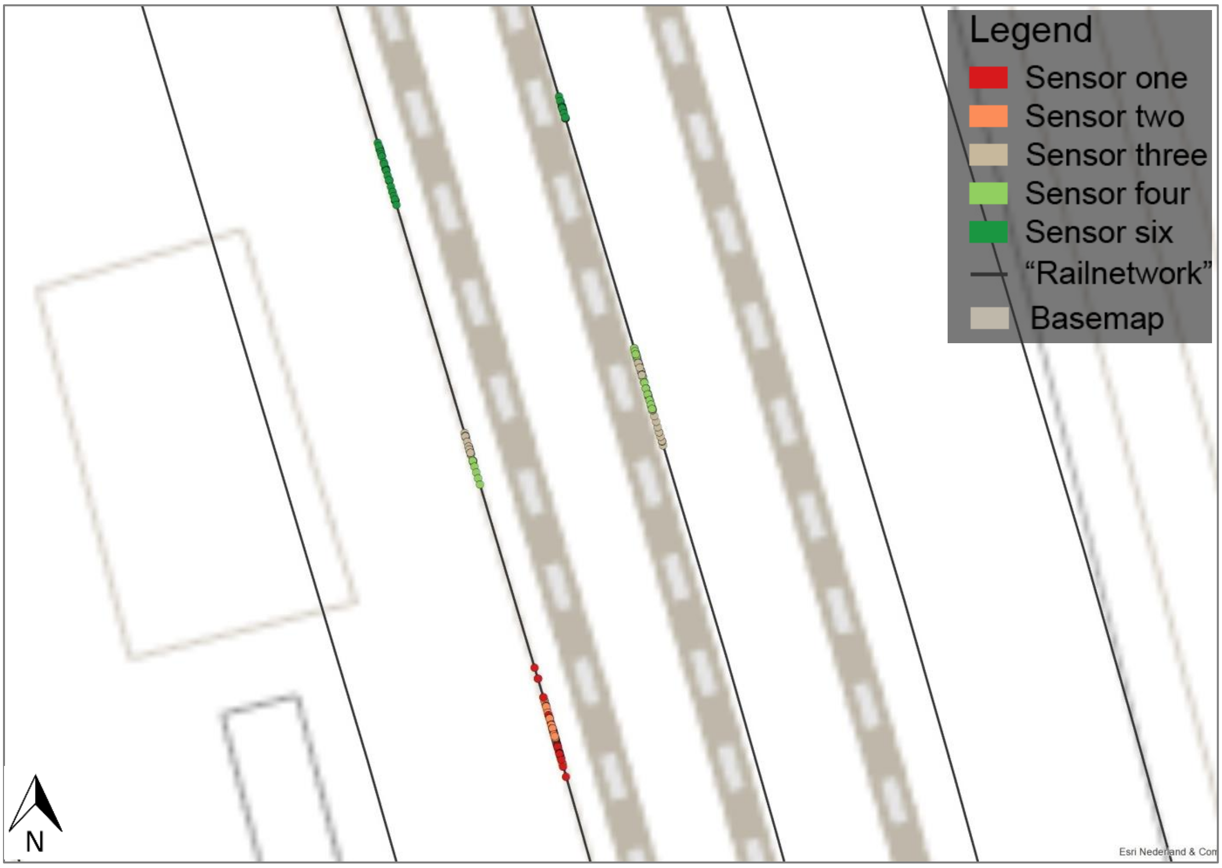


FIGURE 54 TEST THIRD OF MAY MATCHED RESULTS

In this image, it is clearly visible that the largest part of the results are matched to the correct track, but still some results, similar to the average results, are matched to the track next to the correct track. Again sensor one and two are matched best and sensor three and four are matched incorrectly the most. But whether a sensor is matched correctly or not overall is less important than the way the sensor is matched over time. This is the next step in the analysis that can be seen in Figure 55.

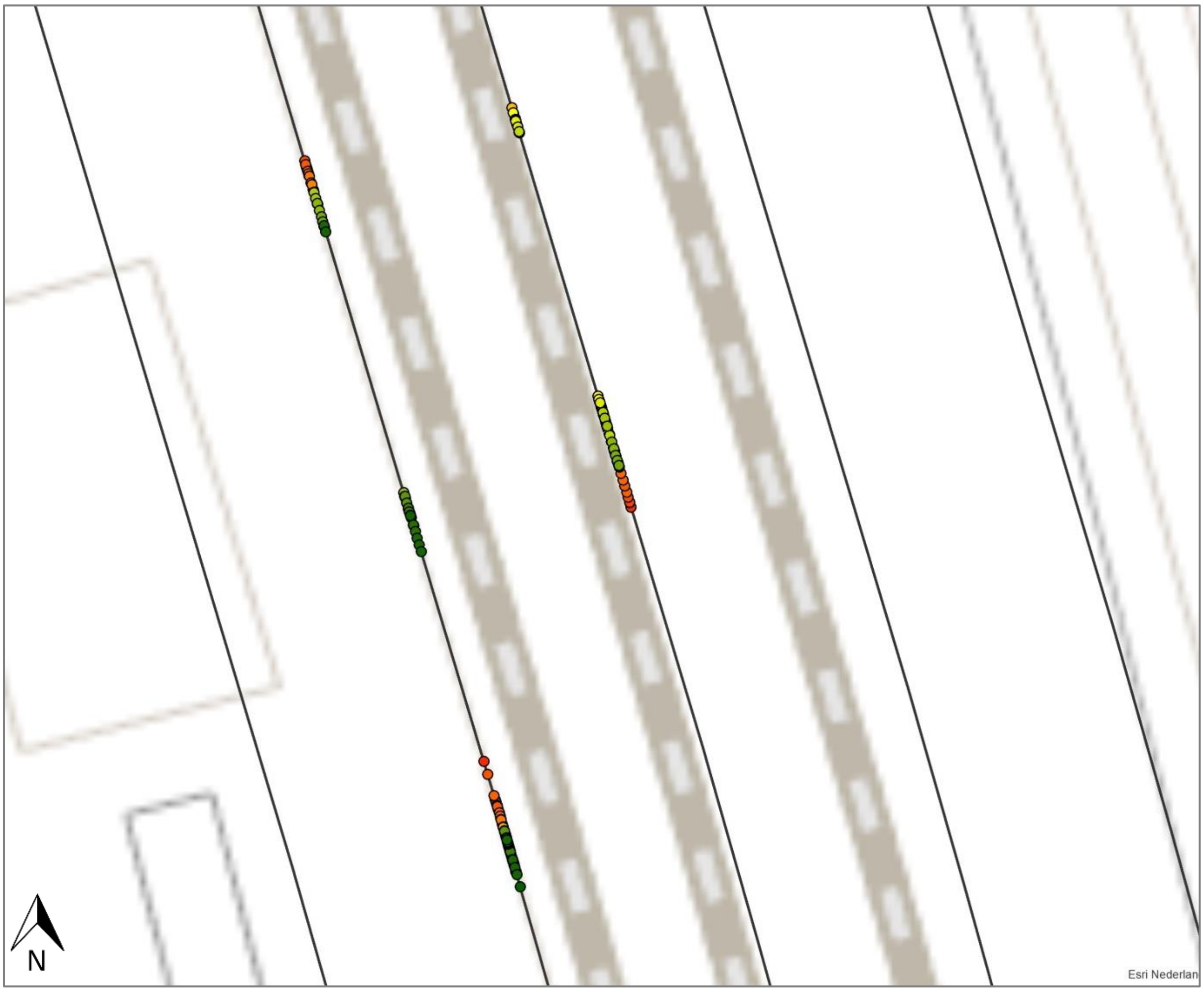
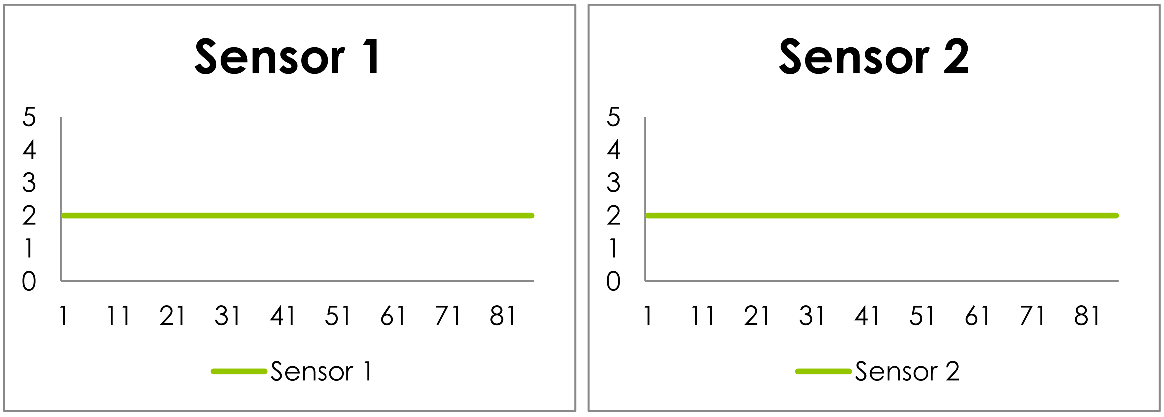


FIGURE 55 MATCHED RESULTS COLOURED BY TIME

One noticeable observation that can be made with this image is that the results that are incorrectly matched, all are not coloured red, the reddest results are all on the correct track. This is a very positive result that shows when more results are added to the median calculation, the map matching happens more often to the correct track. The exact development of to which tracks the results have been matched can be seen in Figure 56.



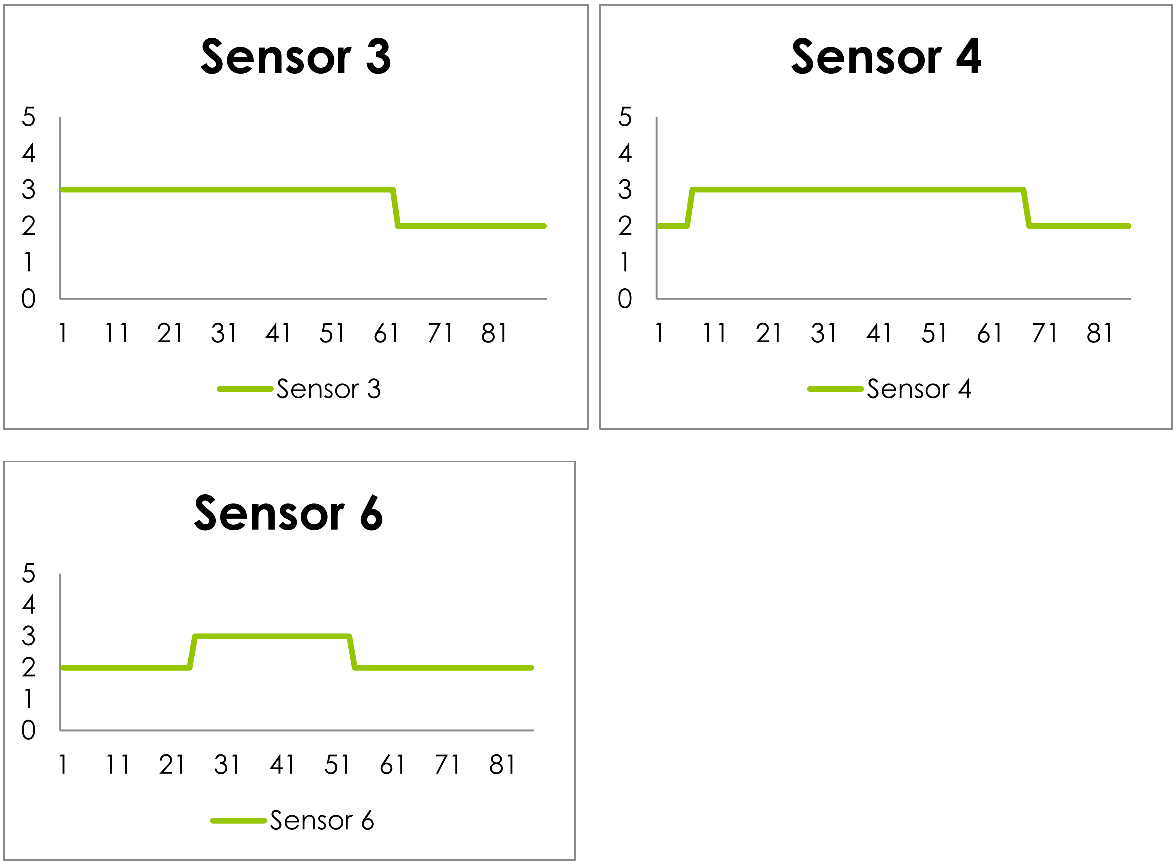


FIGURE 56 GRAPHS OF MATCHED RAIL, Y-AXIS: DISTANCE (METERS), X-AXIS: TIME SINCE FIRST 20 RESULTS (MINUTES)

