

Galileo High Accuracy Services

Analysis of its potential for cadastral surveying

Group 1

Michaja van Capel 4657071

Christos Chontos 5579392

Irina Gheorghiu 5627834

Tendai Mbwanda 521554

Synthesis Project

GEO1101

November 2022



Synthesis Project

**Galileo High Accuracy Service
Analysis of its potential for cadastral
surveying**

Michaja van Capel
Christos Chontos
Irina Gheorghiu
Tendai Mbwanda

November 2022

Supervisor: Edward Verbree
Kadaster: Lennard Huisman
GISCAD-OV: Ivars Nudiens

Abstract

The NMCA's (National Mapping and Cadastral Agencies) of European countries have different cadastral survey accuracy standards (European Global Navigation Satellite Systems Agency, 2019). In order to meet these standards, the appropriate equipment and services should be determined. The augmentation service Galileo High Accuracy Service (HAS), that is planned for 2022, will provide high accuracy Precise Point Positioning (PPP) corrections. Unlike other high-accuracy services, the Galileo HAS will be free of charge and available worldwide, without the need to be close to a base station or to a dedicated provider network. The PPP corrections will be provided through the Galileo signal as well as through the Internet (EUSPA, 2021). Because of the potential of the Galileo HAS, for the Synthesis Project we want to get insight in the accuracy of the augmentation service. Since a big share of cadastral surveys is performed in the built environment, we also want to determine the accuracy in an urban canyon. With the found accuracy, we can possibly judge whether Galileo HAS is suitable for cadastral surveys in the Netherlands, by comparing the measured accuracy to cadastral survey accuracy standards of the Dutch Kadaster.

As a final conclusion for this project, Galileo HAS is still a technique under development and the PPP-based correction methods are currently not as accurate as the RTK-based ones. Galileo HAS will present in the future ways to correct these errors.

Acknowledgements

This report is documenting the research work carried out in order to test and look further into the characteristics and possibilities of the Galileo High Accuracy Services usage in the built environment. This project was conducted by a team of four, second year students from the Geomatics Master, Faculty of Architecture and Built Environment, TU Delft for a period of 10 weeks to implement this topic given by Kadaster, which is a governmental organisation in the Netherlands, responsible for providing Cadastral Surveying services, maintaining the Land Registry Database and Topography Basic Registration. For this topic, there was a lot of effort in acquiring more knowledge regarding the different and broad notions surrounding it.

Throughout this project, the team cooperated with both Kadaster, in obtaining analysis data from Delft, the Netherlands and with the representatives of the H2020 GISCAD-OV project. The experts from the GISCAD-OV project, that were involved in the synthesis, are responsible for conducting the task of data acquisition and agreed, shared and explained their technology, equipment and data with the team, which was central for the successful completion of the project.

We would like to express our gratitude to our supervisor, Edward Verbree, our sponsor from Kadaster, Lennard Huisman and the representatives of the GISCAD-OV project, Ivars Nudiens and Roberto Capua who communicated and shared their knowledge and expertise with us to help in further developing a new GNSS-based technique.

Management - Team and Organisation

0.1 Team



Michaja van Capel

Country: the Netherlands

BSc: Applied Earth Sciences (TU Delft), September 2017 - April 2022

MSc: Geomatics (TU Delft), September 2021 - Present

Role: technical manager - in charge of the data analysis. Responsible for handling the 3D data and programming the line of sight and visibility analysis.



Tendai Mbwanda

Country: Zimbabwe

BSc: GIS and Earth Observation (University of Zimbabwe), September 2015 - June 2019

MSc: Geomatics (TU Delft), September 2021 - Present

Management - Team and Organisation

Role: reporting manager - responsible for recording minutes during meetings and the preparation of presentations and documents. Responsible for programming the coordinate transformations and validation of the visibility analysis.



Irina Gheorghiu

Country: Romania

BSc: Geodesy and Land Surveying (TUCE Bucharest), October 2017 - July 2021

MSc: Geomatics (TU Delft), September 2021 - Present

Role: communication manager - responsible for the communication between the team members and external parties (client, supervisor, other third party contributors). Responsible for the accuracy analysis in the project.



Christos Chontos

Country: Greece

BSc+MSc: Rural, Surveying and Geoinformatics Engineering (September 2012 - September 2018)

Role: project manager - keeps an overview of the project, makes sure the deadlines are met and the project remains within the scope. Responsible for the convergence time analysis.

0.2 Collaboration

The code-related work can be found in this repository:
<https://github.com/tendaiwbm/synthesis/tree/main>

0.3 Geomatics courses

The Synthesis project is a project that combines all knowledge gained from the Geomatics courses. The following courses' contents were used:

- GEO1000: Python Programming
- GEO1001: Sensing Technologies
- GEO1003: Positioning and Location Awareness
- GEO1002: GIS and Cartography
- GEO1015: Digital Terrain Modeling
- GEO1004: 3D Modeling
- GEO1009: Geo-information Governance
- GEO5014: Geomatics as Support for Energy Applications

Contents

| | |
|---|-----------|
| Management - Team and Organisation | ix |
| 0.1 Team | ix |
| 0.2 Collaboration | xi |
| 0.3 Geomatics courses | xi |
| 1 Introduction | 1 |
| 1.1 Problem Definition | 1 |
| 1.2 Research Question | 2 |
| 2 Theory and Context | 3 |
| 2.1 Galileo HAS | 3 |
| 2.1.1 Precision Point Positioning corrections | 4 |
| 2.1.2 PPP versus RTK | 5 |
| 2.2 Convergence Time | 5 |
| 2.3 Cadastral surveying and land administration | 6 |
| 2.4 GISCAD-OV | 8 |
| 3 Methodology | 11 |
| 3.1 Pilot campaigns in the GISCAD-OV project | 11 |
| 3.1.1 Data Acquisition and Equipment | 11 |
| 3.2 Visibility Analysis | 13 |
| 3.2.1 CRS conversion | 15 |
| 3.2.2 Line of Sight | 16 |
| 3.3 Analysis of Datasets | 17 |
| 3.3.1 Coordinate transformation | 18 |
| 3.3.2 Convergence time analysis | 19 |
| 4 Results | 21 |
| 4.1 Datasets GISCAD-OV | 21 |
| 4.2 Visibility analysis | 23 |
| 4.2.1 Results from the script | 23 |
| 4.2.2 Validation | 25 |
| 4.3 Convergence time analysis | 28 |
| 5 Conclusion | 31 |
| 5.1 Performance of Galileo HAS | 31 |
| 5.2 Limitations | 32 |
| Bibliography | 33 |
| Appendix I: Algorithms | 35 |
| 1 Visibility Analysis | 35 |
| 1.1 Line of Sight | 35 |

Contents

1.2 Visibility of satellites 38

List of Figures

| | | |
|------|---|----|
| 2.1 | Galileo HAS (EUSPA, 2021) | 3 |
| 2.2 | Comparison between PPP and NRTK (Henry, 2022) | 5 |
| 2.3 | Proposed basic classes of the Surveying Representation sub-package (Kalogianni, et al., 2022) | 8 |
| 2.4 | logo GISCAD-OV project (European Global Navigation Satellite Systems Agency, 2019) | 9 |
| 3.1 | Steps of the methodology | 11 |
| 3.2 | Measurement Open Sky | 12 |
| 3.3 | Measurement Urban Area | 13 |
| 3.4 | Tile of interest visualized in FZKViewer | 14 |
| 3.5 | Tile of interest visualized in QGIS | 14 |
| 3.6 | Structure of the Two Line Element (CelesTrak, 2022) | 15 |
| 3.7 | Chain of coordinate transformations performed | 16 |
| 3.8 | FME Workbench workflow for CityGML to .obj conversion | 16 |
| 3.9 | the sign of volume (a,b,1,2), sign of volume (a,b,0,1) and sign of volume (a,b,0,2) are all positive (Aftosmis, 1996) | 17 |
| 3.10 | Explanations of the message segments | 18 |
| 4.1 | Visible GPS+Galileo satellites from test point 1 — top view | 24 |
| 4.2 | Visible GPS+Galileo satellites from test point 1 — top side view | 24 |
| 4.3 | Visible GPS+Galileo satellites from test point 2 — top view | 25 |
| 4.4 | Validation location 1 | 26 |
| 4.5 | Satellite skyplot from location 1 (GPS + Galileo) | 26 |
| 4.6 | Validation location 2 | 27 |
| 4.7 | Satellite skyplot from location 2 (GPS + Galileo) | 27 |
| 4.8 | PPP-Convergence Time Diagram | 28 |
| 4.9 | Statistics Table | 28 |
| 4.10 | PPP-RTK-Convergence Time Diagram | 28 |

Acronyms

| | |
|-----------|---|
| CRS | Coordinate Reference System |
| CS | Cadastral Surveying |
| DTM | Digital Terrain Model |
| EUSPA | European Union Agency for the Space Programme |
| GISCAD-OV | Galileo Improved Services for Cadastral Augmentation Development On-field Validation |
| GNSS | Global Navigation Satellite System |
| GPS | Global Positioning System |
| HA | High Accuracy |
| HA DG | High Accuracy Data Generator |
| HAS | High Accuracy Service |
| LADM | Land Administration Domain Model |
| LoS | Line of Sight |
| NMCA | National Mapping and Cadastral Agency |
| NRTK | Network Real Time Kinematic |
| RTK | Real Time Kinematic |
| PPP | Precise Point Positioning |
| PPP-AR | Precise Point Positioning Ambiguity Resolution |
| SSR | State Space Representation |
| TEME | True Equinox Mean Earth |
| TLE | Two-line Element Set |

1 Introduction

Commonly known as Global Navigation Satellite Systems (GNSS), systems of satellites orbiting the earth to enable positioning on, above or below its surface have become widely adopted in everyday life. Cadastral surveying is an example of a GNSS application area which additionally requires high accuracy positioning achieved through augmentation services (EUSPA, 2019). Real Time Kinematic (RTK) and Precise Point Positioning (PPP) are two standardised techniques to attain high accuracy measurements which according to the European Global Navigation Satellite Systems Agency (2019) ranges between centimetre and decimetre levels. Nevertheless, until the introduction of Europe's own GNSS, Galileo, the NAVSTAR and GLONASS constellations largely provided a basis for augmentation. As the number of Galileo satellites increased between 2016 and 2019, Huisman et. al (2020) illustrate the associated improvement in availability of the Galileo GNSS service. This availability was presented using the RTK method by measuring only with signals from the available Galileo satellites. These available signals will later serve in the development of the free of charge Galileo HAS. Galileo High Accuracy Service (HAS) will provide free of charge high accuracy using Precise Point Positioning (PPP) corrections for Galileo and GPS through the E6-B signal, as well as creation of markets to enable and encourage innovative use of these services.

For the synthesis project, the potential of using the Galileo HAS for cadastral surveying is explored. The project is closely related to the H2020 GISCAD-OV project, which is currently at its final year and the consortium comprises of 14 partners including TU Delft. The main scope of the GISCAD-OV project is to design, develop and validate a cost-effective High Accuracy Service based on GPS and Galileo HAS, as well as Precise Point Positioning-Ambiguity Resolution (PPP-AR) quick convergence techniques (European Global Navigation Satellite Systems Agency, 2019). The data acquired, through the pilot project campaign of the GISCAD-OV project will be used to explore the potential of Galileo HAS for cadastral surveying.

1.1 Problem Definition

Unlike other high-accuracy services, the Galileo HAS will be free of charge and available worldwide, without the need to be close to a base station or to a dedicated provider network, as RTK services. The PPP corrections will be provided through the Galileo signal as well as through the Internet (EUSPA, 2021). The benefit of providing cadastral surveying services with comparable accuracy but at a lower expense, with global coverage and no provider network is evident. For these reasons, the Galileo High Accuracy Service has attracted attention at regional and national scales.

