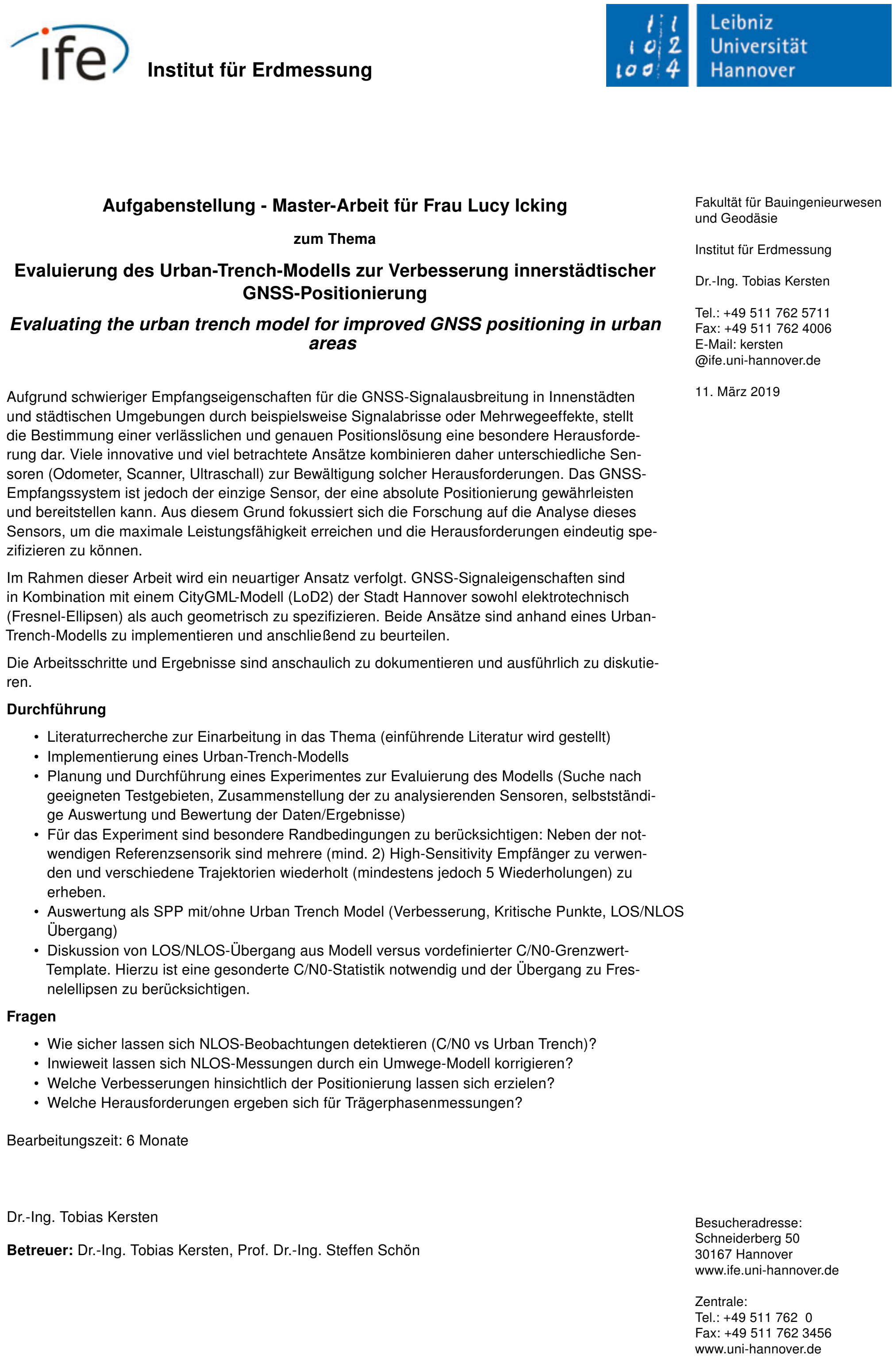
Evaluating The Urban Trench Model For Improved GNSS Positioning In Urban Areas

Master Thesis

Lucy Icking

Institut für Erdmessung (IfE)

September 11, 2019



Institut für Erdmessung

Leibniz Universität Hannover

Aufgabenstellung - Master-Arbeit für Frau Lucy Icking

zum Thema

Evaluierung des Urban-Trench-Modells zur Verbesserung innerstädtischer GNSS-Positionierung

Evaluating the urban trench model for improved GNSS positioning in urban areas

Aufgrund schwieriger Empfangseigenschaften für die GNSS-Signalausbreitung in Innenstädten und städtischen Umgebungen durch beispielsweise Signalabrisse oder Mehrwegeeffekte, stellt die Bestimmung einer verlässlichen und genauen Positionslösung eine besondere Herausforderung dar. Viele innovative und viel betrachtete Ansätze kombinieren daher unterschiedliche Sensoren (Odometer, Scanner, Ultraschall) zur Bewältigung solcher Herausforderungen. Das GNSS-Empfangssystem ist jedoch der einzige Sensor, der eine absolute Positionierung gewährleisten und bereitstellen kann. Aus diesem Grund fokussiert sich die Forschung auf die Analyse dieses Sensors, um die maximale Leistungsfähigkeit erreichen und die Herausforderungen eindeutig spezifizieren zu können.