In this image it is visible that after about 70 results, all sensors are matched to the correct track. The last sensor to be matched correctly is sensor four, which was the sensor that was never matched correctly with the average calculation. This result is very promising, as it shows that when enough time has passed, the results of the median calculation will all be matched correctly. Based on all data that was gathered, between 65% and 80% of all results are matched correctly during the entirety of the test using the median calculation. These are two very important conclusions that can be drawn on the basis of this test.

The fact that after about an hour and a half of testing all results are matched to the correct track has a large impact on the possibility of calculating a correct order of wagons. This will be the topic of the next paragraph.

5.5 CALCULATING THE ORDER OF THE WAGONS

The success of determining the correct track on which the wagons are located has a direct impact on the success of determining the order in which the wagons are on the marshalling yard. As described in paragraph 3.1.3, the order of the wagons is calculated based on the rail to which the results have been matched. Using the distance and the bearing to all sensors on the same track, a list can be made of the order of the wagons.

The calculation of the order of the wagons has also been tried using the data from the test on the ninth of May. This test showed that it is quite easy to determine the order of the wagons once the distance is known. The bearing will either be positive or negative, and all wagons with a positive bearing are on one side of the current sensor, all with a negative bearing on the other side. The distances can then determine the order in which the other sensors are on both sides.

With respect to the test of this method on the data of the ninth of May, it showed that the order of the wagons can be determined. There was error in the order of the wagons that was determined, and this was that sensor three and four were always seen the wrong way around. This has most probably to do with the fact that they were on practically the same location, and therefore no real order could be determined. In the rest of the cases, the order was correctly determined. Seen as the wagons cannot be on the same location, but will always be located a couple of meters from each other, the order can be determined using the current technique.

5.6 CONCLUSIONS TESTING

After all tests were performed and analysed, a couple of conclusions can be drawn about the system and its performance on a marshalling yard.

- The result of GPS measurements can be greatly improved using both an average and a median calculation. Outliers have to be removed on forehand to make sure they do not influence the result and it is only useful to use previous measurements when around 20 measurements have already been collected and stored.
- Matching the result to the network happens correctly in a minimum of 51% of the cases when the average calculation is used and in a minimum of 65% of the cases when the median calculation is used.
- When using the median calculation, which yields the best results, the results are matched to the correct track in all cases after about an hour and a half.
- After the results have been matched, it is possible to determine the order of the wagons on a certain track using the distances between the devices and the bearing. Seen as the sensors on the wagons will always be a couple of meters apart, determining this order will always be possible when the sensors on a certain track are known.

The main conclusion after the tests is that with the time aspect taken into account, this showed that after a certain amount of time, all results will be matched to the correct track, and an order can be determined.

While part of the conclusions of the test could also have been drawn from the theoretical framework alone, performing the tests has been very valuable for this research. First of all, it has been valuable to actually investigate the different aspects of the system individually to see whether and how they performed. It is very important to perform an actual test to see this, as systems that should work in theory, do not necessarily work in practice. In this case the test showed that all of the necessary individual elements work. An example of an element that was expected to work during the development of the workflow but negatively influenced the results in practice, was the percentage adjustment. The idea was that by calculating the percentage a device was matched to a certain rail, the rail with the highest percentage could be assumed to be the correct rail. But based on the analysis of the median calculator, this would actually mean that in this case sensor three four and six would be matched to the wrong rail for a longer amount of time, and the time to be matched to the correct rail would be even longer.

A second characteristic is that the system consists of many different elements that all have to work together and can influence each other both positive and negative. The elements worked together quite well, but an interesting conclusion based on this analysis is that some elements actually do not help each other. In the median calculation, the result gets better and better, closer to the rail. But in the next processor, it can be possible that the result of this median lies just a little closer to the wrong track, which causes it to get matched to the incorrect track, causing the error to actually get larger. This is inherent to the way map matching, which is essential in this case, works, and will only change when more results are taken in to account in the median calculation.

The fact that the tests were done and performed in a way that can be explained by the research, has been very valuable for two reasons, it confirmed the hypothesis, that GPS results can greatly improve by using multiple measurements, and provided a large part of the answer to the research question, but also provided proof that a complete and working system has been developed during this project and supported the business case for CGI, as will be discussed further in the next paragraph.

5.7 PRACTICAL APPLICABILITY

Based on this case study and the development of the system, a couple of things can be said about the practical applicability. This topic is important in this project, as the project was performed for CGI, a company that works together with ProRail on several projects. Not only the academic and technical results of the project are important for CGI, but also the business case.

The first conclusion with respect to the applicability of the project is that the whole process of gathering and processing the data works. Some tweaks might be necessary when actually implementing the system due to the possibility of slightly different sensors or a slightly different system, but overall the system could be implemented as is. The results are satisfactory, a wagon could be found quite easily, and the longer the wagon stays in the same location, the more precise this found location is going to be. The order of the wagons is also a possible outcome of the system, even though some tweaking in the new version of the GeoEvent Extension would be needed to make sure this is outputted in an appropriate format.

That brings us to the second conclusion, when the system would be implemented in a real situation, the newest version of the GeoEvent Extension should definitely be used. This version has more possibilities, can handle more events per second and has in general a better performance. Some investigation would have to be done in the exact set up of servers and software to make sure all data can be handled appropriately, but that would be done by a system engineer.

Another issue to investigate is whether the Undagrid devices would be the best devices to use and whether it would be necessary to use these devices. The main advantage of the Undagrid system is that it uses a wireless mesh network. This would be very good to use if the devices cannot communicate directly with the gateway. In order to see whether a mesh network is necessary, some investigation would have to be done to see whether hops are necessary or whether the devices can communicate directly to the gateway anyway. If hops are necessary, Undagrid could be used, but the company would have to make some adjustments to their software in order to give the necessary data. An accelerometer should be in the devices that measures when the device has stopped, on which moment a timestamp should be added to each of the send packages. When the device has stopped, it should send the location every minute in order to make it possible to use previous measurements.

The battery use of the system should also be investigated. During the tests done with the devices, battery use was not an issue, the batteries that were provided with the demo set did not yield problems. The gateway was powered by a portable power bank, and did not seem to use a lot of power. In the longest test, it had only used about a quarter of the 4000 mAh available. In cooperation with Undagrid or by doing extra tests with another system, the exact battery use would have to be researched in order to see how often the batteries on the wagons would have to be replaced or charged.

The last improvement that could be made is to make a manual DGPS network on the marshalling yard. A GPS sensor should in that case be placed in the gateway, which will be placed on the marshalling yard. This gateway has to be placed on a known location and can be used to calculate corrections to the GPS measurements. These corrections could be used during the processing phase of the data of the other devices. This could improve the accuracy and thereby improve the time the system needs to match the point to the correct rail.

The main conclusion about the applicability of the system is that with some extra research and development effort, the system could very well be implemented, if stakeholders are interested. The potential of the system lies mainly in improving the efficiency of not only the rail transportation, but other types of transportation as well. The system also might be applicable in other cases.

6 CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

The objective of this research was to see whether a low cost system using current localisation techniques could be able to localise freight wagons on marshalling yards. The sub questions were:

1. What techniques could be used for the localisation of freight wagons and what are the advantages and disadvantages?
2. What localisation technique would be the best method to use in this particular case, based on its performance parameters?
3. What would be the best set up to use for the chosen method?
4. Does the method and system architecture work? Does it generate the expected results?
5. On what scale can freight wagons be found using the system?

In this conclusion, the first subparagraph will discuss sub question one to three, and the last paragraph question four and five and the answer to the research question as a whole.

6.1.1 LOCALISATION TECHNIQUES

What techniques could be used for the localisation of freight wagons and what are the advantages and disadvantages?

In a literature study, six different localisation techniques were researched, of which three were appropriate to use on a marshalling yard. The three final techniques were RFID, UWB and GPS/GNSS. The advantage of RFID is that it could result in a specific track for each to the wagon, as one scanner would be placed at the entrance and one at the exit of each track. A disadvantage is the installation time and effort and the costs, as a lot of scanners would be necessary. An advantage of UWB is that it is supposedly quite accurate and precise and is not influenced by multipath in the way Wi-Fi and Bluetooth, which are comparable, are. A disadvantage is that UWB is relatively new and that it would pose a large risk on this project to use a technique that is that new and of which not much is known. An advantage of GPS/GNSS is its availability, as GPS/GNSS localisation is possible anywhere as long as you have receiver. A disadvantage could be the accuracy of the technique, as the rails on the marshalling yard might be too close together to get an appropriate result.

What localisation technique would be optimal to use in this particular case, based on its performance parameters?

After consideration of and consultation with all stakeholders involved, GPS/GNSS was chosen. The reason for this is that GPS/GNSS is easily available and does not involve the risks and set up time that some of the other techniques would have. It also poses an interesting angle for research in this project. As trains are usually stationary on marshalling yards, this project offers a chance to look into the possibility of using previous measurements to improve the resulting localisation of GPS/GNSS.

What would be the best set up to use for the chosen method?

To use this system on freight wagons on marshalling yards, each freight wagon would have to have its own GPS sensor that is affordable and does not use a great amount of battery power. In order to save battery power and money, not each individual device has to have internet connection, only a gateway with which the individual devices connect in order to share the measurement. This connection can either be directly or via other sensors in the network. The gateway can be placed on the marshalling yard. In order to also gather data about the train while driving, a gateway can also be placed in each locomotive. An accelerometer can be used to detect when the train has stopped in order to determine from when previous data has to be used to improve the position.

The GPS/GNSS devices that are used in this project are provided by the company Undagrid, a start-up company that has developed a low cost and low power consuming localisation system that uses GPS/GNSS measurements and communicates those via a mesh network to a gateway and from there to the server.

There are a couple of characteristics of the localisation on a marshalling yard that are really interesting to take into account during the project. The first has already been mentioned and is the fact the wagons are standing still for the largest part of their stay. This characteristic is used to improve the resulting location of the system. This is done by taking the median or the mean of the previous results. This should make the resulting location closer and closer to the real value as more time passes. The advantage of the mean is that it is really easy to use and can pose better results when fewer measurements are available. The advantage of the median is that outliers have less influence on the result. Both methods have been implemented in order to see which one would be best to use in this system.