The European Global Navigation Satellite Systems Agency (GSA) has embarked on the Galileo Improved Services for Cadastral Augmentation Development of On-field Validation (GISCAD-OV) project to design, develop and validate an innovative and cost-effective High Accuracy Service for cadastral surveying applications, based on GPS and Galileo High Accuracy Services (HAS) and Precise Point Positioning-Ambiguity Resolution quick convergence (PPP-AR) advanced techniques. At national level, the Dutch Land Registry known as Kadaster is a key stakeholder in the GNSS value chain as it provides cadastral surveying services and maintains the Land Registry Database and Topography Basic Registration. Because other players in the value chain make use of these services in numerous markets, the importance of high accuracy positioning cannot be over-emphasized. Currently, Kadaster achieves high accuracy positioning through Network Real Time Kinematic (NRTK) positioning. With the introduction of Galileo HAS edging closer, there however is no immediate indication of its accuracy for cadastral surveying in the built environment in comparison to the current techniques, although Huisman *et. al* (2020) demonstrate the possibility of a reconstructing a parcel boundary using Galileo only RTK. On this backdrop, performance of Galileo HAS in urban areas for cadastral surveying applications can be questioned and explored. However, urban areas present blockage of satellite signals by the built environment. Because of the potential of the Galileo HAS, insight in the accuracy, availability and convergence time of the service for cadastral surveying in different environments is desired.

1.2 Research Question

In consultation with Kadaster, the research question of the project has been formed as follows:

To what extent is, the performance of Galileo High Accuracy Service, suitable for cadastral surveying?

The research question is split up into 3 sub-questions, which treat different aspects of High Accuracy Services that are important for cadastral surveying: accuracy, convergence time and availability. Based on the answers of the sub-questions, an answer to the research question can be formulated.

- How accurate are cadastral surveying measurements by Galileo HAS compared to the current way of working?
- What is the convergence time of Galileo HAS measurements compared to the current way of working?
- How does the availability of Galileo change with different environments (for example: urban area vs open sky)?

In the sub-questions the different aspects of the performance of Galileo HAS is compared to the current way of working. The current way of working for high accuracy cadastral surveying by Kadaster is through NRTK positioning, for which accuracies of 2-3 cm can be achieved. The difference between RTK and PPP is explained in Chapter 2. It is expected that the accuracy and convergence time of Galileo HAS will not be as small as the accuracy and convergence time of NRTK. Even though these values can not be reached with the new method, we want to emphasize the potential of Galileo HAS.

2 Theory and Context

In this chapter the necessary background material and core concepts of this research are introduced and explained. In Section 2.1 the method of Galileo HAS will be explained. The concept of PPP corrections and a comparison between PPP and RTK will also be provided in this section. In Section 2.2 convergence time of positioning techniques will be explained. In Section 2.3 cadastral surveying and land administration will be introduced. The last section provides a more detailed background on the GISCAD-OV project.

2.1 Galileo HAS

The main objective of the Galileo High Accuracy Service is to provide free of charge high accuracy corrections through the PPP method, provided via either the E6-B signal of Galileo satellites or over the internet. This means real time positioning performance with a nominal accuracy of 20 cm. By providing the corrections via the Galileo signal, there is no need for the user to be connected to a local terrestrial infrastructure. With RTK techniques, positioning is done using as reference a local base station or a network. Therefore Galileo HAS, which is also a free of charge service and promised to be globally available, will create new possibilities in the GNSS signal augmentation market.

The high accuracy corrections will consist of satellite corrections, including orbits, clock, code and phase biases for each satellite. These corrections are calculated with data from the Galileo Core Infrastructure, by the High Accuracy Data Generator (HA DG), which can be seen in Figure 3.1. In the future, Galileo HAS is promised to provide atmospheric corrections on top of the satellite corrections. Even though the corrections are provided via the Galileo E6-B signal (or by terrestrial means) The HA corrections will be considered multi-constellation/multi-frequency, since the corrections will be computed for both Galileo (E1, E5a, E5b, E6) and GPS satellites (L1, L5, L2C) (Naciri et al., 2020).

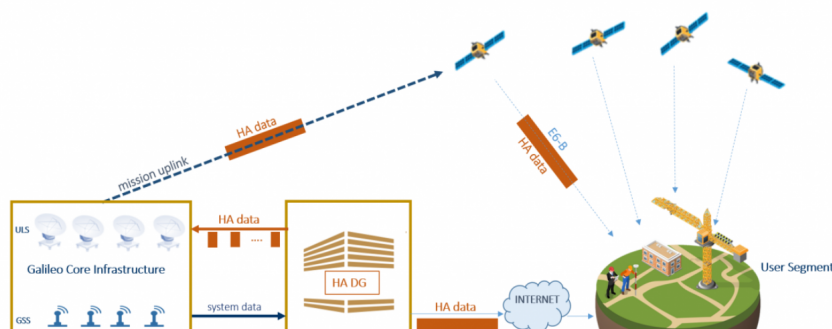


Figure 2.1: Galileo HAS (EUSPA, 2021)

2.1.1 Precision Point Positioning corrections

Galileo HAS provides PPP corrections. Precise Point Positioning (PPP) was first developed as a technique by R. Anderle, 1976. In the 1990s, the technique started to be studied at the National Aeronautics and Space Administration (NASA), mostly the dual-frequency type of PPP. Focus was increased on accessing accurate satellite ephemerides with short latency and frequent updates in order to support real-time kinematic PPP applications. Due to the RTK method being accurate and cost-effective, the focus on PPP has also shifted to real-time solutions (Grinter Roberts, 2011).

PPP is a signal augmentation method which eliminates the GNSS system errors using a single GNSS receiver. This type of technique is based itself on the availability of clock and orbit corrections from the GNSS satellites belonging to the constellations. This makes the PPP technique independent of the tracked data from reference station around the user location. Thus it can be stated that PPP establishes a direct connection between the user and the GNSS satellite coordinates. PPP provides a positioning solution in a reference frame such as the International Terrestrial Reference Frame (Altamimi et al, 2011)

According to the Novatel Manual - An Introduction to GNSS, Chapter 5, a PPP solution will require a period of time to convergence to decimetre accuracy so as to solve biases such as atmospheric conditions, satellite geometry and multipath environment. The converge time and the accuracy required are dependent on the quality of the corrections and the way they are applied on the receiver.

There are ways in which the main PPP error sources can be diminished (Novatel, 2015):

- Dual-frequency operation: the first order ionospheric delay can be eliminated using combinations of dual-frequency measurements, due to the delay being dependent on the carrier wave frequency
- External error correction data: the satellite orbit and clock corrections
- Modeling: correction of the tropospheric delay. However this delay is highly varying and it can never be totally corrected. The PPP receiver will also use Modeling to correct the solid earth tides effect.
- PPP Kalman algorithms: The Extended Kalman Filter will minimize the noise in the system and will make possible the estimation of the position with centimetre level accuracy. The estimates for the EKF state are improved with successive GNSS measurement, until they converge to stable and accurate values.

The main usage for this method are the broad range of applications in remote areas, distant areas from the reference stations that offer limited inter-visibility.

It has to be mentioned that the main limitation for PPP corrections is that the ambiguities of the carrier phase errors are not solved, but rather an estimation of them is used. One limitation of this aspect is that the convergence period will be long. For example, the estimated convergence time for a horizontal error under 10 cm is between 20-40 minutes. However, this aspect is also dependent on the satellite geometry, number of available satellites, quality of the corrections, receiver multipath environment and atmospheric conditions (Novatel, 2015).

State space representation (SSR) is the message format used for the PPP correction method. It separated the error and provides correction for each error. This format is independent from the user location and it will provide the PPP corrections (European GSA, 2019).

2.1.2 PPP versus RTK

Both the PPP and RTK method offer more accurate positioning by providing corrections. Both PPP and RTK methods provide corrections for the satellite SIS errors such as orbit, clock, and signal code bias. On top of that, RTK also provides corrections for the atmospheric effects. Therefore, the PPP-RTK method offers even higher accuracy than using PPP on its own. The level of accuracy is dependent on the distance length between location of the base station and the rover. If the distance between the base station and the rover increases, then the accuracy will decrease. RTK also has a smaller convergence time, but it has a more complex setup and is more expensive.

The PPP method setup has a simpler configuration and it is not dependent on the baseline length between rover and base station, as it does not directly use one. (Novatel, 2015)

| PPP (-RTK) | Network RTK |
|-----------------------------------|----------------------------------|
| Global Coverage | Regional Coverage |
| One directional correction data | Bi-directional / individualised |
| Satellite Based Augmentation SBAS | Ground Based Augmentation GBAS |
| Sparse Network of Ref. Stations | Dense Network of Ref. Stations |
| ITRF coordinates (dynamic Europe) | Local coordinates (ETRS89) |
| Convergence Time for Accuracy | Almost Instantaneous Accuracy |
| Accuracy Sub decimetre / cm 2D | cm accurate 3D |
| Geostat. satellites / Internet | Mobile Internet for comm. |
| More suited for mass markets | More suited for specialists (3D) |
| Improving by more GNSS / freq. | Improving by more GNSS / freq. |

Figure 2.2: Comparison between PPP and NRTK (Henry, 2022)

2.2 Convergence Time

For different positioning techniques, the time to obtain a certain level of accuracy widely varies. As previously mentioned, for traditional PPP this convergence time is rather long. The long convergence time is caused by the estimation of real-valued constants for the ambiguity parameters (Ke et al., 2015).

PPP convergence can be defined as position estimates which reach a steady accuracy level. The convergence of PPP depends on the following factors (Abou-Galala et al, 2018):

- Number and geometry of observed satellites,
- Environment and dynamics of the receiver,
- Quality of observations and sampling interval.

Although PPP measurements need long convergence time to reach a constant value, this limitation can be exploited in various surveying and GIS applications for the diverse horizontal accuracy results (Abou-Galala et al, 2018).

2.3 Cadastral surveying and land administration

Land Administration is defined as the process of determining and recording the relationship between people and land. (<https://www.iso.org/obp/ui/iso:std:iso:19152:ed-1:v1:en>).

The International Organization for Standardization (ISO) TC211 published ISO 19152:2012 about the Land Administration Domain Model (LADM) to provide a shared conceptual information model for countries in establishing or improving Land Administration Systems and to better representing the relationship between people and land (and space) (Lemmen et al. 2013). The standard aims to establish a shared vocabulary and a common ontology for rights, restrictions and responsibilities (RRRs) affecting the land administration and its geometric components, enabling in this way the communication between related parties, organisations or countries.

First edition of the LADM is structured in four packages and one-subpackage, as presented below:

- party package, related to all the entities about different LA actors involved in a transaction;
- administrative package, containing information about a. the basic administrative unit, a concept adopted in LADM to arrange and combine spatial units with the same or homogeneous RRR and the rights, restrictions and responsibilities;
- spatial unit package, related to the different types of spatial units and their characteristics;
- surveying and representation sub-package, modelling the spatial dimensions of the legal objects, including the spatial sources;

LADM provides a common terminology for land administration, based on various national and international systems, that is as simple as possible in order to be adapted at international level. At the same time, it provides flexible modelling mechanism, so that the unique characteristics at national level can be modelled and represented. The terminology allows a shared description of different formal or informal practices and procedures in various jurisdictions. It provides a basis for national and regional profiles and enables the combining of land administration information from different sources in a coherent manner (<https://www.iso.org/obp/ui/iso:std:iso:19152:ed-1:v1:en>).

Since its vote as ISO standard in 2012, a significant number of country profiles based on LADM has been developed, integrating the legal and institutional context governing RRRs with the desired Land Administration Systems' (LASs) advancements. (Kalogianni et al., 2021). ISO standard are subject to periodic revision, typically in a 6 to 10 years cycle. The revision of LADM is currently ongoing, initiated by ISOTC211 and supported by OGC and other organisations (UNGGIM, World bank, FIG, IHO, ect). The new edition of the standard

will have a wider scope including information about valuation, spatial planning, as well as marine spaces. The review is needed in order to provide more complete tooling to improve tenure security and better land and property rights for all (van Oosterom, 2022). The new edition will be multipart, comprising of 6 parts, as follows:

- **Generic Conceptual Model.** This part provides the scope and definitions, a general overview of the model, its core classes and its individual packages and a more detailed overview of the `LASource` and `VersionedObject` classes,
- **Land Registration.** This part introduces the Land Registration Standard including a refined Survey and Representation package with a range of measurement techniques. It further includes updates around the functionality published in Edition 1 of the LADM,
- **Marine Space Georegulation.** This part is based on IHO's S121 standard: Maritime Limits and Boundaries,
- **Valuation Information.** This part specifies the characteristics and semantics of data in valuation registries maintained by public authorities,
- **Spatial Plan Information.** This includes planned land use (zoning) to be converted into rights, restrictions and responsibilities (RRR),
- **Implementations.** This part will address a range of topics needed for implementations of LADM: developing a country profile, modelling processes/workflows, and encodings.

"The first Edition of the LADM included a very simple survey model based on the ISO 19156:2011 Observations and Measurement Standard (OM), focusing on standardised modelling of information at the conceptual level, having a dedicated sub package for Spatial and Surveying representation. The ongoing LADM revision develops a refined survey model to support a broad range of surveying and data acquisition approaches and accuracies." (Kalogianni et al., 2022)

During the revision on the LADM, new attributes for the tools and methods of data acquisition techniques are introduced to better describe them. In the proposed new UML diagram there are new subclasses depicting specializations, detailing further information about spaces of parcels.

"The main refinement refers to the `LASpatialSource`, which includes new attributes compared to LADM Edition I and has two subclasses; the `LASurveySource` and the `LADesignSource`, supporting the modelling of information arising various data acquisition techniques (from single distance measurements to point-cloud and GNSS measurements) through the inclusion of new classes, attributes, code list values and associations." (Kalogianni et al., 2022).

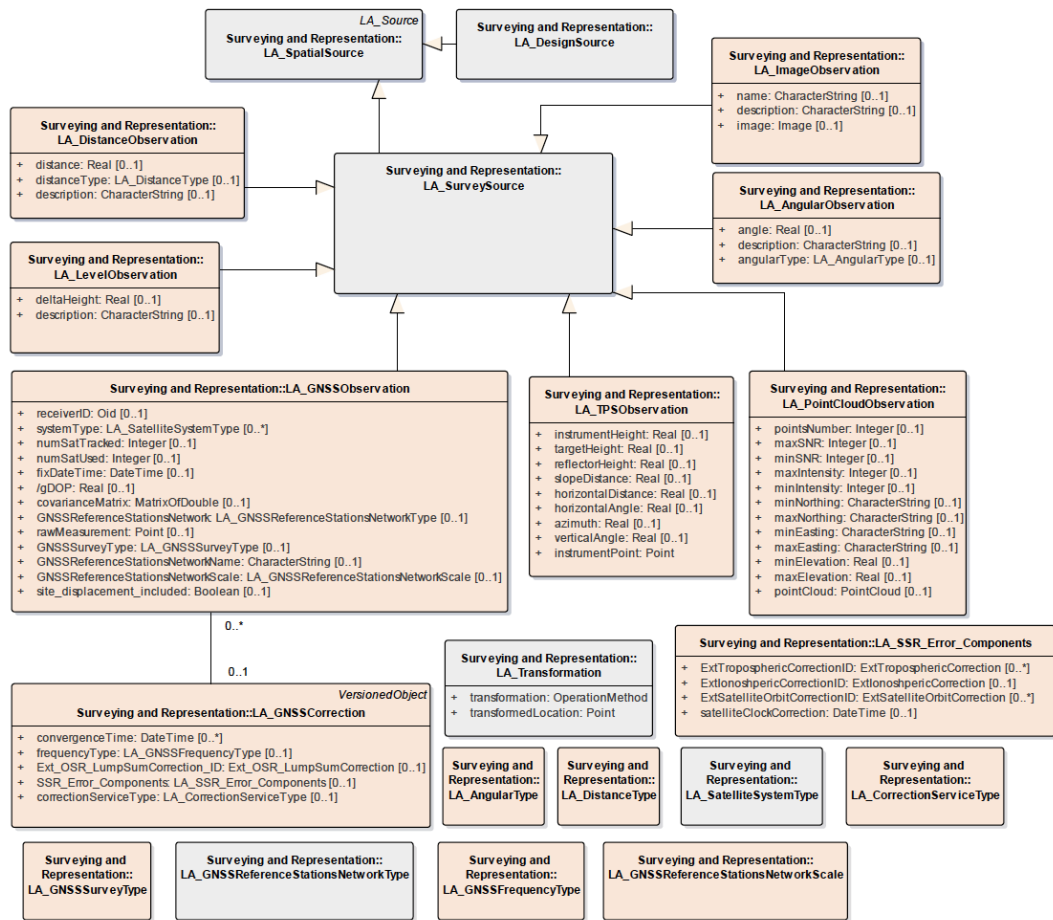


Figure 2.3: Proposed basic classes of the Surveying Representation sub-package (Kalogianni, et al., 2022)

The Galileo HAS is supported through the second edition of the standard through the new classes that have been added at the Surveying Representation sub-package of the Part 2 of the standard and specifically the class 'LAGNSSCorrections'.

2.4 GISCAD-OV

The work of the Synthesis project closely relates to the GISCAD-OV project: Galileo Improved Services for Cadastral Augmentation Development On-field Validation. The GISCAD-OV project is a cadastral surveying project executed by a consortium of 14 partners, funded by the EUSPA, under the Grant Agreement Number 870231. The main scope of the GISCAD-OV project is to design, develop and validate a cost-effective High Accuracy Service based on GPS and Galileo HAS, as well as Precise Point Positioning-Ambiguity Resolution (PPP-AR) quick convergence techniques. For the EUSPA project, pilot campaigns in several locations throughout Europe are executed. During these campaigns, observations are made in

environments with varying characteristics, some more urban and others more rural. Multiple constellations are used for positioning, while PPP-RTK and the Galileo High Accuracy Service are used for augmentation (European Global Navigation Satellite Systems Agency, 2019).



Figure 2.4: logo GISCAD-OV project (European Global Navigation Satellite Systems Agency, 2019)

Kalogianni et al. (2022) reflect on the synergy between the H2020 GISCAD-OV project and the revision of the ISO 19152 LADM, highlighting that main conclusions from the project concerning the support of Galileo HAS are included at the ISO, while the proposal to model and standardise the GNSS corrections, has been reviewed and approved by the international community that participates in the ISO meetings.

3 Methodology

3.1 Pilot campaigns in the GISCAD-OV project

One of the GISCAD-OV campaigns took place in an urban setup in Germany from 19-22 September, and therefore fitted with the timeline of the synthesis project. With the supervision of our mentor, the project team participated and acquired two sets of data from this pilot campaign, having the opportunity to test in real life the equipment and software tools to support Galileo HAS, which are currently being developed. Participating in the pilot campaign and validating the project with real-world datasets is very important for our project's reflection. The data collected in the field were analyzed and from each set of observations, precision will be computed, and subsequently the relative accuracy of HAS.

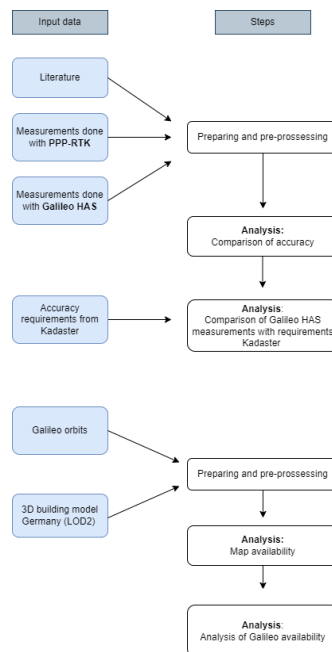


Figure 3.1: Steps of the methodology

3.1.1 Data Acquisition and Equipment

In Germany, two sites were visited. The first parcel was situated just outside of Olpe, in the grasslands. The sky was mainly open, and there was little blockage present. Except from the first two measurements, which were carried out next to the road that had a canopy.

3 Methodology

When the conditions were not good enough, in terms of limited satellite availability, the points were only measured with PPP-RTK. This was because the Galileo HAS would not give an accurate result in these conditions. The other points at this parcel were measured with both PPP-RTK and Galileo HAS. The datasets were gathered using the GPS-Galileo multiconstellation (9 GPS and 6 Galileo satellites during the gathering of the data).



Figure 3.2: Measurement Open Sky

The second parcel was situated in the town of Olpe. The parcel of a local surveying office was measured. One withdrawal of the usage of the PPP method in rural areas would be the multipath. The buildings in the neighbourhood were not very tall, as they were stand-alone residential houses, but still blocking the open sky. This means the buildings are potentially blocking satellite signals.

Regarding the equipment, the measurements were acquired using a GNSS receiver antenna. However, additional devices had to be attached to the main equipment in order to receive both the PPP-RTK and HAS corrections. The interface of the equipment thus used three additional devices:

- a receiver which applied the PPP-RTK corrections
- a receiver which gathered the raw data
- a decoder which applied the HAS corrections for the raw data

A limitation aspect of the Galileo HAS that has to be mentioned is that the points which were measured using Galileo HAS took longer due to the PPP method having longer convergence time than the PPP-RTK corrections. This is covered in section 4.3.



Figure 3.3: Measurement Urban Area

3.2 Visibility Analysis

The performance of GNSS is affected by a decreased satellite visibility caused by buildings blocking the satellite signals, also called its Lines of Sight (Kleijer et al. 2009). The performance of Galileo HAS can also be negatively influenced by a decreased satellite visibility. With limited satellite visibility, there will be larger positioning errors to correct (Lyu Gao, 2022).

A visibility analysis of the satellites in an urban environment was performed, to get an insight into the performance of Galileo HAS in different environments. The visibility analysis was performed for the parcel where the measurements were taken in Olpe, Germany. It was decided to perform the analysis at the same location where the Galileo HAS measurements were taken. Even though the visibility analysis could have been performed in any urban environment, the results could explain any poor measurement taken at the parcel.

The visibility analysis was performed in Python and visualized in MeshLab. The files used in the script are as follows:

- **TLE (Two-line element set) files for Galileo and GPS satellites:** TLE is a data format encoding a list of orbital elements of an Earth-orbiting object for a given point in time. The TLE files for Galileo and GPS were obtained for the day of measurements in Germany.
- **3D building model from Olpe:** A freely available 3D building model from Olpe in Level of Detail 2 (LoD2). It was obtained from the Geobasis from North Rhine-Westphalia (Geobasis NRW) provided by the German governmental district of Cologne. (Bezirksregierung Köln, 2020). The format of the building model tiles is CityGML. The

3 Methodology

tile LoD2_32_418_5653.1_NW.gml contained the area of the urban parcel and was therefore used. The horizontal accuracy of the model corresponds to that of the underlying building floor plan. The vertical accuracy is +/- 1 m.



Figure 3.4: Tile of interest visualized in FZKViewer

- **Height model from Olpe:** From the Geobasis NRW, a height model was used as well. The Digital Terrain Model (DTM) was used to derive the z-coordinates from the points of interest. The same tile as the one used for the 3D building model was downloaded.