The second characteristic of the marshalling yard is the fact that the wagons have to be on a certain track. The track can be seen as an extra reference system. Wagons also cannot change tracks when standing still on the marshalling yard. This is the same with the order of the wagons, which will also remain the same. Based on the resulting track the sensors are calculated on, the order of the wagons can also be determined in order to give a description of the location of the wagons which can be understood by all stakeholders, instead of just coordinates.

The GeoEvent processor by Esri is used to develop the system, as this software is especially designed to process data real time and is customable with processors that can be developed using their java SDK. In this GeoEvent processor, the data is extracted from the Undagrid server, and then goes through four custom processors: the average/median processor, the map matching processor, the percentage on rail processor and the order of wagons processor. After this, the data is represented on a map using ArcGIS for server. The location can also be described by the track the train is on and the order of the wagons on that track.

6.1.2 DOES THE SYSTEM WORK (TESTS AND CONCLUSIONS)

After some problems in communication with ProRail and the transporter Locon and the time available, it was unfortunately not possible to test the system on the trains of Locon and by extend not possible to test it on a marshalling yard. To simulate a marshalling yard as good as possible, it was logical to still use the dataset provided by ProRail. To do this, the tests took place on the station of Geldermalsen in the Netherlands, as the edge of the platform also was a line in this dataset similar to the tracks on the marshalling yard. By placing the boxes right next to the platform, a comparable situation was created.

Does the method and system architecture work? Does it generate the expected results?

During the tests, the whole system architecture on the GeoEvent Extension was up and running. At one or two of the tests, the system in the GeoEvent Extension did not work, but these were usually small errors, for example typos, somewhere in the code. At the final test, the system worked as a whole and generated the expected results, being a location, a track number and a distance and bearing towards the other sensors on that track. The last result was created in excel due to limited possibilities of the current GeoEvent Extension.

On what scale can freight wagons be found using the system?

After the tests, all steps of the system were analysed on their performance and their contribution to an accurate localisation of the wagons on the marshalling yards. With the raw data it was possible to just give an area on the marshalling yard on which the wagon would be. Whereas this is already an improvement compared to the current situation, the result could be better.

The average and median calculators were analysed on their performance. The average calculator did improve the location significantly, instead of a scattered cloud of locations, snakelike patterns were the result, that were a lot closer to the actual location than the raw data. But for this test, it was not only the improvement of the average calculator that was important, it was also the fact what would happen if the result of this average were matched to the rail network. After matching the conclusion could be drawn that a minimum of 51 % was matched correctly over the course of the whole test. The result was expected to get better over time, and this was also the case in the tests. After about five quarters of an hour, four of five sensors were always matched to the correct track.

The median calculation was analysed next and yielded even better results. The pattern of the median was quite different, and it was clearly visible that the points became closer and closer together with more results. After the map matching of the median results, it became apparent that at least 65% of the results were correctly matched over the course over the whole test. But the test became really successful once the time aspect was taken into account. After about 70 minutes of testing, the results of all devices were matched to the correct track at all times.

Based on this correctly matched data, it was not difficult to determine the order of the wagons. After the test, device thee and four were switched in comparison to the real situation, but as these two were located practically on the same location, this did not diminish the success of the test, as the devices on two different wagons are never on the same place. Once a couple of meters, in this case ten meters, are in between the sensors, the correct order can always be determined.

6.1.3 FINAL CONCLUSION

To what extent is it possible, using current and affordable positioning techniques, to localise freight trains on marshalling yards in such a way that it can be determined on which shunting track the coach is located and what the order of the wagons is?

To determine the location of the freight wagons, affordable GNSS can be used as this is easily available and is already used in part of the rail industry. Using affordable GNSS will not immediately yield the accuracy necessary to determine the correct track and order of the wagons, but by taking previous measurements into account by means of a median calculation, after an hour and a half all results are matched to the correct track. With this track known, it is possible to determine the order of the wagons.

The system that was developed works and it will be interesting to make it into an appropriate system that can be used on actual freight trains all over the Netherlands. The project also posed some interesting research results, especially with respect to the use of previous measurements and map matching of stationary GNSS measurements.

In this paragraph, some recommendations for future work, on which more research is necessary, will be done.

First of all, the focus of this project was on the performance of the system with respect for localisation, to see whether using previous results would make it possible to use low cost devices to localise trains. The larger scale of the system was not in the scope of this project, but a couple of things should be taken into account when implementing the system on the larger scale. When all wagons in the Netherlands or even in Europe would have devices like this, a lot more data has to be processed than was the case in this test. The GeoEvent Extension, a very useful tool with this type of processes, should be able to handle datasets this big. If tens of thousands of devices send their location every minute, the GeoEvent Extension should be able to process this. The history calls that are made to find the previous results should also be possible, which would probably need large servers with a lot of computation power. These are all aspects of the larger scale of the project that should be investigated when the system would be actually implemented.

Some work should also be done to find whether the system that was used during this test, the system of Undagrid, is in fact an appropriate system to use. The power of this system lies in the elaborate mesh network the devices set up between them, but it would be the question whether this mesh network would actually be needed. An investigation could be done towards the range of a gateway to see if the placement of a couple of gateways on a marshalling yard could actually be enough to gather all data. To investigate this, more sensors would be needed than five, to see whether they use the mesh network or not. If the mesh network is not necessary, it might be better to set up a custom system or use another supplier. When developing the system to be implemented, some investigation should also be done to see whether the Undagrid system should be used at all, or whether another supplier or custom system could also be used in the case that a mesh network is necessary.

During the development phase, the system of Undagrid posed some problems. As explained in paragraph 4.1, the standard settings of the devices are to determine the position only once when the device came to a halt, according to the accelerometer. This is not ideal when using multiple measurements on a marshalling yard. However, by changing this to calculate the location every minute while stationary, the time on battery will decrease a lot, as will be discussed in the next point.

Another interesting case to develop further is an investigation of the battery use of the system. During this project, the battery use was outside the scope, so no real conclusion can be drawn about the battery use. This is a very interesting investigation because it impacts the applicability greatly. If the battery use is good, it will be possible to use this system on the trains, as the transporters will not like to change or charge the batteries very often. An aspect to investigate in with respect to this issue is whether a condition similar to the original condition of Undagrid can be set. If after for example three hours, the resulting location will always be on the correct track and in the correct order, it will not be necessary to determine the location every minute anymore, but less often or not at all. The time limit for this condition would have to be investigated by doing more tests on a larger scale.

The last recommendation for future work has to do with the method of positioning. During this project, just a low cost GPS system was used on its own, with which it was possible to determine the track the devices were on. In order to get to this result faster, it is worth investigating the influence of setting up a custom DGPS network on the marshalling yard. In order to achieve this, the gateway would have to be equipped with a GPS sensor as well and put on a known location. The errors in satellite positioning can then be found using these known GPS locations, and can be used in further processing to determine the location of the wagons even faster and more precise.

7 BIBLIOGRAPHY

Aiello, R., & Batra, A. (2006). *Ultra Wideband Systems.* Oxford: Elsevier Inc.

AWF-Wiki. (2011, May 30). *Global Positioning System (GPS)*. Retrieved from AWF wiki (Forest Inventory Remote Sensing): http://wiki.awf.forst.uni-goettingen.de/wiki/index.php/Global_Positioning_System_(GPS)

Bakker, P. d., Knoop, V., Tiberius, C., & Arem, B. v. (2012). *Mapping Motorway Lanes and Real-Time Lane Identification with Single-Frequency Precise Point Positioning.* Delft: Delft University of Technology.

Belinter Logistic. (2012). *Belinter logistics international transportation*. Retrieved from Belinter: http://www.bil.by/en/info/useful/railwaywagons/

Benyamina, A., Hafid, A., & Gendreau, M. (2012). *Wireless mesh networks design- a survey.* IEEE Communciations Surveys & Tutorials.

Berens, F. (2009, October 6). The European UWB Regulation and Standardisation. Wasserbilig, Luxembourg: FBConsulting S.a.r.l.

Berger, T., Denny, J., Robins, D., Koop, L., Payne, E., & Twitchell, R. (2012). *Patent No. US8279067.* United States of America.

Bernstein, D., & Kornhauser, A. (1996). *An introduction to map-matching for personal navigation assistants.*

Blewitt, G. (1997). *Basics of the GPS Technique: Observation Equations.* Newcastle upon Tyne: Swedish Land Servey.

Bluetooth. (2014). *Bluetooth Basics*. Retrieved from Bluetooth: http://www.bluetooth.com/Pages/Basics.aspx

Boon, L. (2012, March 03). *RFID introduction*. Retrieved from NFC (Near Field Communication), RFID: http://www.siongboon.com/projects/2012-03-03_rfid/index.html

Calais, E. (Unknown). GPS Observables. West Lafayette, Indiana, United States of America: Purdue University.

California Institute of Technology. (2013, November 14). *Integrity & Performance Monitoring*. Retrieved from NASA; Jet Propulsion Laboratory: http://www.gdgps.net/products/monitoring.html

Carlsen, S., & Petersen, S. (2010). *When the Industry Goes Wireless: Drivers, Requirements, Technology and Future Trends.* Trondheim, Norway: Intech.

Collins, G. (2004). *The foundations of celestial machines.* NASA Astrophysics Data System.

de Bakker, P., Knoop, V., van Arem, B., & Tiberius, C. (2013). *Mapping motorway lanes and real-time lane identification with single-frequency precise point positioning.* Delft: TU Delft.

Devlin, K. (1993). *The Joy of Sets.* New York: Springer-Verlag.

Dixon, W. (1953). Processing Data for Outliers. *Biometrics*, 74-89.

Djebouri, J. (2014). Avaraging Correlation for Fast GPS Satellite Signal Acquisition in Multipath Rayleigh Fading Channels. *Microwave Journal | Frequency Matters*.

Dobkin, D. (2012, December). *The RF in RFID*. Retrieved from Enigmatics: http://www.enigmatic-consulting.com/Communications_articles/RFID/Link_budgets.html

Dreamstime. (Unknown). *Map magnifier*. Retrieved from Dreamstime: http://www.dreamstime.com/stock-images-map-magnifier-image259045(74)

Esri. (2014). *ArcGIS GeoEvenProcessor Developer Guide.* ESRI.

Esri. (2014). *ArcGIS GeoEvent Extension for Server*. Retrieved 05 01, 2015, from ESRI: http://www.esri.com/software/arcgis/arcgisserver/extensions/geoevent-extension

Esri. (2015). *ArcGIS REST API*. Retrieved May 9, 2015, from ArcGIS Resources: http://resources.arcgis.com/en/help/arcgis-rest-api/index.html#//02r3000000sr000000

European Space Agency. (2014, September 18). *GNSS Performances - Navipedia*. Retrieved from eesa nauipedia: http://www.navipedia.net/index.php/GNSS_Performances#cite_note-0

Fuhr, P. (2008, August 1). *Asset Tracking*. Retrieved from Sensors Online: http://www.sensorsmag.com/networking-communications/asset-tracking/asset-tracking-industrial-settings%E2%80%94a-review-wireless-technol-1500

Garcia Castano, J., Svensson, M., & Ekstrom, M. (2004). *Local Positioning for Wireless Sensors Based on Bluetooth.* Västeras: Malardalen University.