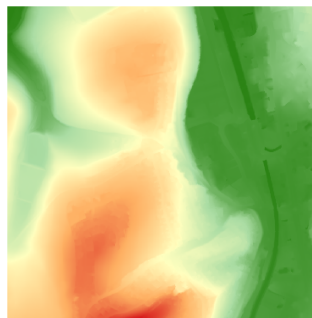


Figure 3.5: Tile of interest visualized in QGIS

The visibility analysis consists of two parts: The CRS conversion of the TLE files and the Line of Sight calculation and analysis. The steps are explained in the following subsections.

3.2.1 CRS conversion

As aforementioned, the TLE data format contains orbital elements necessary to compute the position of a satellite at a given time. Figure 3.6 shows its basic structure.

| Line 1 | |
|--------|---|
| Column | Description |
| 01 | Line Number of Element Data |
| 03-07 | Satellite Number |
| 08 | Classification (U=Unclassified) |
| 10-11 | International Designator (Last two digits of launch year) |
| 12-14 | International Designator (Launch number of the year) |
| 15-17 | International Designator (Piece of the launch) |
| 19-20 | Epoch Year (Last two digits of year) |
| 21-32 | Epoch (Day of the year and fractional portion of the day) |
| 34-43 | First Time Derivative of the Mean Motion |
| 45-52 | Second Time Derivative of Mean Motion (Leading decimal point assumed) |
| 54-61 | BSTAR drag term (Leading decimal point assumed) |
| 63 | Ephemeris type |
| 65-68 | Element number |
| 69 | Checksum (Modulo 10) (Letters, blanks, periods, plus signs = 0; minus signs = 1) |

| Line 2 | |
|--------|---|
| Column | Description |
| 01 | Line Number of Element Data |
| 03-07 | Satellite Number |
| 09-16 | Inclination [Degrees] |
| 18-25 | Right Ascension of the Ascending Node [Degrees] |
| 27-33 | Eccentricity (Leading decimal point assumed) |
| 35-42 | Argument of Perigee [Degrees] |
| 44-51 | Mean Anomaly [Degrees] |
| 53-63 | Mean Motion [Revs per day] |
| 64-68 | Revolution number at epoch [Revs] |
| 69 | Checksum (Modulo 10) |

Figure 3.6: Structure of the Two Line Element (CelesTrak, 2022)

From this information, the position of each GPS and Galileo satellite in a terrestrial reference system at the time of the survey in Olpe was computed through a series of coordinate transformations. Directly from the orbital elements coordinates are obtained in a True Equinox Mean Earth (TEME) reference system which is celestial and inertial. It differs from terrestrial reference systems which rotate with the earth. As such, the position of each satellite went through a chain of transformations shown in figure 3.6 which converted it from TEME to ETRS89, an Earth Centred Earth Focused system appropriate for representing terrestrial positions. Projecting ETRS89 coordinates to ETRS89/UTM32 (EPSG: 25832) would appear at a glance to be a straightforward pathway to obtaining satellite positions in the same coordinate system as that of the 3d city model of North-Rhine Westphalia which is later used

in performing line of sight tests. However, there is a challenge associated with it. In the resulting transformation, all z coordinates are treated as ellipsoidal height. This means that for each satellite, its ECEF height coordinate becomes positive after performing an ETRS89 to ETRS89/UTM32 transformation. Hence, all GPS and Galileo satellites will appear to be above the horizon, which is a false representation of the satellite geometry. For this reason, the choice to work with a geocentric ECEF frame, ETRF2000, was justified.

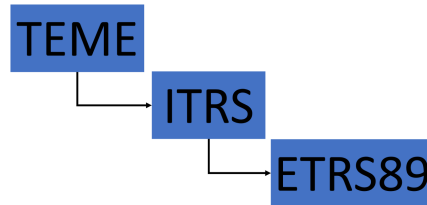


Figure 3.7: Chain of coordinate transformations performed

3.2.2 Line of Sight

For the calculation of the Lines of Sight and their possible blockage, some preparation steps are performed.

First of all, the 3D building model was converted to an object file using FME Workbench. It was decided to do this conversion because the .obj format fitted Python libraries to load the model into Python. The conversion was done by creating BRepSolids from the thematic surfaces of the LoD2 model. The solids were written into an .obj file. The workflow in FME Workbench is shown in the following figure. The object file was loaded into Python and converted to a triangle mesh.

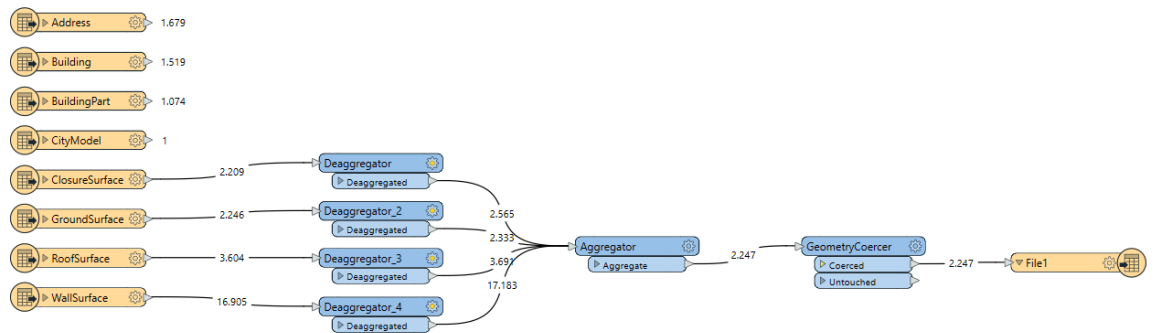


Figure 3.8: FME Workbench workflow for CityGML to .obj conversion

Therefore, a raster for the area of interest was created. The minimum and maximum x,y-coordinates of the raster were taken from the extents of the mesh. With a predetermined variable cellsize the raster filled up. The corresponding z-coordinates were taken from the DTM. From the coordinates of the points in the raster the lines to the Galileo and GPS satellites were created. Subsequently, for all lines there was checked if there was an intersection

with one of the triangles in the triangle mesh from the building model. Also, two big triangles were added underneath the mesh, so that the satellites that are currently underneath the Earth's surface are eliminated as well.

The following method was used to check if there is an intersection with one of the triangles. A Boolean method was chosen, because it was only desired to know if there is an intersection, and not at which exact point. This also makes the calculation less demanding.

You have a line between the points q_1 and q_2 , and a triangle between the points p_1, p_2, p_3 . If the sign of volume of the tetrahedron (q_1, p_1, p_2, p_3) equals the sign of volume (q_2, p_1, p_2, p_3) AND the sign of volume (q_1, q_2, p_1, p_2) equals sign of volume (q_1, q_2, p_2, p_3) and sign of volume (q_1, q_2, p_3, p_1) there is an intersection between the triangle (p_1, p_2, p_3) and the line (q_1, q_2) (Aftosmis, 1996). In the following figure, the last three volumes are all positive. If this condition is true, there is an intersection and the test function will return True.

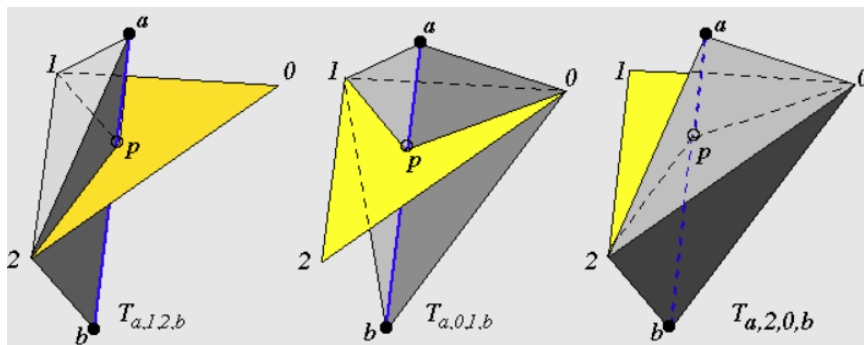


Figure 3.9: the sign of volume $(a,b,1,2)$, sign of volume $(a,b,0,1)$ and sign of volume $(a,b,0,2)$ are all positive (Aftosmis, 1996)

The lines that have an intersection with the mesh, are representing blocked satellite signals. The satellite is not visible from the point on the raster because its Line of Sight is blocked by a building.

3.3 Analysis of Datasets

In order to test the accuracy of the Galileo HA Service, the datasets acquired by the GISCAD-OV project were used to visualise the difference in accuracy between the PPP-RTK and the Galileo HAS correction methods.

The datasets were obtained from the GISCAD-OV repository after being given access to them in three formats.

- RINEX: PPP-RTK data
- Novatel log files: PPP-RTK data
- THD Box folder: contained a .nmea format file and .pos files with the PPP in which the PPP corrections were distributed
- .xlsx file containing the coordinates of the reference points

3.3.1 Coordinate transformation

The files from the THD Box folder contained coordinates converted from WGS84 to the ECEF coordinate system. The .nmea file contained the gathered data in WGS84 and the applied PPP correction data in ECEF coordinate system. The .pos files contain the most accurate positions of the dataset acquired using Galileo HAS.

A xlsx containing the coordinates of the control point that would be used for reference in comparing the correction methods was also provided by the GISCAD-OV team. The Novatel Log files were converted from .log format to .ASC file using the Novatel Converter program. The content from the files was filtered in order to obtain only the position coordinates to provide an accuracy comparison. The lines in the files from which the data was used were the BESTPOSA for the PPP-RTK method.

Example of BESTPOS line:

```
BESTPOSA,FILE,0,60.5,FINESTEERING,2228,140898.000,03000000,b1f6,15990;SOLCOMPUTED,
NARROWINT,51.02707841316,7.83756393580,353.0709,48.0000,WGS84,0.0137,0.0115,0.0198,
"0",1.000,0.000,34,27,27,23,00,20,3f,37*23f0ef2a
```


| Field | Field type | Description | Format | Binary Bytes | Binary Offset |
|-------|--------------|--|---------|--------------|---------------|
| 1 | Log header | BESTPOS header For information about log headers, see ASCII , Abbreviated ASCII or Binary . | | H | 0 |
| 2 | sol stat | Solution status, see Table: Solution Status | Enum | 4 | H |
| 3 | pos type | Position type, see Table: Position or Velocity Type | Enum | 4 | H+4 |
| 4 | lat | Latitude (degrees) | Double | 8 | H+8 |
| 5 | lon | Longitude (degrees) | Double | 8 | H+16 |
| 6 | hgt | Height above mean sea level (meters) | Double | 8 | H+24 |
| 7 | undulation | Undulation - the relationship between the geoid and the ellipsoid (m) of the chosen datum  When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84. | Float | 4 | H+32 |
| 8 | datum id# | Datum ID number 61 = WGS84 63 = USER | Enum | 4 | H+36 |
| 9 | lat σ | Latitude standard deviation (m) | Float | 4 | H+40 |
| 10 | lon σ | Longitude standard deviation (m) | Float | 4 | H+44 |
| 11 | hgt σ | Height standard deviation (m) | Float | 4 | H+48 |
| 12 | stn id | Base station ID | Char[4] | 4 | H+52 |
| 13 | diff_age | Differential age in seconds | Float | 4 | H+56 |
| 14 | sol_age | Solution age in seconds | Float | 4 | H+60 |

Figure 3.10: Explanations of the message segments

For the Galileo HAS data, the .pos files given by GISCAD-OV were used. The set of coordinates was read and visualised using libraries from the Python programming language, mainly pandas, numpy and matplotlib.

The filtered coordinates were introduced in text files and then in order to make the files easier to plot in python using the pandas library, they were converted to .csv format using Microsoft Excel.

The datasets had to be represented in the projected coordinate system used in Germany (EPSG: 25832) rather than the decimal degrees format (PPP-RTK) or the ECEF converted coordinates (Galileo HAS) from the input files. It has to be mentioned that when converting from ECEF to epsg:25382, the transformed height will take into consideration the height of the instrument and the undulation of the surface. The transformation function was created using transformations commands from the pyproj library to convert both the obtained datasets and the coordinates of the reference point.

The analysis was further carried by calculating the root mean square error for each PPP-RTK and Galileo HAS dataset in reference to the given coordinates of the geodetic control points and by calculation the standard deviation of each Galileo HAS dataset.

$$RMSE = \sqrt{\left(\frac{\sum (X_{obs} - X_{pred})^2}{n}\right)} \quad (3.1)$$

Where the parameters are represented by:

- X_{obs} - the coordinates of the observations from the acquired datasets
- X_{pred} - the coordinates of the reference points

$$StD = \sqrt{\left(\frac{\sum (X_{obs} - X_{mean})^2}{n}\right)} \quad (3.2)$$

Where the parameters are represented by:

- X_{obs} - the coordinates of the observations from the acquired datasets
- X_{mean} - the mean of the dataset

3.3.2 Convergence time analysis

While examining the accuracy of the two different techniques we observed that a certain level of accuracy is achieved within a specific time interval for each case. Convergence time is the time that it takes, so as the accuracy, for a set of measurements of a point, to become stable and reach a specific number. The process is named like this because a couple of minutes are needed so as the errors of the position (x-axis, y-axis) of a point to “converge” to a specific value which is usually closer to zero with the passing time, meaning closer to the real coordinates of the point. Below, it is presented the comparison between the convergence time of the PPP technique and the PPP-RTK technique. Also, it is known that the Galileo HAS is a PPP technique, so the PPP convergence time is a simulation of Galileo HAS measurements.

Convergence time varies depending on the state of GNSS constellation, the level of multi-threading, the proximity of obstacles such as high trees and buildings, and the calculation algorithms used by the receiver (Siejka, 2018). In our case the software that was used to process and visualize our data was the “rtkplot”. The data were collected at the area of

3 Methodology

Delft with both PPP and PPP-RTK methods for the purpose of the project. For the PPP measurements we chose one out of the five NMEA files that were available and for the PPP-RTK measurements we used twenty out of five hundreds LOG files that existed in total, but first we converted them to ASC format through the “NovAtel Convert 2.6.7” software, so as to be readable by the “rtkplot”. After we calculated the results and visualized them, we calculated the average convergence time regarding the two methods. The results, figures and the statistics table is available at the ‘Results’ section.

4 Results

4.1 Datasets GISCAD-OV

Note: For this research, datasets from a pilot campaigns of the GISCAD-OV project were used. The paragraphs in this section refer to results (plots) that are not presented in this part of report due to confidentiality. For further information regarding the data and the results, contact the authors.

For plotting and calculating the Root Mean Square Error, the measurements had to be filtered. For the PPP-RTK observations, the ones that contained the narrow integer solution, as stated on the Novatel website, were selected as they provided the best position. However, the datasets for the Galileo HAS observations contained also the points which were not converged. The points from the datasets were thus filtered using the 7 decimeter and 2 decimeter thresholds on the X and Y coordinates and counted, with further mention that the error values on the vertical coordinate Z are of meter level.

According to the error analysis, the Galileo HAS method presents larger errors compared to those of the PPP-RTK datasets. Additional analysis was done by counting the number of plotted points. As it can be observed, the errors on the 19th of September were too large to be calculated. The plot on the 20th of September was done using 237 points out of 2355, out of which 29 presented errors under 2 decimeters. The plot on the 21th of September has a larger dataset, with 1499 out of 5999 points being utilised and 418 having the errors under 2 decimeters, thus showing that the dataset on the 21st of September has the best quality out of the three datasets.

| Date | X (m) | Y (m) | Z (m) |
|-------|-------|-------|-------|
| 19/09 | 0.009 | 0.009 | 0.016 |
| 20/09 | 0.013 | 0.016 | 0.021 |
| 21/09 | 0.005 | 0.003 | 0.003 |

Table 4.1: Table containing the RMSE values of the PPP-RTK data

| Date | X (m) | Y (m) | Z (m) |
|-------|--------|--------|--------|
| 19/09 | 5.555 | 1.56 | 10.078 |
| 20/09 | 10.807 | 19.807 | 7.613 |
| 21/09 | 0.384 | 1.86 | 1.577 |

Table 4.2: Table containing the standard deviation values of the PPP Galileo HAS data

Note: The standard deviation is calculated using the total numbers of observations from the datasets

4 Results

| Date | Total number of points | Points within 0.7m error | Points within 0.2m error |
|-------|------------------------|--------------------------|--------------------------|
| 19/09 | 3194 | 0 | 0 |
| 20/09 | 2355 | 237 | 29 |
| 21/09 | 5999 | 1499 | 418 |

Table 4.3: Table containing the used number of points to calculate the RMSE of the PPP Galileo HAS data

| Date | X (m) | Y (m) | Z (m) |
|-------|------------|---------|-----------|
| 19/09 | errors too | high to | calculate |
| 20/09 | 0.118 | 0.054 | 4.247 |
| 21/09 | 0.126 | 0.136 | 5.792 |

Table 4.4: Table containing the RMSE values of the PPP Galileo HAS data

The data from the 19th of September has more outliers and larger errors in the Galileo HAS dataset than the data gathered on the 20th and 21st of September, while the PPP-RTK method presents the expected accuracy. The position of the point is in a rural area, where the satellite signal would be disturbed by buildings and cars passing by.

The point measured on the 20th of September is positioned in an open area. The HAS gathered observations present considerable differences compared to the PPP-RTK method. It is stated in table 4.3 that only 29 out of 2599 observations present the proposed 2 dm accuracy of Galileo HAS. It has to be mentioned that based on the number of used points and on the high values of the standard deviation from table 4.2, the measurements on the 20th of September have not been converged, even after the indicated time.

The data from the 21th of September seems to present the most accurate plot as the Galileo HAS measurements are close to the PPP-RTK dataset and the control point. The data is gathered at the outskirts of Olpe, in an empty area, close to the road.

The environment of an open sky seems to present the better positioning accuracy results for both the methods, as seen in the plots. The points within the 2 dm threshold are united using a line. Another important aspect that has to be taken into consideration when analysing the Galileo HAS datasets is if the used dataset is converged and its convergence time, as seen for the measurements from the 20th of September.

As a conclusion regarding the comparison between the datasets, Galileo HAS is quite unpredictable and presents bigger errors than the PPP-RTK method. The elongated shapes of the plotted datasets which display in an orbital way around the reference point (as seen especially in the plot for point 20) could be results of the longer convergence time which is needed for the PPP method to reach a constant position (further discussed in the Convergence Time analysis section). It also has to be mentioned that various other factors could influence the data acquisition negatively, such as multipath, human error, log error, geometry of the satellite, atmospheric errors etc.

As a further notice and another aspect of the conclusion is that with the limited number of datasets and time, a further detailed analysis could not be carried and the results presented in this section might not be representable for the Galileo HA service.

4.2 Visibility analysis

4.2.1 Results from the script

For the visibility analysis, it turned out filling up a raster with visible satellite values was too computationally expensive. Therefore, the visibility analysis was done for two test points in the area of interest. The resulting non-blocked lines were written to an .obj file. The resulting Lines of Sight of non-blocked satellite signals could, together with the triangle mesh of the 3D building model, be visualized in MeshLab. A few of the non-blocked lines that were written to the obj file were pointing downwards. Even though two triangles were added underneath the 3D building model mesh. It is possible that these lines did not intersect with the bigger triangles, and the bigger triangles should have had a better coverage. These lines were manually deleted.

The test points were selected on both sides of the building, to be able to see the effect of the presence of an obstacle on both sides. The coordinates of the test points are as follows:

Coordinates of the first test point:

$x = 418521.0, y = 5653473.0, z = 346.42(+1.8)$

Coordinates of the second test point:

$x = 418483.00, y = 5653482.72, z = 352.77$

For both test points, the resulting visible satellites are written down in the following tables. The lines are plotted in Meshlab. The visible Galileo satellite signals are shown in blue and the GPS lines are shown in blue.

Table 4.5: Visible satellites from test point 1

| GPS | Galileo |
|----------------------|--------------------|
| GPS BIIF-4 (PRN 27) | GSAT0215 (PRN E21) |
| GPS BIIF-9 (PRN 26) | GSAT0220 (PRN E13) |
| GPS BIII-2 (PRN 18) | |
| GPS BIIR-8 (PRN 16) | |
| GPS BIIRM-5 (PRN 29) | |

Table 4.6: Visible satellites from test point 2

| GPS | Galileo |
|----------------------|--------------------|
| GPS BIIF-4 (PRN 27) | GSAT0215 (PRN E21) |
| GPS BIIF-7 (PRN 09) | GSAT0220 (PRN E13) |
| GPS BIIF-9 (PRN 26) | GSAT0221 (PRN E15) |
| GPS BIII-2 (PRN 18) | GSAT0213 (PRN E04) |
| GPS BIIR-8 (PRN 16) | GSAT0210 (PRN E01) |
| GPS BIIRM-5 (PRN 29) | GSAT0209 (PRN E09) |
| GPS BIIRM-6 (PRN 07) | GSAT0203 (PRN E26) |
| GPS BIIRM-8 (PRN 05) | GSAT0201 (PRN E18) |