Gerkis, A. (2006). *A survey of wireless mesh networking security technology and threats.* Swansea: SANS Institute.

Glover, B., & Bhatt, H. (2006). *RFID Essentials.* USA: O'Reilly Media, Inc.

Google Maps. (2014). *Google Maps*. Retrieved from Google: https://www.google.nl/maps/@51.8326487,4.5958577,18z

Groundspeak. (2015). *1.6. Calculating Distance Between Two Sets of Coordinates*. Retrieved 5 2, 2015, from Groundspeak: https://support.groundspeak.com/index.php?pg=kb.page&id=211

Hagen, J. (2014, December 2-4). ArcGIS GeoEvent Processor for Server. *Southwest User Conference Esri*. Santa Fe, New Mexico, United States of America: Esri Solutions.

Haughey, D. (2011). *MoSCoW Method*. Retrieved from Project Smart: http://www.projectsmart.co.uk/moscow-method.php

Heng, L., XingXin Gao, G., Walter, T., & Enge, P. (2011). Innovation: Digging into GPS Integrity. *GPS World*.

Hennigens, R. (2012). Current approaches of WiFi positioning. *Service-centric Networking seminar.* Berlin: TU-Berlin.

Henry, J.-P. (2014). 06-GPS : The Most Versatile Network. 06-GPS.

Herrmann, R. (No Date). *Definition of basic terms and properties; UWB at a glance*. Retrieved from UKoLoS: http://www-emt.tu-ilmenau.de/ukolos/tecfacts_1.php

Hofmann-Wellenhof, B., Lichtenegger, H., & Wasle, E. (2008). *GNSS; Global Navigation Satellite Systems; GPS, GLONASS, Galileo & more.* Wien, Austria: Springer-Verlag.

Holtkamp, E. (2014, May 4). Goederenwagons. Retrieved from https://www.flickr.com/photos/63566785@N05/14088992799

Hunt, D., Puglia, A., & Puglia, M. (2007). *RFID A Guide to Radio Frequency Identification.* Hoboken: John Wiley & Sons, Inc.

Inspectie van Verkeer en Waterstaat. (2011). *Weten wat er staat.* Den Haag: Inspectie Verkeer en Waterstaat.

Japan Aerospace Exploration Agency. (2003). *Positioning to know your location and time*. Retrieved from JAXA: http://global.jaxa.jp/countdown/f18/overview/gps_e.html

Kamer van Koophandel. (2014). *Vervoerssoorten*. Retrieved from Ondernemersplein: https://www.ondernemersplein.nl/ondernemen/internationaal-ondernemen/internationaal-transport/vervoerssoorten/

Kane, L., Verma, B., & Jain, S. (2008). Vehicle tracking in public transport domain and associated spatio-temporal query processing. *Computer Communications*, 2862-2869.

Kouba, J. (2003). *A Guide to Using International GPS Service (IGS) Products.* Ottowa, Ontario: Natural Resources Canada.

Langley, R. (1990). Diltution of Precision. *GPS WORLD*, 52-59.

Lloret, J., Tomas, J., Canovas, A., & Bellver, I. (2011). *GeoWiFi: A Geopositioning System Based on WiFi Networks.* Valencia, Spain: Universidad Politecnica de Valencia.

Martin, E., Liu, L., Covington, M., Pesti, P., & Weber, M. (2010). Positioning Technologies in Location-Based Services. In S. Ahson, & M. Ilyas, *Location-Based Services Handbook* (pp. 1-45). CRC Press.

Max. (2014, June 07). *Tag RFID - 001*. Retrieved from EmmeShop electronics: http://www.emmeshop.eu/blog/?q=node/19

Minesterie van Defensie. (2013, 09 20). *PCTrans*. Retrieved from Ministerie van Defensie: https://www.defensie.nl/onderwerpen/hydrografie/inhoud/nautische-producten/nautische-computerprogramma%E2%80%99s/pctrans

Nagabhushan, T., Shiva Prakash, S., & Kinkin, K. (2013). Power-saving routing algorithms in wireless mesh networks: a survey. *Converence of FRUCT association.* St Petersburg: FRUCT association.

National Coordination Office for Space-Based Positioning, Navigation and Timing. (2014, October 30). *Space Segment*. Retrieved from GPS.gov: http://www.gps.gov/systems/gps/space/

National Instruments. (2014, April 9). *Bluetooth*. Retrieved from National Instruments: http://www.ni.com/white-paper/7104/en/

NOS. (2013). *150 wagons kwijt in Frankrijk*. Retrieved from NOS: http://nos.nl/artikel/489053-150-wagons-kwijt-in-frankrijk.html

NOS.nl. (2015, April 29). *Goederenvervoer over het spoor minder populair*. Retrieved April 29, 2015, from NOS.nl: http://nos.nl/artikel/2033072-goederenvervoer-over-het-spoor-minder-populair.html

Ogaja, C. A. (2011). *Applied GPS for Engineers and Project Managers.* Virginia: American Society of Civil Engineers.

Oliveti, M., Mulder, D., Abdhilakh Missier, G., Zervakis, D., & Wu, H. (2014). *Creating a smartphone application to locate colleagues.* Delft: Delft University of Technology.

Omni-ID. (2008). *The Technology of On-Metal RFID.* Foster City: Omni-ID Cayman, ltd.

Penrose, R. (2004). *The Road to Reality: A Complete Guide to the Laws of the Universe.* Vintage Books.

Pronk, M., Theunisse, I., Jonker, K., Chen, C., & Willems, H. (2014). *Geofort; the realisation of a marauders map at the Geofort GeoExperience.* Delft: Delft University of Technology.

ProRail. (Unknown). Basisbeheerkaart (BBK). *Inframanagement, Infra Informatie, Basisbeheerkaart*. Prorail.

Quddus, M., Ochieng, W., & Noland, R. (2007). Current map-matching algorithms for transport applications: State-of-the-art and future research directions. *Transportation Research*, 321-328.

RFID product. (2008). *Contactless cards*. Retrieved from RFID product: http://www.rfid-product.com/

Rote, G. (2010, 7 15). *Point Matching*. Retrieved 5 4, 2015, from Ipelets: http://www.inf.fu-berlin.de/inst/ag-ti/people/rote/Software/ipelets.html

Rousseuw, P., & Croux, C. (2012). Alternatives to the Median Absolute Devioation. *Journal of the American Statistical Association*, 1273-1283.

Sahinoglu, Z., Gezici, S., & Guvenc, I. (2008). *Ultra-wideband positioning systems.* Cambridge: Cambridge University Press.

Samuel, S. (2011). *Handboek Railgoederenvervoer.* Utrecht: Prorail.

Sassano, M. (2012, July 31). *Network Infrastructure: Where to Build?* Retrieved from Sunlight; Shedding light on infrastructure solutions: http://sunlight.sunesys.com/2012/07/

Schoubben, M., & Barzeele, P. (2011, September 3). *Rurtalbahn met een autotrein*. Retrieved from Spoorwegfotografie: http://spoorwegnieuws.blogspot.nl/2011/09/rurtalbahn-met-een-autotrein-op-lijn-34.html

Schreutelkamp, F., & Strang van Hees, G. (Unknown). *Benaderingsformules voor de transformatie tussen RD- en WGS84-kaartcoordinaten.* Delft: Stichting 'de Koepel'en TU Delft.

Sheppard, W., & Soule, C. (2008). *Practical navigation.* New York: World Technical Institute.

Spoor pro. (2014, September 9). *Rangeerterrein Kijfhoek zaterdag open voor publiek*. Retrieved from Wpoorpro vakblad voor railbedrijven: http://www.spoorpro.nl/goederenvervoer/2014/09/09/rangeerterrein-kijfhoek-zaterdag-open-voor-publiek/

Tenpierik, M. (2014). *Handleiding aanleveren beladingsgegevens.* Rotterdam: Prorail.

Terrier, S. (2005). Differential GPs principles. *Journal of NeuroEngeneering and Rehabilitations*.

Thomas, R. (2013, June 20). *Conic section hide and seek*. Retrieved from Plus magazine: http://plus.maths.org/content/conic-section-hide-seek

Tweakers. (2014). *Toshiba Tecra S11*. Retrieved from Tweakers: http://tweakers.net/pricewatch/265967/toshiba-tecra-s11-13m/specificaties/

Universal Avionics. (2013). *Understanding Required Navigation Performance (RNP) and Area Navigation (RNAV) Operations.* Tuscon, Arizona: Unversal Avionics.

van Bree, R., Buist, P., Tiberius, C., & van Arem, B. (2011). *Lane identification with real time single frequency precise point positioning: a kinematic trial.* Delft: TU Delft.

Warner, J., & Johnston, R. (2003). *GPS Spoofing Countermeasures.* New Mexico: Los Alamos National Laboratory.

What When How. (Unknown). *GPS Details*. Retrieved from what-when-how | In Depth Tutorials and Information: http://what-when-how.com/gps/gps-details/

WiFi Versterker. (2014, September 27). *Hoe wifi signaal versterken?* Retrieved from WiFi versterker: http://www.wifiversterker24.nl/tag/wifi-versterken/

Wikipedia. (2014, December 2). *Dilution of precision*. Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Dilution_of_precision_(GPS)

Wikipedia. (2014, June 12). *Kijfhoek (rangeerterrein) - Wikipedia*. Retrieved from Wikipedia: http://nl.wikipedia.org/wiki/Kijfhoek_%28rangeerterrein%29

Yadav, A., & Sharma, A. (2014). Near Field Communication; An Era of Smart Touch. *Journal of The International Association of Advanced Technology and Science*.

Yaqub, M., & Shaikh, U. (2012). *Near Field Communication.* Dhahran: King Fahd University of Petroleum and Minerals.

z24. (2013, November 6). *Schade ontspoorde goederentrein Borne bedraagt zeker 1 miljoen euro*. Retrieved from Z24 nieuws: http://www.z24.nl/geld/schade-ontspoorde-goederentrein-borne-bedraagt-zeker-1-miljoen-euro-400092

Zhang, Y., Li, L., & Zhang, Y. (2009). *Research and Design of Location Tracking System Used in Underground Mine Based on WiFi technology.* Handan: Heibei University.

Zwich, T., Wiesbeck, W., Timmerman, J., & Adamiuk, G. (2013). *Ultra-wideband RF System Engineering.* Cambridge: Cambridge University Press.

A. PLANS TO TEST THE SYSTEM

TESTPLAN 1

Brief aan Locon en notities Wim:

Beste Toon Habers,

Als afstudeerproject voor de TU Delft in samenwerking met het bedrijf CGI, ben ik onderzoek aan het doen naar het ‘vinden’ van treinen op rangeerterreinen door middel van GPS. Hierover heb ik contact gehad met Wim Verheul van Prorail die mij vertelde dat er wellicht ook een test gedaan gaat worden met RFID op jullie treinen. Het systeem dat ik aan het ontwikkelen ben maakt geen gebruik van RFID, maar van GPS om de locatie van de wagons en locs te bepalen, dit zijn dus tests voor een andere methode. Wim vertelde mij ook dat jullie voornamelijk goederen vervoeren tussen Vlissingen en Moerdijk, elke dag dezelfde route. Dit zou voor het testen van mijn systeem ideaal zijn.