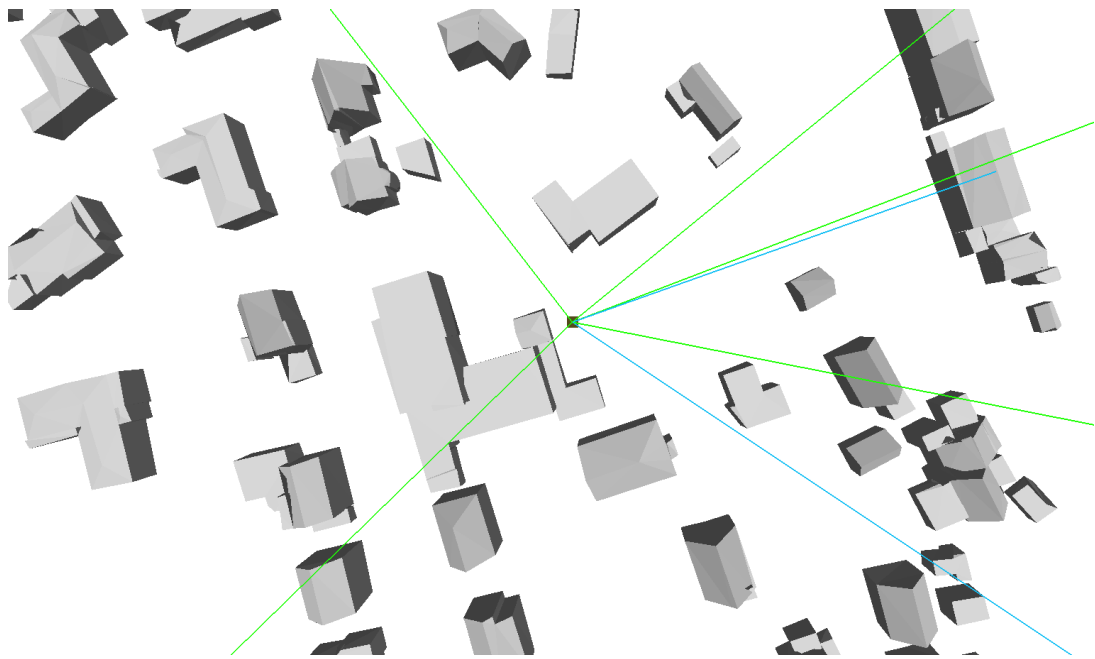


Figure 4.1: Visible GPS+Galileo satellites from test point 1 — top view

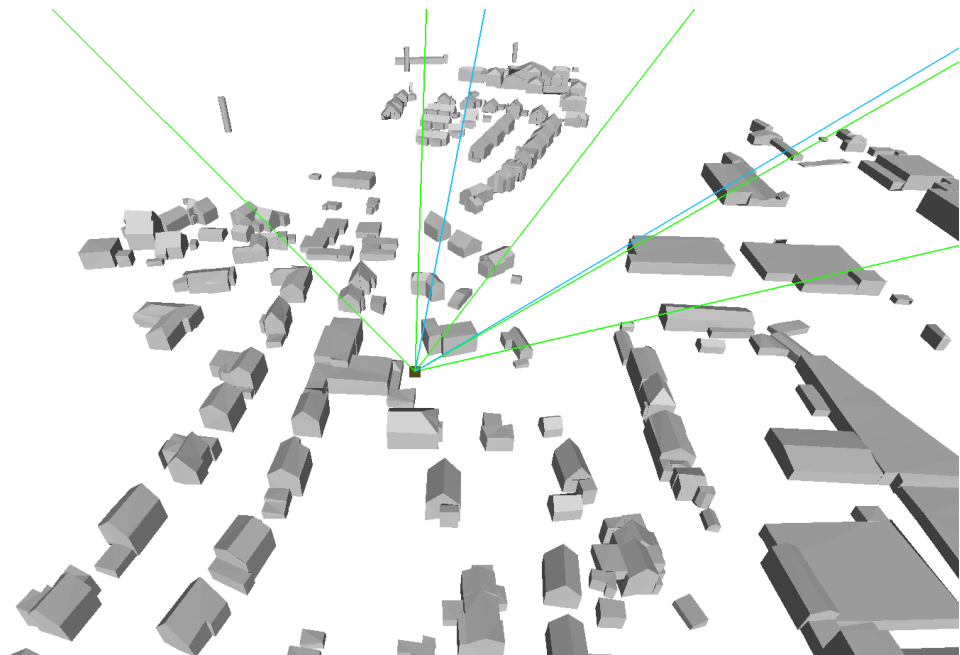


Figure 4.2: Visible GPS+Galileo satellites from test point 1 — top side view

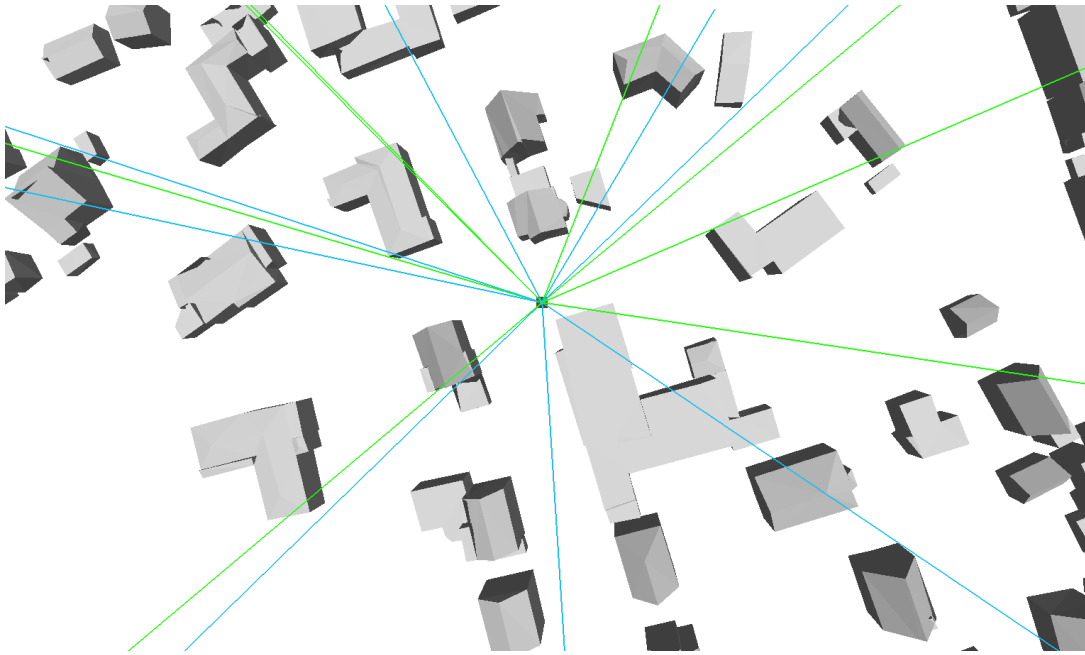


Figure 4.3: Visible GPS+Galileo satellites from test point 2 — top view

4.2.2 Validation

From satellite orbital information encoded in TLE sets, we were able to predict positions of Galileo and GPS satellites at a date and time when a particular test point was surveyed during the field campaign in Olpe. To validate the accuracy of our satellite geometry reconstruction, there was a need to make a comparison with another tool capable of completing the same task. We used the Trimble GNSS Planning Service Online for this purpose. In the validation, the coordinates of a survey point were used to generate a skyplot containing the same two constellations using the Trimble mission planning tool. A date and time of 20 September 2022 and 15:00, 15° elevation cutoff and height above ground of 1.8m were used as parameters. As the generated skyplot in figure 4.15 shows, the satellites Galileo E04, E05, E09, E31, E34 and E36, and GPS G02, G05, G07, G09, G11, G13, G20 and G30 meet the parameters specified above. Our findings agree to a lesser extent, as we only identify E05 and E34 for Galileo, and G02, G13 and G20 for GPS as subsets of satellites above the horizon for each of the two constellations. We also identify Galileo E21 and GPS G16, but they are below the 15° cut-off according to the Trimble service.

Even though the results of the visibility analysis do not fully align with the theoretical visible satellites from the Trimble GNSS Planner, it was noticed that several lines were eliminated because they were blocked. Also, the resulting visible satellites are of a smaller quantity than the quantity of theoretically visible satellites (without obstacles). This means that the satellite signals are blocked by the buildings. The performance can be negatively influenced by this decreased satellite visibility. With limited satellite visibility, there will be larger positioning errors to correct (Lyu Gao, 2022). Also if the satellites are not evenly spread across the sky, there are larger error in positional fix. For example for test point 1, there are no satellites visible in the West.

4 Results

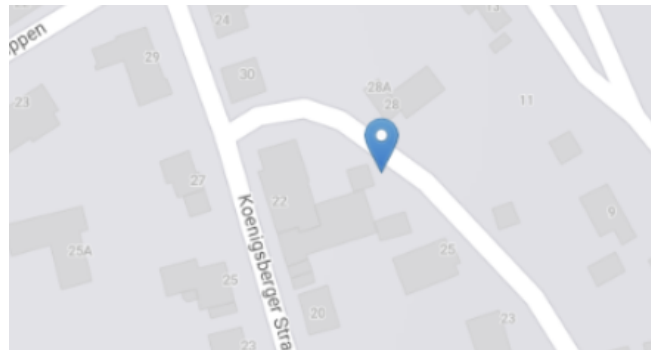


Figure 4.4: Validation location 1

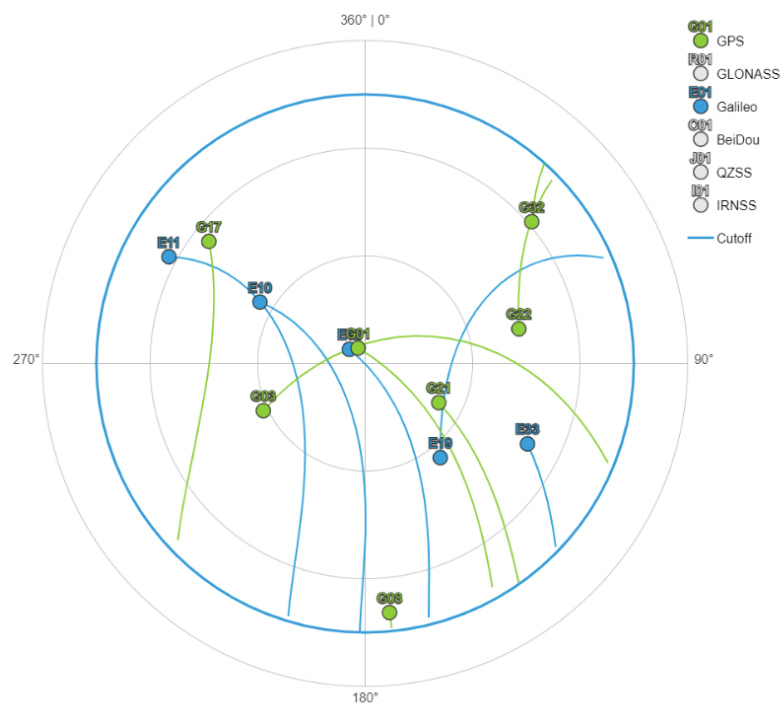


Figure 4.5: Satellite skyplot from location 1 (GPS + Galileo)

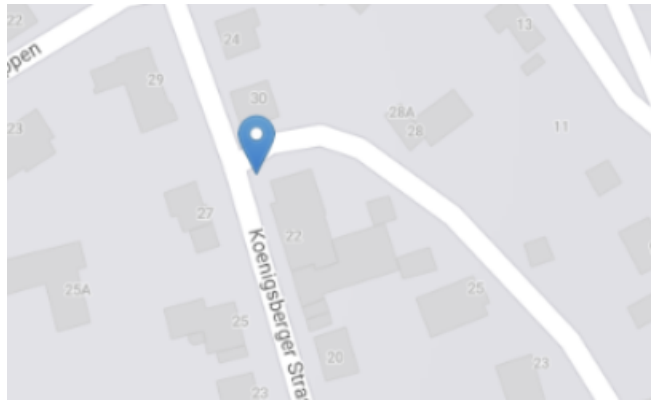


Figure 4.6: Validation location 2



Figure 4.7: Satellite skyplot from location 2 (GPS + Galileo)

4.3 Convergence time analysis

The two images below summarize and visualize in detail the results of the convergence time of the two data samples, giving us information regarding the time interval of the measurements and the error for each specific moment in time. The figure 4.8 represents the PPP measurements and the figure 4.10 represents the PPP-RTK measurements:

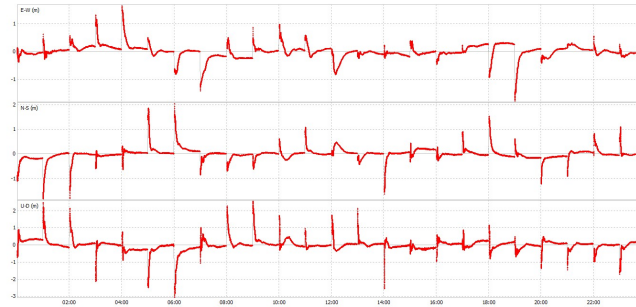


Figure 4.8: PPP-Convergence Time Diagram

| | | | | | | | | | | | | |
|-----------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| convergence time in minutes | 00.00-01.00 | 01.00-02.00 | 02.00-03.00 | 03.00-04.00 | 04.00-05.00 | 05.00-06.00 | 06.00-07.00 | 07.00-08.00 | 08.00-09.00 | 09.00-10.00 | 10.00-11.00 | 11.00-12.00 |
| E-W | 20 | 25 | 20 | 15 | 20 | 25 | 25 | 25 | 20 | 20 | 30 | 25 |
| N-S | 15 | 30 | 15 | 10 | 15 | 25 | 30 | 20 | 20 | 20 | 35 | 20 |
| U-D | 10 | 25 | 20 | 10 | 15 | 30 | 30 | 25 | 25 | 25 | 35 | 20 |
| convergence time in minutes | 12.00-13.00 | 13.00-14.00 | 14.00-15.00 | 15.00-16.00 | 16.00-17.00 | 17.00-18.00 | 18.00-19.00 | 19.00-20.00 | 20.00-21.00 | 21.00-22.00 | 22.00-23.00 | 23.00-24.00 |
| E-W | 30 | 15 | 10 | 10 | 10 | 15 | 15 | 20 | 25 | 30 | 20 | 20 |
| N-S | 35 | 20 | 15 | 15 | 15 | 25 | 25 | 15 | 20 | 15 | 20 | 10 |
| U-D | 35 | 20 | 15 | 15 | 15 | 25 | 25 | 20 | 10 | 20 | 15 | 20 |
| Average Convergence Time | E-W | N-S | U-D | | | | | | | | | |
| | 21.3158 | 20.2273 | 21.8421 | | | | | | | | | |

Figure 4.9: Statistics Table

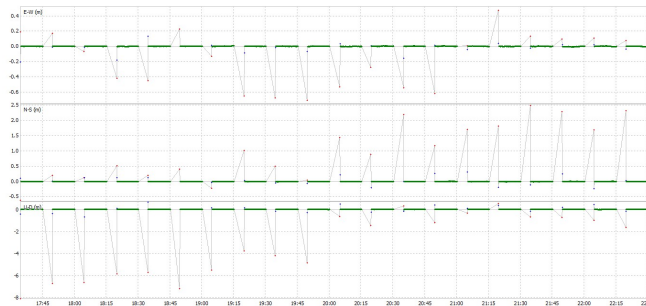


Figure 4.10: PPP-RTK-Convergence Time Diagram

In figure 4.8 is pictured 24 sessions of measurements and the time interval for every session is 1 hour. Within each of these 1-hour sessions, we observe that after a certain moment in time the position error gets below a certain threshold. The highest error that we encounter at the E-W axis is about 2 meters and at the N-S axis is also about 2 meters. Every graph consists of several points, which are extremely dense and each of these points represents a single measurement. The fact that these points have red color is because of the data quality indicator, which is 1 in our case meaning that is converging PPP. The threshold that we set

for the E-W and N-S axis is 20cm. In the statistics table below (figure 4.9) is presented the convergence time for each of the measurement sets and with red color are the measurement sets that they failed to converge to a position error below the threshold that we decided. Moreover the average convergence time is presented, which is about 21 minutes.

Regarding figure 4.10 we see that 5 hours of measurements are pictured and the time interval for every new measurement set is 15 minutes. Within each of these 15-minutes sessions, we observe again as previously, that after a certain moment in time the position error gets below a certain threshold, but in the case of PPP-RTK this happens almost instantly. The highest error that we encounter at the E-W axis is about 0.65 meters and at the N-S axis is about 2.5 meters. Moreover, we see that from all the measurements there is only one point with data quality indicator equal to 1, a couple of points that are in blue which means equal to 2 (converging PPP) and the rest the points are in green which means RTK fixed ambiguity solution and the indicator is equal to 4. For the PPP-RTK analysis, we only used the positions with a RTK fixed quality. So, after only a few measurements the PPP-RTK method converges. Moreover the threshold that we set for the E-W axis is 5cm and for the N-S axis is also 5cm, which are both achieved by all the measurement sets.

Another interesting element that we extract by observing these convergence diagrams is the value of error that each measurement is stabilized at. The PPP-RTK converges faster, and the accuracy is better (lower error). Therefore, we see that for the N-S axis the error is almost 0m, but at the case of PPP the error is around the value of 0.2m. For the E-W axis PPP gives errors between 0 and 0.2m, and at the case PPP-RTK the error is again almost 0m.

5 Conclusion

5.1 Performance of Galileo HAS

EUSPA promise to deliver a position accuracy within 2 dm through the Galileo High Accuracy Service. Being free of charge, transmitting corrections via satellite and requiring no additional infrastructure apart from a GNSS receiver are cornerstone benefits of HAS. Potentially, the GNSS market could be transformed in one way or another to cater for different user groups. National Mapping and Cadastral Agencies such as Kadaster are major stakeholders in the GNSS value chain. Currently, they achieve 2-3cm accuracy using a system of continuously measuring reference receivers, also known as NRTK. As HAS employs a different positioning technique, PPP, there was need to evaluate its performance against the well-known and current way of working. This would give an impression of the usability of HAS to perform Cadastral Surveying.

In this project, evaluation was a 3-step process carried out by making simultaneous PPP-RTK and PPP field observations, comparing their accuracy and time to first fix, and determining the number of Galileo satellites in view from an observation point since corrections will be provided through the E6B signal. In our findings, the results for Galileo HAS were not as accurate as expected and showed inadequate performance for Cadastral Surveying. However that could also be due to factors that do not depend on the positioning service, such as equipment or human error, so these results might not be representative for the Galileo HA service. In contrast to those for PPP-RTK, its accuracy was lower and time to first fix much greater. In addition, our research shows that only a subset of Galileo and GPS satellites have an unobstructed path between themselves and the two test points which were surveyed. Some satellite signals were blocked by the 3D building model triangle mesh. This leads to the answer to the research question of the Synthesis project:

To what extent is, the performance of Galileo High Accuracy Service, suitable for cadastral surveying?

Galileo HAS does not meet the required accuracy (in terms of 2-3 cm) for cadastral surveying. It only factors in orbital corrections, hence it's limited to an accuracy of about 2 dm. Longer convergence time in comparison to PPP-RTK also makes HAS inefficient for surveying. In the built environment, the accuracy and convergence time of Galileo HAS is also negatively influenced because of blocked signals. However, this counts for all positioning services in the built environment.

The required accuracy of about 2-3 cm for cadastral surveying applies to most developed countries that have a well-established land administration system. However, in countries where property rights have not been registered leading to insecure land tenure, Galileo HAS could be used for cadastral surveying and its accuracy of about 2 dm, can be sufficient given those circumstances and taking into account that it is free of charge. What is more, Galileo HAS can be used in other GNSS applications, such as Intelligent Transportation Systems (ITS), unmanned vehicles and drones, and augmented reality. Galileo HAS can also be used

5 Conclusion

as a positioning service in remote areas, such as forests, due to the PPP correction method in which only satellite signal to perform the observations is needed (EUSPA, 2020).

5.2 Limitations

Due to the limited number of datasets used for the analysis and the technical problems encountered while gathering and obtaining the data, time and experience, the presented results might not represent the true performance of the Galileo High Accuracy Service.

Bibliography

- Abou-Galala, M. et al. (2018) *Assessment of the accuracy and convergence period of Precise Point Positioning*. Retrieved October 21, 2022, from <https://www.sciencedirect.com/science/article/pii/S1110016817301795>
- Aftosmis, M. (1996). *Intersection of Generally Positioned Polygons in R^3* . Retrieved October 22, 2022, from https://www.nas.nasa.gov/publications/software/docs/cart3d/pages/bool_intersection.html
- BESTPOS. (2022). *Log explanations*. <https://docs.novatel.com/OEM7/Content/Logs/BESTPOS.html>
- Bezirksregierung Köln. (2020). *Geobasis NRW - Produkte und Dienstleistungen*. Retrieved November 1, 2022, from https://www.bezreg-koeln.nrw.de/brk_internet/geobasis/index.html
- Capua R., Antonelli, D. Nudiens, I. *OV-GISCAD project - Datasets for both PPP-RTK and Galileo HAS*
- CelesTrak. (2022). *NORAD Two-Line Element Set Format*. Retrieved October 12, 2022, from <https://celestrak.org/NORAD/documentation/tle-fmt.php>
- emlid.com. (n.d.). *Understanding PPK solution and analyzing logs from Reach — RTK Modules*. Available at: <https://docs.emlid.com/reach/tutorials/post-processing-workflow/analyzing-logs/>
- EUSPA. (2021, May 23). *Galileo High Accuracy Service (HAS)*.
- European Global Navigation Satellite Systems Agency. (2019). *GISCAD-OV User Requirements Document*.
- European GSA (2019) *PPP-RTK market and technology report*
- GPGGA, (n.d.). *GPS fix data and undulation* Retrieved October 21, 2022, from <https://docs.novatel.com/OEM7/Content/Logs/GPGGA.htmGPSQualityIndicators>
- Grinter, T. Roberts, G. (2011) *Precise Point Positioning: Where are we now?*. Retrieved October 18, 2022, from https://www.spatial.nsw.gov.au/_data/assets/pdf_file/0020/165701/2011-Grinter_and_Roberts_IGNSS2011_PPP_where_are_we_now.pdf
- Henry, J. (2022). *GNSS High-accuracy positioning*.
- Huisman, L., Hoentjen, K., Vet, A., Buist, P. (2020). *Galileo-Only Cadastral Survey*.
- Kalogianni, E., Grulel, H., Bar-Moar, A., Harold, B., Lemmon, T., Lemmen, C., van Oosterom, P.J.M. (2022). *Investigating the Requirements for the ISO 19152 LADM Survey Encodings*
- Ke, M., Lv, J., Chang, J., Dai, W., Tong, K., Zhu, M. (2015, November 30). *Integrating GPS and LEO to accelerate convergence time of precise point positioning*. <https://doi.org/10.1109/WCSP.2015.7341230>

Bibliography

Kleijer, F., Odijk, D., Verbree, E. (2009). *Prediction of GNSS availability and accuracy in urban environments – case study schiphol airport*. https://doi.org/10.1007/978-3-540-87393-8_23

Lyu, Z., Gao, Y.. (2022). *Precise Point Positioning in Urban Environments*. <https://insidegnss.com/precise-point-positioning-in-urban-environments/>

Geographic information — Land Administration Domain Model (LADM).

<https://www.iso.org/obp/ui/iso:std:iso:19152:ed-1:v1:en>

Naciri, N., Yi, D., Bisnath, S., Javier De Blas, F., Capua, R. (2022). *Assessment of Galileo High Accuracy Service (HAS) test signals and preliminary positioning performance*.

Naciri, N., Yi, D., Bisnath, S., de Blas, F.J. and Capua, R. (2022). Validation of a European High Accuracy GNSS Service for Cadastral Surveying Applications. ION GNSS+, The International Technical Meeting of the Satellite Division of The Institute of Navigation. [online] doi:10.33012/2022.18458.

Novatel (2015) *An introduction to GNSS - Chapter 5*

Novatel website *BESTPOS - Best Position*

<https://docs.novatel.com/OEM7/Content/Logs/BESTPOS.htm>

Siejka, Z. (2018) *Validation of the Accuracy and Convergence Time of Real Time Kinematic Results Using a Single Galileo Navigation System*. *Sensors*, 18(8), p.2412. doi:10.3390/s18082412.