Hierdoor vroeg ik me af of het mogelijk zou zijn om het systeem dat ik ontwikkel voor mijn afstudeerproject te testen op uw treinen? Dit houdt in dat er het liefst tussen 20 en 27 april voor ongeveer een week kleine GPS kastjes op de wagons worden vastgemaakt, foto’s hiervan zijn bijgevoegd. Hier hoeft u verder niks mee te doen, deze zitten er alleen. In de locomotief of op het rangeerterrein wordt dan een gateway geplaatst, die de informatie uit de kastjes zal versturen, ook hiervan is een foto bijgevoegd. Op elke wagon worden één of twee kastjes geplaatst, dit kan ook op alleen een paar wagons, maar er geldt hoe meer kastjes ik kan plaatsen, hoe beter ik het systeem kan testen. De kastjes zullen bevestigd worden aan de buitenkant van de wagons, door middel van een combinatie van duct tape en tie wraps, waardoor ze goed vast zullen zitten. Aan het eind van de week zal ik de kastjes weer van de wagons afhalen. Het zou het handigst voor mij zijn als de plaatsing en het weghalen van de kastjes op het rangeerterrein in Moerdijk kan gebeuren.

De units die geplaatst worden hebben wij te leen van de firma Undagrid BV, deze zullen dan ook eigendom van deze firma blijven.

Als u nog vragen hebt, schroom dan niet om die te vragen, alles moet uiteraard zo duidelijk mogelijk zijn.

Vriendelijke groet en alvast bedankt,

Hester Willems

Plan

Tussen 13 en 24 april 5 dagen testen op een trein (die rijdt dan 5 keer heen en weer).

Op maandag: naar rangeerterrein om de gateway op het rangeerterrein te zetten en devices op de trein te plakken.

Devices bevestigen dmv: sterke magneet of sterke tape (Thomas om hulp vragen voor inspiratie)

Gateway aanzetten en een week lang testen.

Controleren of Gateway het blijft doen > Undagrid server

Aan het eind van de week (vrijdag) weer naar het rangeerterrein om de gateway van het terrein af te halen en devices van de trein.

Regelen:

- Gateway op rangeerterrein
- Eventueel accu vinden als stroomvoorziening niet beschikbaar is
- Ikzelf toegang tot rangeerterrein, ook tijdens de week als er iets mis is
- Manier van bevestigen duidelijk krijgen
- Rijschema van Locon krijgen
- Netwerk van rangeerterreinen duidelijk krijgen

TESTPLAN 2

Tussen 13 en 24 april een dag of twee dagen testen op station

Start: Undagrid boxen dichtbij rand van het perron leggen en gateway aansluiten. Zorgen dat alle stappen van de GeoEvent extension klaar staan om de juiste data te ontvangen

Testen: Undagrid boxen een bepaalde tijd stil laten liggen om data te verzamelen

Eind: Undagrid devices en gateway pakken

Regelen:

- Eventueel accu vinden als stroomvoorziening niet beschikbaar is
- Netwerk op station regelen

B. REFLECTION

In this thesis, a method and system to find freight wagons on marshalling yards was researched and developed. The research was performed according to the intended planning and took eight months. The research took place at the IT company CGI located in Rotterdam. The fact that the research took place at a company meant that more stakeholders were involved than in a regular graduation project. Sometimes this was difficult to manage, as a balance had to be found between the practical applicability and the research aspect of the project. This was a challenge, but the added value of performing the graduation project at a company was also large, resources and motivation from the employees was always available.

In the master Geomatics for the Built Environment, the geo-information chain is a methodical line that is often followed. This chain is visualised in Figure 57. In this project, all steps of the geo-information chain were implemented. The globe represents the real world, the actual situation. In this project, the actual situation consisted of freight wagons devices on a marshalling yard. Data capture in this project was done using GPS/GNSS to get measurements of the location of the freight wagons.

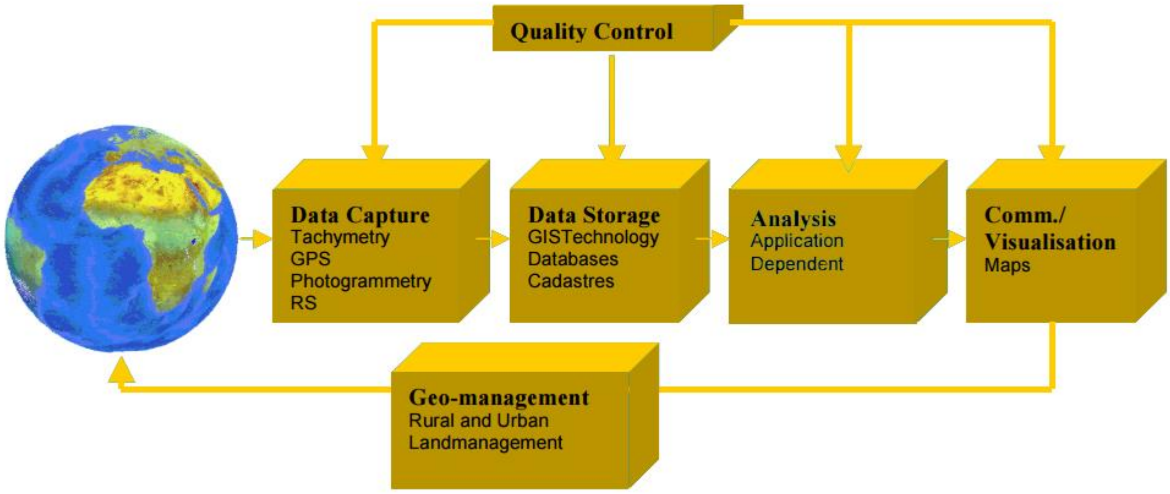


FIGURE 57 GEOINFORMATION CHAIN (LEMMENS, 2013)

Data storage was done in several ways, the first step was in the database of Undagrid, where a REST endpoint could be queried to extract the result. During the processing of the data, feature services from ESRI were used to store intermediate data. The analysis that was done consisted of several processes that were invoked on the data, including median calculations and map matching algorithms. Another part of the analysis was determining whether the algorithms and map matching had effect on the data and how this would influence the accuracy of the result. This was done using visualisations and comparing the known distances to the distances that were calculated based on the data that was received and processed. Another step in the analysis was to see how the time aspect of the stationary vehicles was visible in the track to which they were matched over time. Visualisation was done using ArcMap, ArcGIS online and the operations dashboard. All aspects of the geo-information chain were implemented, following the methodical line of the Master Geomatics for the Built Environment.

Part of the research and application field of Geomatics is research towards tracking, tracing and localisation techniques. This project adds to that field of research, by investigating stationary GPS and methods to improve the generated location. Another interesting aspect of this research was the rail network that played an important role in determining the actual location. These two aspects combined result in a model were the location of the train will get better and better the more time passes and can be communicated between the stakeholders as a location, not a position.

The project fits in the wider social context by providing a more efficient ways of transporting goods. The transportation of goods is very important for the economy of any country. In the Netherlands, one of the main points in exporting and importing goods is the Rotterdam harbour. From this harbour, goods are transported all over Europe. With the environment and traffic congestions being an ever increasing problem, an alternative for transportation over roads is needed. Transportation using rails is a very good alternative, but is actually used less and less because of inefficiency in the process. This inefficiency is partly caused by the fact that trains and wagons can get lost for easily a week. The system that was researched and developed during this project, solves this problem and provides a way to always find the wagons when they are on a marshalling yard.

C. SOURCE CODES GEOEVENT PROCESSOR

B1 INPUT TRANSPORT UNDAGRID DATA

As this input transporter is based on existing code, only a few parameters were changed and the code was not written by me. I will still show it here as it is very relevant and will note which part was filled in for this particular case.

```
package com.cgi.gep.tomtom;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import sun.misc.BASE64Encoder;
public class TomTomHttpStrategy {

      public String getUndagridData(String username, String password)
        throws IOException
      {
        long date = System.currentTimeMillis() / 1000;

       String start = String.valueOf(date - (60));// STARTTIME AND ENDTIME DEFINED
        String end = String.valueOf(date); // ENDTIME = CURRENT TIME, STARTTIME = 60 SEC EARLIER

        String url = "https://api.undagrid.com:443/v1/channels/8f8f9600-d93a-11e4-b958-02c2e988c447/"
                + "history?from=" + start + "&to=" + end + ""; // UNDAGRID CHANNEL

        URL obj = new URL(url);
        HttpURLConnection con = (HttpURLConnection)obj.openConnection();
        BASE64Encoder enc = new BASE64Encoder();
        String userpassword = username + ":" + password;
        String encodedAuthorization = enc.encode(userpassword.getBytes());
        con.setRequestProperty("Authorization", "Basic " + encodedAuthorization);
        con.setRequestMethod("GET");
        con.setRequestProperty("User-Agent", "Mozilla/5.0");
        int responseCode = con.getResponseCode();
        BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
        StringBuffer response = new StringBuffer();
        String inputLine;
        while ((inputLine = in.readLine()) != null) {
          response.append(inputLine);
        }
        in.close();
        return response.toString();
      }
    }
```

B2 AVERAGE OR MEDIAN PROCESSOR

PROCESSOR

```java
package com.cgi.processor.undagrid;

 * #%L

import java.util.ArrayList;
import java.util.Collections;
import com.esri.core.geometry.Point;
import com.esri.ges.core.component.ComponentException;
import com.esri.ges.core.geoevent.GeoEvent;
import com.esri.ges.core.validation.ValidationException;
import com.esri.ges.manager.geoeventdefinition.GeoEventDefinitionManager;
import com.esri.ges.processor.GeoEventProcessorBase;
import com.esri.ges.processor.GeoEventProcessorDefinition;


public class UndagridGPSAvarageProcessor extends GeoEventProcessorBase { // START OF PROCESSOR
    // Spatial spatial;
    GeoEventDefinitionManager manager;
        public UndagridGPSAvarageProcessor(GeoEventProcessorDefinition definition,
            GeoEventDefinitionManager m) throws ComponentException {
        super(definition);
        // spatial = s;
        geoEventMutator = true;
        manager = m;
    }

    public void afterPropertiesSet() {

    public synchronized void validate() throws ValidationException {

    @Override
    public GeoEvent process(GeoEvent ge) throws Exception { // START OF THE ACTUAL OPERATION PHASE OF THE PROCESSOR
        // Operation phase...

        double x_old = (Double) ge.getField("lon"); // IMPORT OF THE X, Y AND NAME FROM THE PREVIOUS STEPS IN THE GEOEVENT PROCESSOR
        double y_old = (Double) ge.getField("lat");
        String name = (String) ge.getField("sensorId");

        Connector getData = new Connector();//SET UP A CONNECTOR TO GET THE PREVIOUS DATA FROM THE SERVER (WHERE IT WAS SAVED IN THE GEOEVENT SERVICE)

        ArrayList<Point> points = getData.getLastUpdate(name, x_old, y_old); // EXECUTE THE METHOD TO GET THE PREVIOUS POINTS


        if (points == null){ // IF STATEMENT IN CASE NOTHING IS RETURNED (WHEN SERVER IS STILL EMPTY AT START)
            return ge;
        }

        //Point point = getAvarage(points); // AVERAGE CALCULATION METHOD IS EXECUTED (CAN BE SWITCHED ON INSTEAD OF MEDIAN)

        Point point = getMedian(points); // MEDIAN CALCULATION METHOD IS EXECUTED

        double x = point.getX(); // RESULTING X AND Y ARE COMPUTED
        double y = point.getY();

        ge.setField("lon", x);// RESULTING X AND Y ARE ADDED TO THE GEOEVENT SERVICE
        ge.setField("lat", y);

        return ge;
        }

    @Override
    public void shutdown() {
        // Destruction Phase
        super.shutdown();
    }

    @Override
    public boolean isGeoEventMutator() {
        return true;
    }

    public void onServiceStart() {
        // Service Start Phase
    }

    public void onServiceStop() {
        // Service Stop Phase
    }
    public Point getAvarage(ArrayList<Point> points){ // BASIC AVERAGE CALCULATION METHOD
        double total_x = 0;
        double total_y = 0;
        for (int i=0;i<points.size(); i++){ // X'S AND Y'S ARE ADDED
            double x = points.get(i).getX();
            total_x = total_x + x;
            double y = points.get(i).getY();
            total_y = total_y + y;
        }

        double AvX = total_x / (points.size()); // X'S AND Y'S ARE DIVIDED BY THE AMOUNT OF PREVIOUS RESULTS
        double AvY = total_y / (points.size());
        Point FinPoint = new Point(AvX, AvY);
        return FinPoint;
```

```
    }
    private Point getMedian(ArrayList<Point> points) { // MEDIAN METHOD

        ArrayList<Double> xpoints= new ArrayList<Double>(); // EMPTY ARRAYLISTS ARE CREATED
        ArrayList<Double> ypoints= new ArrayList<Double>();

        for (int i=0; i < points.size(); i ++){
            xpoints.add(points.get(i).getX()); // X'S AND Y'S ARE ADDED TO THE EMPTY ARRAYLISTS
            ypoints.add(points.get(i).getY());
        }
        Collections.sort(xpoints); // THE ARRAYLISTS ARE SORTED FROM SMALLEST TO LARGEST
        Collections.sort(ypoints);

        double medianX = getMiddle(xpoints); // METHOD TO CALCULATE THE MIDDLE X AND Y VALUE IS EXECUTED
        double medianY = getMiddle(ypoints);

        Point medianPoint = new Point(medianX, medianY); // RESULTING POINT IS CREATED
        return medianPoint;
    }
    private double getMiddle(ArrayList<Double> points){ // METHOD TO CALCULATE THE MIDDLE VALUE
        double result = 0;
        if (points.size()%2 == 0){ // IN CASE THE AMOUNT OF POINTS IS EVEN
            int i = points.size()/2;
            result = (points.get(i)+points.get(i-1))/2;

        }
        else { // IN CASE THE AMOUNT OF POINTS IS NOT EVEN
            int i = points.size()/2;
            result = points.get(i);
        }
        return result;
    }

}
```