Appendix I: Algorithms

1 Visibility Analysis

1.1 Line of Sight

```
import os
from sgp4.api import Satrec
from sgp4.api import jday
from astropy.coordinates import TEME, CartesianDifferential, CartesianRepresentation
from astropy import units as u
from astropy.time import Time
from astropy.coordinates import ITRS, EarthLocation
from pyproj import Transformer
from tle import tle_json

def tle_to_itrs(l1,l2,day=20,month=9,year=2022,hour=15,minute=0,second=0):

    satellite = Satrec.twoline2rv(l1, l2)
    jd, fr = jday(year,month,day,hour,minute,second)
    error, teme_position, teme_velocity = satellite.sgp4(jd,fr)

    # transform teme_position vector to True Equator Mean Equinox object
    teme_position = CartesianRepresentation(teme_position*u.km)
    teme_velocity = CartesianDifferential(teme_velocity*u.km/u.s)
    teme = TEME(teme_position.with_differentials(teme_velocity),obstime=Time(jd,format='jd'))
    #print(teme)

    # transform teme object to itrs geocentric coordinates
    itrs = teme.transform_to(ITRS(obstime=Time(jd,format='jd')))
    pos = [itrs.earth_location.geocentric[0].value*1000,itrs.earth_location.geocentric[1].value*1000,itrs.earth_location.geocentric[2].value*1000]
    #print(pos)
    #position_itrs = (itrs.earth_location.geodetic.lon.value,itrs.earth_location.geodetic.lat.value,itrs.earth_location.geodetic.height.value*1000)
    #print(itrs.earth_location.geodetic.lon)
    return pos#position_itrs

def convert_crs(from_crs,to_crs,pos_vector):

    transformer = Transformer.from_crs(from_crs,to_crs,True)
    position = transformer.transform(pos_vector[0],pos_vector[1],pos_vector[2])
    return position
```

Figure 1

Appendix I: Algorithms

```
def german3d(pos_vector):

    transformer = Transformer.from_pipeline(''+proj=pipeline
                                           +step +proj=axisswap +order=2,1
                                           +step +proj=unitconvert +xy_in=deg +z_in=m +xy_out=rad +z_out=m
                                           +step +proj=utm +zone=32 +ellps=GRS80'')

    position = transformer.transform(pos_vector[0],pos_vector[1],pos_vector[2])
    #print(position)
    return position

def sat_pos():

    print(os.getcwd())
    gps_path = '../data/gps.txt'
    galileo_path = '../data/galileo.txt'
    print('Reading TLE')
    satellites = [tle_json(gps_path),tle_json(galileo_path)]

    print('From TEME Compute Satellite Position in ETRS89/UTM32')
    for i in satellites:
        for tle in i:

            two_lines = list(tle.values())
            key = list(tle.keys())[0]

            s = two_lines[0][0]
            t = two_lines[0][1]
            #yr,mon,day,hr,minute,sec = 2022,9,20,13,0,0

            # get satellite position in itr2014
            position_itrs_2k14 = tle_to_itrs(s,t)
            print('ITRF2014:\t{}'.format(position_itrs_2k14))

            # get satellite position in itr2000
            #position_itrs_2k = convert_crs(7789,7909,position_itrs_2k14)
            position_itrs_2k = convert_crs(7789,4919, position_itrs_2k14)
            print('ITRF2000:\t{}'.format(position_itrs_2k))

            # get satellite position in etrf2000
            position_etrs = convert_crs(4919,7930, position_itrs_2k)
            print('ETRF2000:\t{}'.format(position_etrs))

            # convert to 3d crs
            position_de = convert_crs(7930,32632,position_etrs)
            print('ETRS89/UTM32:\t{}'.format(position_de))

            tle[key].append(position_etrs)
```

Figure 2

```

idx = 1

point = convert_crs(32632,7930,[418521.00, 5653473.00, 346.42])
#point = convert_crs(4979,4919,wgs84)
...

with open('../data/sat.txt','w') as sat:
    sat.write('id|wkt\n')
    for i in satellites[1]:
        v = [(i[list(i.keys())[0]][2][0] - p[0])*0.2, (i[list(i.keys())[0]][2][1] - p[1])*0.2, (i[list(i.keys())[0]][2][2] - p[2])]
        sat.write('{}|LINESTRING(418521.00 5653473.00 346.42,{} {} {})\n'.format(idx,v[0],v[1],v[2]))
        idx += 1
    ...

with open('../data/sat.obj','w') as lines:
    for i in satellites[1]:
        v = [(i[list(i.keys())[0]][2][0]-point[0]), (i[list(i.keys())[0]][2][1]-point[1]), (i[list(i.keys())[0]][2][2]-point[2])]
        #print(v)
        lines.write('v {} {} {} \n'.format(round(v[0],2),round(v[1],2),round(v[2],2)))
    lines.write('v {} {} {} \n'.format(point[0],point[1],point[2]))
    for i in range(1,29):
        lines.write('1 29 {} \n'.format(i))

return satellites

if __name__ == "__main__":
    sat_pos()

```

Figure 3

1.2 Visibility of satellites


```

import numpy as np
import open3d as o3d
import math
import sympy
from sympy import Point3D
from sympy.abc import L
from sympy.geometry import Line3D, Segment3D, Plane, Polygon

from los import sat_pos, convert_crs, tle_to_itrs
from tle import tle_json

def create_triangle_mesh(input_file):
    #Read triangle mesh from obj file
    mesh = o3d.io.read_triangle_mesh(input_file,True)

    #Print information on triangle mesh
    print(mesh)
    print('Vertices:')
    print(np.asarray(mesh.vertices))
    print('Triangles:')
    print(np.asarray(mesh.triangles))

    return mesh

def visualize(mesh):
    # Visualization preparation
    mesh.compute_vertex_normals()
    #print(np.asarray(mesh.triangle_normals))
    mesh.paint_uniform_color([0.24, 0.70, 0.44])

    # Visualize 3D model
    vis = o3d.visualization.Visualizer()
    vis.create_window(window_name='obj visualization', width=1000, height=1000)
    vis.add_geometry(mesh)
    vis.run()

def create_grid(input_mesh, cellsize_grid):
    #Create grid from mesh extents
    #first convert array of vertices to list of vertices
    list_pts_prep = []
    list_pts = []
    for coor in input_mesh.vertices:
        list_pts_prep.append([coor.tolist()])

```

Figure 4

Appendix I: Algorithms

```
#list of x and y coordinates as integers
for coor in list_pts_prep:
    for x,y,z in coor:
        list_pts.append([int(x), int(y)])

#extents: upper x, upper y, lower x, lower y
ux = 418539
uy = 5653512
lx = 418448
ly = 5653423

"""
print('length of list_pts_prep = ', len(list_pts_prep))
for a in range(len(list_pts_prep)):
    if a == 0:
        ux = list_pts[a][0]
        uy = list_pts[a][1]
        lx = list_pts[a][0]
        ly = list_pts[a][1]
    else:
        if list_pts[a][0] > ux:
            ux = list_pts[a][0]
        if list_pts[a][1] > uy:
            uy = list_pts[a][1]
        if list_pts[a][0] < lx:
            lx = list_pts[a][0]
        if list_pts[a][1] < ly:
            ly = list_pts[a][1]
"""

print('ux is', ux)
print('uy is', uy)
print('lx is', lx)
print('ly is', ly)

ncols = math.ceil(((uy - ly) / cellsize_grid))
nrows = math.ceil(((ux - lx) / cellsize_grid))

# Create list of centerpoints of cells
centerpoints = []

first_cp_y = ly + 0.5 * cellsize_grid
first_cp_x = lx + 0.5 * cellsize_grid

for j in range(1, nrows + 1):
    for i in range(1, ncols + 1):
        centerpoints.append([i * cellsize_grid + first_cp_x, j * cellsize_grid + first_cp_y])

array_centerpoints = np.array(centerpoints)
```

```

    return ncols, nrows, lx, ly, centerpoints

def read_height_model(height_model_file):
    coordinates = {}
    xyz = open(height_model_file)

    for line in xyz:
        x,y,z = line.split()
        coordinates[int(float(x)),int(float(y))] = float(z)
    xyz.close()
    return coordinates

def write_raster(ncols, nrows, lx, ly, cellsize_grid, nested_satellite_values, file):
    with open(file, 'w+') as fh:
        fh.write('NCOLS ' + str(ncols) + '\n')
        fh.write('NROWS ' + str(nrows) + '\n')
        fh.write('XLLCORNER ' + str(lx) + '\n')
        fh.write('YLLCORNER ' + str(ly) + '\n')
        fh.write('CELLSIZE ' + str(cellsize_grid) + '\n')
        fh.write('NODATA_VALUE -9999')
        for i in reversed(nested_satellite_values):
            fh.write("\n")
            for point in i:
                fh.write(str(point) + " ")

    print('File written to', file)

def satellite_lines(xyz):
    satellites = sat_pos()
    GPS = []
    Galileo = []
    i = 0
    for constellation in satellites:
        for satellite in constellation:
            values = list(satellite.values())
            if i == 0:
                GPS.append(values[0][2])
            if i == 1:
                Galileo.append(values[0][2])
        i += 1

    lines = []
    for satellite in GPS:
        line = [xyz, [satellite[0], satellite[1], satellite[2]]]
        lines.append(line)
    return lines

```

Figure 6

Appendix I: Algorithms

```
def mesh_to_triangles(mesh):
    list_vertices = []
    for i in mesh.vertices:
        list_vertices.append(i.tolist())
    triangles = []
    for triangle in mesh.triangles:
        point_1 = list_vertices[int(triangle[0])]
        point_2 = list_vertices[int(triangle[1])]
        point_3 = list_vertices[int(triangle[2])]
        triangles.append([point_1, point_2, point_3])
    v1 = [418444, 5653431, 300]
    v2 = [418556, 5653517, 300]
    v3 = [418556, 5653431, 300]
    v4 = [418444, 5653517, 300]
    triangles.append([v1, v2, v4])
    triangles.append([v1, v3, v2])
    return triangles

def sign_of_volume(a,b,c,d):
    B = np.array([b[0] - a[0], b[1] - a[1], b[2] - a[2]])
    C = np.array([c[0] - a[0], c[1] - a[1], c[2] - a[2]])
    D = np.array([d[0] - a[0], d[1] - a[1], d[2] - a[2]])
    return np.sign((1.0 / 6.0) * np.dot(np.cross(B, C), D))

def test(line, triangles):
    intersections = []
    q1 = line[0]
    q2 = line[1]
    for i in triangles:
        p1 = i[0]
        p2 = i[1]
        p3 = i[2]
        if (sign_of_volume(q1,p1,p2,p3) == sign_of_volume(q2,p1,p2,p3)) & (sign_of_volume(q1,q2,p1,p2) == sign_of_volume(q1,q2,p2,p3) == sign_of_volume(q1,q2,p3,p1)):
            return True
        else:
            continue

def main():
    #define input files
    input_file = 'Lod2_32_418_5653_1_NW.obj'
    height_model = 'dgm1_32_418_5653_1_nw.xyz'
    #define output file
    output_file = 'out.asc'
    #define cell size for grid
    cellsize_grid = 2
    #define test point
    point = [418521.00, 5653473.00, 346.42]
```

Figure 7

```

mesh = create_triangle_mesh(input_file)
visualize(mesh)
ncols, nrows, lx, ly, array_centerpoints = create_grid(mesh, cellsize_grid)

list_xyz = read_height_model(height_model)

list_centerpoints_z = []
for [x,y] in array_centerpoints:
    list_centerpoints_z.append([x,y,list_xyz[int(x),int(y)]])

list_z_values = []
for i in range(len(array_centerpoints)):
    list_z_values.append(list_centerpoints_z[i][2])

nested_z_values = []
for i in range(0, len(list_z_values), (ncols)):
    nested_z_values.append(list_z_values[i:(i+ncols)])

triangles = mesh_to_triangles(mesh)

visible_list = []

lines = satellite_lines(point)
print(lines)
visible = []
for line in lines:
    if test(line, triangles) == True:
        print("this line is blocked")
    else:
        print("this line is not blocked")
        visible_list.append(line)
number_of_visible_satellites = len(visible_list)
print(number_of_visible_satellites)
print("number of visible satellites = ", number_of_visible_satellites)

print("visible list:", visible_list)
nested_visible_list = []
for i in range(0, len(visible_list), (ncols)):
    nested_visible_list.append(visible_list[i:(i + ncols)])

write_raster(ncols, nrows, lx, ly, cellsize_grid, nested_visible_list, output_file)

if __name__ == "__main__":
    main()

```

Figure 8

Colophon

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