CONNECTOR

```
package com.cgi.processor.undagrid;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.esri.core.geometry.Point;

public class Connector {// INITIATING CONNECTOR CLASS

    private final static String USER_AGENT = "Mozilla/5.0";

    public Connector() {
    }
    public ArrayList<Point> getLastUpdate(String nodeName, double x , double y) throws IOException,JSONException {// STARTING METHOD

        String url = "http://cgi.godenzonenvanatlas.nl/arcgis/rest/services/HesterTrainTracking/PreviousResults/FeatureServer/0/query?where=sensorId+%3D+%27"
        +nodeName +
        "%27&objectIds=&time=&geometry=&geometryType=esriGeometryEnvelope&inSR=&spatialRel=esriSpatialRelIntersects&relationParam=&outFields=*&returnGeometry"
        + "=true&maxAllowableOffset=&geometryPrecision=&outSR=&gdbVersion=&returnDistinctValues=false&returnIdsOnly=false&returnCountOnly=false&orderByFields="
        + "&groupByFieldsForStatistics=&outStatistics=&returnZ=false&returnM=false&f=pjson";
        // REQUEST IS IMPORTED WITH NODENAME AS VARIABLE
        String response = getData(url);// METHOD TO GET DATA FROM THE REQUEST URL

        ArrayList<Point> points = getPoints(response); // GET THE POINTS FROM THE IMPORTANT JSON STRING

        return points;
    }

    public String getData(String url) throws IOException { // DATA IS IMPORTANT AND RETURNED AS A JSON STRING

        URL obj = new URL(url);
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();

        // optional default is GET
        con.setRequestMethod("GET");

        // add request header
        con.setRequestProperty("User-Agent", USER_AGENT);
```

```java
        BufferedReader in = new BufferedReader(new InputStreamReader(
                con.getInputStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();

        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();

        return response.toString();

    }

    public ArrayList<Point> getPoints(String response) throws JSONException{ // POINTS ARE IMPORTED FROM THE JSON FILE
        ArrayList<Point> points = new ArrayList <Point>();
        JSONObject data_input = new JSONObject(response);
        JSONArray features = data_input.getJSONArray("features");
        System.out.println("LENGTH OBJECT ARRAY: "+ features.length());
        for (int i=0; i<features.length(); i++){
            if (features.getJSONObject(i).has("attributes")) {
                if (features.getJSONObject(i).getJSONObject("attributes").has("lon") && features.getJSONObject(i).getJSONObject("attributes").has("lat")){
                    double x = features.getJSONObject(i).getJSONObject("attributes").getDouble("lon");
                    double y = features.getJSONObject(i).getJSONObject("attributes").getDouble("lat");
                    Point point = new Point(x , y);
                    points.add(point);
                }
            }
        }
        return points;
    }
```

B3 MAP MATCHING PROCESSOR

```java
package com.cgi.processor.undagrid;

 * #%L

import com.esri.core.geometry.GeometryEngine;
import com.esri.core.geometry.Point;
import com.esri.core.geometry.Polyline;
import com.esri.core.geometry.Proximity2DResult;
import com.esri.ges.core.component.ComponentException;
import com.esri.ges.core.geoevent.GeoEvent;
import com.esri.ges.core.validation.ValidationException;
import com.esri.ges.manager.geoeventdefinition.GeoEventDefinitionManager;
import com.esri.ges.processor.GeoEventProcessorBase;
import com.esri.ges.processor.GeoEventProcessorDefinition;

public class SnapToNetworkProcessor extends GeoEventProcessorBase { // PROCESSOR INITIATED
    // Spatial spatial;
    GeoEventDefinitionManager manager;
        public SnapToNetworkProcessor(GeoEventProcessorDefinition definition,
            GeoEventDefinitionManager m) throws ComponentException {
        super(definition);
        // spatial = s;
        geoEventMutator = true;
        manager = m;
    }

    @Override
    public void afterPropertiesSet() {
        // Initialization Phase ...

    }

    @Override
    public synchronized void validate() throws ValidationException {

    }

    @Override
    public GeoEvent process(GeoEvent ge) throws Exception { // PROCESSOR OPERATION PHASE STARTED
        // Operation phase...
        double x = (Double) ge.getField("lon");// X AND Y IMPORTED FROM EARLIER STEPS IN THE GEOEVENT EXTENSION
        double y = (Double) ge.getField("lat");
        Point point = new Point(x,y); // POINT CREATED FROM THE X AND Y
        Connector getData = new Connector(); // CONNECTER INITIATED TO ENABLE EXTRACTION OF BUFFERS AND LINES FROM SERVER
        int ID = getData.getLastUpdateBuffers(x,y);// THE ID OF THE BUFFER THAT THE POINT IS IN IS RETURNED

        if (ID ==-1){// IF THE ID IS MINUS ONE, THE POINT WAS NOT IN A BUFFER AND THE GEOEVENT WILL BE RETURNED WITHOUT SNAPPING
```

```
            ge.setField("trackId", ID);
            return ge;
        }

        Polyline trackline = getData.getLastUpdateLine(ID); // THE APPROPRIATE LINE IS EXTRACTED FROM THE SERVER
        Proximity2DResult new_point = GeometryEngine.getNearestCoordinate(trackline, point, false);//THE ESRI METHOD TO GET THE SNAPPED POINT IS EXECUTED
        Point snappedPoint = new_point.getCoordinate();

        double xnew = snappedPoint.getX(); // THE X AND Y ARE EXTRACTED FROM THE POINT
        double ynew = snappedPoint.getY();

        ge.setField("lon", xnew); // X, Y AND TRACK NUMBER ARE ADDED TO THE GEOEVENT
        ge.setField("lat", ynew);
        ge.setField("trackId", ID);

        return ge;
        }


    public class Dist{ // CLASS OF THE DISTANCE AND A POINT IS CREATED
        private double distance;
        private Point point;

        public double getDistance(){
            return distance;
        }
        public void setDistance(double d){
            this.distance = d;
        }
        public Point getPoint (){
            return point;
        }
        public void setPoint(Point p){
            this.point = p;
        }
    }

    @Override
    public void shutdown() {
        // Destruction Phase
        super.shutdown();
    }

    @Override
    public boolean isGeoEventMutator() {

          return true;
      }

      public void onServiceStart() {
          // Service Start Phase
      }

      public void onServiceStop() {
          // Service Stop Phase
      }

}
```

CONNECTOR

```
package com.cgi.processor.undagrid;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.esri.core.geometry.Point;

public class Connector {// INITIATING CONNECTOR CLASS

    private final static String USER_AGENT = "Mozilla/5.0";

    public Connector() {
    }

    public int getLastUpdateBuffers(double x, double y) throws IOException, JSONException {// STARTING METHOD TO GET THE BUFFER IDS

        String url = "http://cgi.godenzonenvanatlas.nl/arcgis/rest/services/HesterTrainTracking/SnapTestBuffersFS/FeatureServer/0/query?where=1%3D1&"
                + "objectIds=&time=&geometry=%7B\"x\"+%3A+%27" + x + "%27%2C+\"y\"+%3A+%27" + y + "%27%2C+\"spatialReference\"+%3A+%7B4326%7D%7D&geo"
                        + "metryType=esriGeometryPoint&inSR=&spatialRel=esriSpatialRelIntersects&relationParam=&outFields=*&returnGeometry=true&max"
                        + "AllowableOffset=&geometryPrecision=&outSR=&gdbVersion=&returnDistinctValues=false&returnIdsOnly=false&returnCountOnly=false"
                        + "&orderByFields=&groupByFieldsForStatistics=&outStatistics=&returnZ=false&returnM=false&f=pjson";
        // REQUEST IS IMPORTED
        String response = getData(url);// METHOD TO GET DATA FROM THE REQUEST URL

        int ID = getPolygon(response); // GETS THE IDS OF THE DIFFERENT POLYGONS OUT OF THE JSON STRING

        return ID;

    }

    public ArrayList<Point> getLastUpdateLine(int ID) throws IOException, JSONException {// STARTING METHOD TO GET THE LINES
        String url = "http://cgi.godenzonenvanatlas.nl/arcgis/rest/services/HesterTrainTracking/SnapTestLinesFS/FeatureServer/0/query?where=OBJECTID+%3D+%27"
                +ID +
                "%27&objectIds=&time=&geometry=&geometryType=esriGeometryEnvelope&inSR=&spatialRel=esriSpatialRelIntersects&relationParam=&outFields=*&return"
                + "Geometry=true&maxAllowableOffset=&geometryPrecision=&outSR=&gdbVersion=&returnDistinctValues=false&returnIdsOnly=false&returnCountOnly=false"
                + "&orderByFields=&groupByFieldsForStatistics=&outStatistics=&returnZ=false&returnM=false&f=pjson";
        // REQUEST IS IMPORTED WITH THE ID AS VARIABLE
        String response = getData(url);// METHOD TO GET DATA FROM THE REQUEST URL
```

```java
        ArrayList<Point> line = getPolyline(response); // GETS THE LINES THAT ARE REQUESTED OUT OF THE JSON STRING
        return line;
    }


    private ArrayList<Point> getPolyline(String response) throws JSONException { //GETS THE LINES OUT OF THE JSON STRING
        JSONObject data_input = new JSONObject(response);
        JSONArray features = data_input.getJSONArray("features");
                    JSONArray paths = features.getJSONObject(0).getJSONObject("geometry").getJSONArray("paths").getJSONArray(0);

                    ArrayList<Point> line = new ArrayList<Point>();
                    for (int i = 0; i < paths.length(); i++){
                        Point point = new Point(paths.getJSONArray(i).getDouble(0),paths.getJSONArray(i).getDouble(1));
                        line.add(point);
                    }
        return line;
    }


    private int getPolygon(String response) throws JSONException {//GETS THE IDS OUT OF THE JSON STRING
        JSONObject data_input = new JSONObject(response);
        JSONArray features = data_input.getJSONArray("features");
        int ID=-1;
        for (int i=0; i < features.length(); i++){
            if (features.getJSONObject(i).has("attributes")) {
                if (features.getJSONObject(i).getJSONObject("attributes").has("OBJECTID")){
                    ID= features.getJSONObject(i).getJSONObject("attributes").getInt("OBJECTID");
                }
            }
        }
        return ID;
    }

    public String getData(String url) throws IOException {// DATA IS IMPORTED AND RETURNED AS A JSON STRING

        URL obj = new URL(url);
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();

        // optional default is GET
        con.setRequestMethod("GET");

        // add request header
        con.setRequestProperty("User-Agent", USER_AGENT);

        BufferedReader in = new BufferedReader(new InputStreamReader(
                con.getInputStream()));


        String inputLine;
        StringBuffer response = new StringBuffer();

        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();

        return response.toString();

    }

}
```

B4 PERCENTAGE ON TRACK PROCESSOR

PROCESSOR

```java
package com.cgi.processor.undagrid;

 * #%L

import java.util.ArrayList;
import java.util.Collections;
import com.esri.ges.core.component.ComponentException;
import com.esri.ges.core.geoevent.GeoEvent;
import com.esri.ges.core.validation.ValidationException;
import com.esri.ges.manager.geoeventdefinition.GeoEventDefinitionManager;
import com.esri.ges.processor.GeoEventProcessorBase;
import com.esri.ges.processor.GeoEventProcessorDefinition;

public class PercentageonRailProcessor extends GeoEventProcessorBase {// PROCESSOR INITIATED
    // Spatial spatial;
    GeoEventDefinitionManager manager;
        public PercentageonRailProcessor(GeoEventProcessorDefinition definition,
            GeoEventDefinitionManager m) throws ComponentException {
        super(definition);
        // spatial = s;
        geoEventMutator = true;
        manager = m;
    }

    @Override
    public void afterPropertiesSet() {
        // Initialization Phase ...

    }

    @Override
    public synchronized void validate() throws ValidationException {

    }

    @Override
    public GeoEvent process(GeoEvent ge) throws Exception {// PROCESSOR OPERATION PHASE STARTED

        int trackId = (Integer) ge.getField("trackId"); // TRACKID AND SENSORID ARE IMPORTED FROM EARLIER GEOEVENT
        String name = (String) ge.getField("sensorId");
        Connector getData = new Connector(); // CONNECTER INITIATED TO ENABLE EXTRACTION OF PREVIOUS TRACK NUMBERS FROM SERVER

        ArrayList<Integer> trackIds = getData.getLastUpdate(name); // METHOD EXECUTED THAT RETURNS AN ARRAYLIST OF THE TRACKNUMBERS
        if (trackIds.size() == 0){ // IF THERE ARE NO PREVIOUS RESULTS (FOR EXAMPLE IN THE BEGINNING) NOTHING IS CHANGED AT THE GEOEVENT
            return ge;
        }
        trackIds.add(trackId); // THE CURRENT TRACK NUMBER IS ALSO ADDED


      int percentageTrack = percentageCalculator(trackId, trackIds); // THE PERCENTAGE CALCULATOR IS EXECUTED

      ge.setField("PercentageOn", percentageTrack); // THE PERCENTAGE AND THE LARGEST TRACK NUMBER ARE ADDED TO THE GEOEVENT

      return ge;
  }



  private int percentageCalculator(double trackId,ArrayList<Integer> trackIds) { // THE PERCENTAGE CALCULATION METHOD
      int occurences = Collections.frequency(trackIds,  trackId);
      int toPercentage = occurences * 100;
      int percentage = toPercentage/trackIds.size();

      return percentage;
  }

  @Override
  public void shutdown() {
      // Destruction Phase
      super.shutdown();
  }

  @Override
  public boolean isGeoEventMutator() {
      return true;
  }

  public void onServiceStart() {
      // Service Start Phase
  }

  public void onServiceStop() {
      // Service Stop Phase
  }
```

CONNECTOR

```java
package com.cgi.processor.undagrid;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.esri.core.geometry.Point;


public class Connector {// INITIATING CONNECTOR CLASS

    private final static String USER_AGENT = "Mozilla/5.0";

    public Connector() {
    }

    public ArrayList<Integer> getLastUpdate(String name) throws IOException,JSONException {// STARTING METHOD TO GET THE PREVIOUS TRACK NUMBERS


        String url = "http://cgi.godenzonenvanatlas.nl:6080/arcgis/rest/services/HesterTrainTracking/PreviousTrackIds/FeatureServer/0/query?where=sensorId+%3D+%27"
        +name +
        "%27&objectIds=&time=&geometry=&geometryType=esriGeometryEnvelope&inSR=&spatialRel=esriSpatialRelIntersects&relationParam=&outFields=*&returnGeometry=true&"
        + "maxAllowableOffset=&geometryPrecision=&outSR=&gdbVersion=&returnDistinctValues=false&returnIdsOnly=false&returnCountOnly=false&orderByFields=&groupByFields"
        + "ForStatistics=&outStatistics=&returnZ=false&returnM=false&f=pjson";
        // REQUEST IS IMPORTED WITH SENSORID AS VARIABLE
        String response = getData(url);// METHOD TO GET DATA FROM THE REQUEST URL

        ArrayList<Integer> trackIds = getIds(response);//METHOD TO GET TRACK NUMBERS OUT OF JSON STRING

        return trackIds;

    }

    public ArrayList<Point> getLastUpdateLine(int ID) throws IOException, JSONException {// STARTING METHOD FOR SNAP OPERATION (SEE SNAP ALGORITHM

        String url = "http://cgi.godenzonenvanatlas.nl/arcgis/rest/services/HesterTrainTracking/SnapTestLinesFS/FeatureServer/0/query?where=OBJECTID+%3D+%27"
                +ID +
                "%27&objectIds=&time=&geometry=&geometryType=esriGeometryEnvelope&inSR=&spatialRel=esriSpatialRelIntersects&relationParam=&outFields=*&return"
                + "Geometry=true&maxAllowableOffset=&geometryPrecision=&outSR=&gdbVersion=&returnDistinctValues=false&returnIdsOnly=false&returnCountOnly=false"

                + "&orderByFields=&groupByFieldsForStatistics=&outStatistics=&returnZ=false&returnM=false&f=pjson";
        String response = getData(url);
        ArrayList<Point> line = getLine(response);
        return line;

    }

    public String getData(String url) throws IOException {// DATA IS IMPORTED AND RETURNED AS A JSON STRING

        URL obj = new URL(url);
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();

        // optional default is GET
        con.setRequestMethod("GET");

        // add request header
        con.setRequestProperty("User-Agent", USER_AGENT);

        BufferedReader in = new BufferedReader(new InputStreamReader(
                con.getInputStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();

        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();
        return response.toString();

    }

    public ArrayList<Integer> getIds(String response) throws JSONException{ // METHOD TO GET THE TRACK NUMBERS OUT OF THE JSON STRING
        ArrayList<Integer> trackIds = new ArrayList <Integer>();
        JSONObject data_input = new JSONObject(response);
        JSONArray features = data_input.getJSONArray("features");
        for (int i=0; i < features.length(); i++){
            if (features.getJSONObject(i).has("attributes")) {
                if (features.getJSONObject(i).getJSONObject("attributes").has("trackId")){
                    int Id = features.getJSONObject(i).getJSONObject("attributes").getInt("trackId");
                    trackIds.add(Id);
                }
            }
        }
        return trackIds;

    }
    private ArrayList<Point> getLine(String response) throws JSONException { // METHOD TO GET THE LINE (SEE SNAP ALGORITHM)
```

```java
        ArrayList<Point> line = new ArrayList<Point>();
        JSONObject data_input = new JSONObject(response);
        JSONArray features = data_input.getJSONArray("features");
        for (int i=0; i < features.length(); i++){
            if (features.getJSONObject(i).has("geometry")) {
                if (features.getJSONObject(i).getJSONObject("geometry").has("paths")){
                    JSONArray paths = features.getJSONObject(i).getJSONObject("geometry").getJSONArray("paths");
                    double tempx1 = paths.getJSONArray(0).getJSONArray(0).getDouble(0);
                    double tempy1 = paths.getJSONArray(0).getJSONArray(0).getDouble(1);
                    double tempx2 = paths.getJSONArray(0).getJSONArray(1).getDouble(0);
                    double tempy2 = paths.getJSONArray(0).getJSONArray(1).getDouble(1);
                    Point point1 = new Point (tempx1,tempy1);
                    Point point2 = new Point (tempx2, tempy2);
                    line.add(point1);
                    line.add(point2);
                }
            }
        }
        return line;
    }
```

B5 ORDER OF WAGONS PROCESSOR

PROCESSOR

```java
package com.cgi.processor.undagrid;

 * #%L

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

import com.cgi.processor.undagrid.Connector.Combi;
import com.esri.core.geometry.Point;
import com.esri.core.geometry.SpatialReference;
import com.esri.ges.core.component.ComponentException;
import com.esri.ges.core.geoevent.GeoEvent;
import com.esri.ges.core.validation.ValidationException;
import com.esri.ges.manager.geoeventdefinition.GeoEventDefinitionManager;
import com.esri.ges.processor.GeoEventProcessorBase;
import com.esri.ges.processor.GeoEventProcessorDefinition;
import com.esri.core.geometry.GeometryEngine;

public class OrderofWagonsProcessor extends GeoEventProcessorBase {// PROCESSOR INITIATED
    // Spatial spatial;
    GeoEventDefinitionManager manager;
        public OrderofWagonsProcessor(GeoEventProcessorDefinition definition,
            GeoEventDefinitionManager m) throws ComponentException {
        super(definition);
        // spatial = s;
        geoEventMutator = true;
        manager = m;
    }

    @Override
    public void afterPropertiesSet() {
        // Initialization Phase ...

    }

    @Override
    public synchronized void validate() throws ValidationException {

    }

    @Override
    public GeoEvent process(GeoEvent ge) throws Exception { // PROCESSOR OPERATION PHASE STARTED
        // Operation phase...
        double lat = (Double) ge.getField("lat"); // X, Y, TIMESTAMP AND SENSORNAME ARE EXTRACTED FROM THE EARLIER GEOEVENT
        double lon = (Double) ge.getField("lon");
        Point point = new Point(lon,lat);
```

```
long ts = Math.round((Double) ge.getField("ts2"));
String sensorName = (String) ge.getField("sensorName");

Connector getData = new Connector(); // CONNECTOR IS INITIATED TO EXTRACT DATA FROM THE SERVER
long ts2 = ts - 50; // TIMESTAMP FOR THE DATA COLLECTION IS SET
ArrayList<Combi> coorscombi = getData.getLastUpdate(trackId,ts2); // POINT DATA FROM BOTH NEIGHBOURS IS EXTRACTED FROM THE SERVER

if (coorscombi.isEmpty()){// IF THERE IS NO POINT DATA ON THE SERVER, THE GEOEVENT WILL BE RETURNED
    ge.setField("point", toJSON(lon,lat));
    return ge;
}
SpatialReference s = null;
s.create(4326);
ArrayList<Combi> distances = new ArrayList<Combi>();
for (int i = 0; i < coorscombi.size(); i++){
    double distance = GeometryEngine.distance(coorscombi.get(i).getPoint(), point, s);
    coorscombi.get(i).setDistance(distance);
    distances.add(coorscombi.get(i));
}
Collections.sort(distances, new Comparator<Combi>() {
    public int compare(Combi dist1, Combi dist2)
    {
        return  Double.compare(dist1.getDistance(), dist2.getDistance());
    }
});
ArrayList<String> result = new ArrayList<String>();
result.add(sensorName);
double bearing1 = getBearing(distances.get(0).getPoint(),point);
result.add(distances.get(0).getSensor());
for (int j=1;j<distances.size();j++){
    double bearing = getBearing(distances.get(j).getPoint(), point);
    if (Math.abs(bearing - bearing1) > 90){
        result.add(0,distances.get(j).getSensor());
    }
    else{
        result.add(distances.get(j).getSensor());
    }
}
int id = result.indexOf(sensorName);
if (id != 0){
    String neighbour1 = result.get(id-1);
    ge.setField("neighbour1", neighbour1);
}
if (id != result.size()-1){
    String neighbour2 = result.get(id+1);
    ge.setField("neighbour2", neighbour2);
}

    ge.setField("point", toJSON(lon,lat));
    return ge;
    }

private double getBearing(Point point1, Point point2) {
    double x1 = point1.getX();
    double x2 = point2.getX();
    double dx = x1-x2;
    double distance = getDistance(point1,point2);
    double result = Math.asin(dx/distance);
    double resultDegree = Math.toDegrees(result);
    return resultDegree;
}

private double getDistance(Point point1, Point point2){
    double x1 = point1.getX();
    double x2 = point2.getX();
    double y1 = point1.getY();
    double y2 = point2.getY();
    double dx = x1-x2;
    double dy = y1-y2;
    double distance = Math.sqrt(Math.pow(dx, 2)+Math.pow(dy, 2));
    return distance;
}

private Object toJSON(double lon, double lat) { // METHOD TO MAKE THE POINT INTO A JSON REPRESENTATION
    String stringify = "{\"x\":" + lon + ",\"y\":" + lat + ", \"spatialReference\":{\"wkid\":4326}}";


     return stringify;
}

@Override
public void shutdown() {
    // Destruction Phase
    super.shutdown();
}

@Override
public boolean isGeoEventMutator() {
    return true;
}

public void onServiceStart() {
    // Service Start Phase
}
```

```
    public void onServiceStop() {
        // Service Stop Phase
    }

}
```

CONNECTOR

```
package com.cgi.processor.undagrid;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import com.esri.core.geometry.Point;

public class Connector {// INITIATING CONNECTOR CLASS

    private final static String USER_AGENT = "Mozilla/5.0";

    public Connector() {
    }

    public Point getLastUpdate(String name, double ts) throws IOException,JSONException {// STARTING METHOD TO GET NEIGHBOURING POINT
        String url = "http://cgi.godenzonenvanatlas.nl:6080/arcgis/rest/services/HesterTrainTracking/PreviousTrackIds/FeatureServer/0/query?where=sensorName+%3D+%27"
        +name + "&27+AND+ts>&27" +ts +
        "%27&objectIds=&time=&geometry=&geometryType=esriGeometryEnvelope&inSR=&spatialRel=esriSpatialRelIntersects&relationParam=&outFields=*&returnGeometry=true&"
        + "maxAllowableOffset=&geometryPrecision=&outSR=&gdbVersion=&returnDistinctValues=false&returnIdsOnly=false&returnCountOnly=false&orderByFields=&groupByFields"
        + "ForStatistics=&outStatistics=&returnZ=false&returnM=false&f=pjson";
        // REQUEST IS IMPORTED WITH NODENAME AND TIMESTAMP AS VARIABLE
        String response = getData(url);// METHOD TO GET DATA FROM THE REQUEST URL

        Point point = getPoint(response);// GET THE POINT FROM THE IMPORTANT JSON STRING
        return point;
    }

    public String getData(String url) throws IOException {// DATA IS IMPORTED AND RETURNED AS A JSON STRING

        URL obj = new URL(url);
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();

        // optional default is GET
        con.setRequestMethod("GET");

        // add request header
        con.setRequestProperty("User-Agent", USER_AGENT);

        BufferedReader in = new BufferedReader(new InputStreamReader(
                con.getInputStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();

        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();
        return response.toString();
    }

    public Point getPoint(String response) throws JSONException{ // THE POINT IS IMPORTED FROM THE JSON STRING
        Point point= new Point();
        JSONObject data_input = new JSONObject(response);
        JSONArray features = data_input.getJSONArray("features");
        for (int i=0; i<features.length(); i++){
            if (features.getJSONObject(i).has("attributes")) {
                if (features.getJSONObject(i).getJSONObject("attributes").has("lon") && features.getJSONObject(i).getJSONObject("attributes").has("lat")){
                    double x = features.getJSONObject(i).getJSONObject("attributes").getDouble("lon");
                    double y = features.getJSONObject(i).getJSONObject("attributes").getDouble("lat");
                    point.setX(x);
                    point.setY(y);
                    }
                }
            }

        return point;
    }
}
```

D. TEST RESULTS

C1. GRAPHS OF RAW DATA

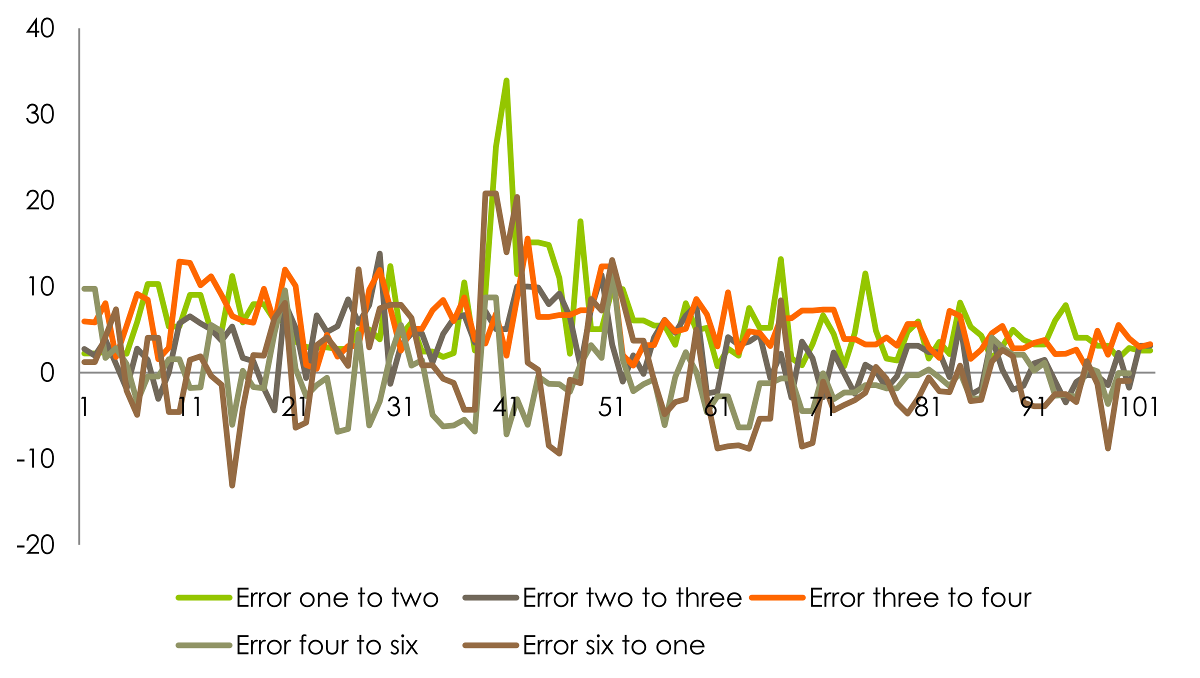


FIGURE 58 VARIANCES IN DISTANCES OF RAW DATA 9TH OF MAY 2015, Y-AXIS: DISTANCE (METERS), X-AXIS: TIME SINCE FIRST 20 RESULTS (MINUTES)

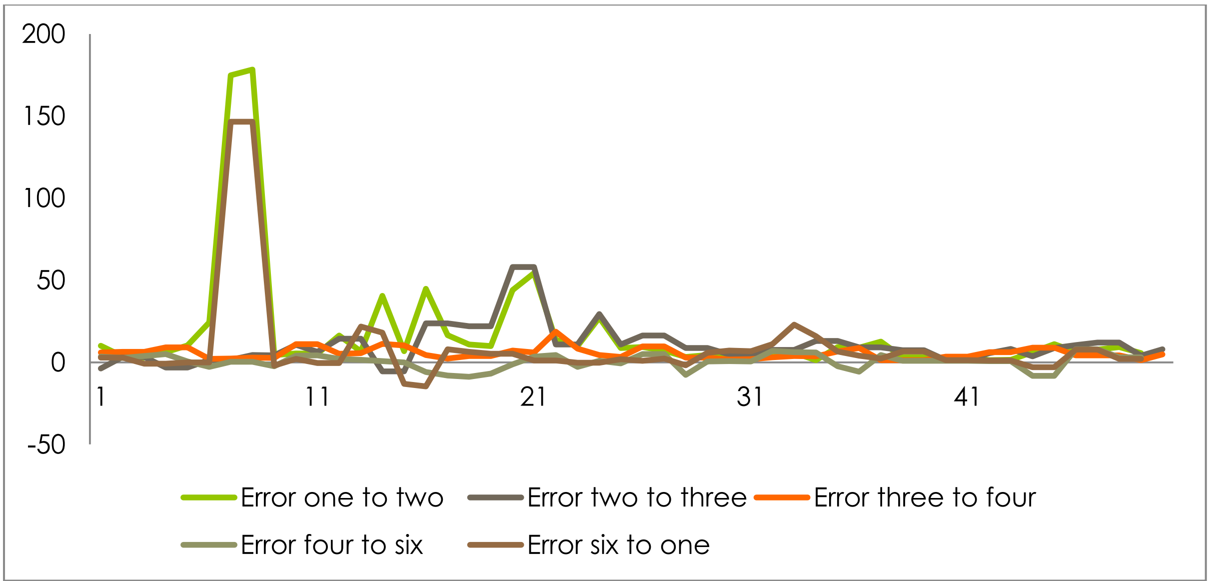


FIGURE 59 VARIANCES IN DISTANCES OF RAW DATA 3RD OF MAY 2015, Y-AXIS: DISTANCE (METERS), X-AXIS: TIME SINCE FIRST 20 RESULTS (MINUTES)

FIGURE 60 VARIANCES IN DISTANCES OF RAW DATA 4TH OF MAY 2015, Y-AXIS: DISTANCE (METERS), X-AXIS: TIME SINCE FIRST 20 RESULTS (MINUTES)