Im Rahmen dieser Arbeit wird ein neuartiger Ansatz verfolgt. GNSS-Signaleigenschaften sind in Kombination mit einem CityGML-Modell (LoD2) der Stadt Hannover sowohl elektrotechnisch (Fresnel-Ellipsen) als auch geometrisch zu spezifizieren. Beide Ansätze sind anhand eines Urban-Trench-Modells zu implementieren und anschließend zu beurteilen.

Die Arbeitsschritte und Ergebnisse sind anschaulich zu dokumentieren und ausführlich zu diskutieren.

Durchführung

- Literaturrecherche zur Einarbeitung in das Thema (einführende Literatur wird gestellt)
- Implementierung eines Urban-Trench-Modells
- Planung und Durchführung eines Experimentes zur Evaluierung des Modells (Suche nach geeigneten Testgebieten, Zusammenstellung der zu analysierenden Sensoren, selbstständige Auswertung und Bewertung der Daten/Ergebnisse)
- Für das Experiment sind besondere Randbedingungen zu berücksichtigen: Neben der notwendigen Referenzsensorik sind mehrere (mind. 2) High-Sensitivity Empfänger zu verwenden und verschiedene Trajektorien wiederholt (mindestens jedoch 5 Wiederholungen) zu erheben.
- Auswertung als SPP mit/ohne Urban Trench Model (Verbesserung, Kritische Punkte, LOS/NLOS Übergang)
- Diskussion von LOS/NLOS-Übergang aus Modell versus vordefinierter C/N0-Grenzwert-Template. Hierzu ist eine gesonderte C/N0-Statistik notwendig und der Übergang zu Fresnelellipsen zu berücksichtigen.

Fragen

- Wie sicher lassen sich NLOS-Beobachtungen detektieren (C/N0 vs Urban Trench)?
- Inwieweit lassen sich NLOS-Messungen durch ein Umwege-Modell korrigieren?
- Welche Verbesserungen hinsichtlich der Positionierung lassen sich erzielen?
- Welche Herausforderungen ergeben sich für Trägerphasenmessungen?

Bearbeitungszeit: 6 Monate

Dr.-Ing. Tobias Kersten

Betreuer: Dr.-Ing. Tobias Kersten, Prof. Dr.-Ing. Steffen Schön

Fakultät für Bauingenieurwesen
und Geodäsie

Institut für Erdmessung

Dr.-Ing. Tobias Kersten

Tel.: +49 511 762 5711
Fax: +49 511 762 4006
E-Mail: kersten
@ife.uni-hannover.de

11. März 2019

Besucheradresse:
Schneiderberg 50
30167 Hannover
www.ife.uni-hannover.de

Zentrale:
Tel.: +49 511 762 0
Fax: +49 511 762 3456
www.uni-hannover.de

Abstract

Urban environments still form a challenge for Global Navigation Satellite System (GNSS) positioning due to its difficult conditions for signal propagation. Many obstruction sources disturb the GNSS signal, signal loss or multipath are examples for effects occurring in these cases.

This work focuses on an approach which distinguishes visible line-of-sight (LOS) satellite signals from obstructed non-line-of-sight (NLOS) signals. These are determined in a self-developed algorithm with the help of a 3D CityModel that is provided by the city of Hanover. For the evaluation, a kinematic experiment is conducted with geodetic and high-sensitivity receivers in a repeated trajectory. Further sensors were used to generate a reference trajectory.

Open questions not only concern the signal characterization with respect to LOS/NLOS properties but also to what extent NLOS observations can be used in a constructive sense. Therefore, reflection points of NLOS signals are calculated, examined and introduced in a Single Point Positioning (SPP) solution. Additionally, these issues are put in the context of different receivers, each of which has different properties and hence diverse outcomes.

Results show that the LOS status can be determined reliably but that the computed detour lengths not always match the Observed Minus Computed (OMC) values. Taking only those additional path lengths into account that fit the OMC, an improvement could be achieved in an SPP solution. Further findings show that OMC are especially high in street sections that have a perpendicular orientation compared to the satellite's azimuth angle.

Declaration in Lieu of an Oath

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

This work has not been published in any form before and has not been submitted to any other examination office.

(place, date)

(signature)

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation and Outline

Urban environments still form a challenge for Global Navigation Satellite System (GNSS) positioning due to its difficult conditions for signal propagation. Many obstruction sources disturb the GNSS signal, signal loss or multipath are examples for effects occurring in these cases.

This work focuses on an approach which distinguishes visible line-of-sight (LOS) satellite signals from obstructed non-line-of-sight (NLOS) signals. These are determined in a self-developed algorithm with the help of a 3D CityModel that is provided by the city of Hanover. For the evaluation, a kinematic experiment is conducted with geodetic and high-sensitivity receivers in a repeated trajectory. Further sensors were used to generate a reference trajectory.

Open questions not only concern the signal characterization with respect to LOS/NLOS properties but also to what extent NLOS observations can be used in a constructive sense. Therefore, reflection points of NLOS signals are calculated, examined and introduced in a Single Point Positioning (SPP) solution. Additionally, these issues are put in the context of different receivers, each of which has different properties and hence diverse outcomes.

Over the course of this work, first some background information is given in chapter 2 which the programming and analysis is based on. It is followed by a detailed description of the algorithm structure that is used for LOS/NLOS determination and reflection calculations. Next, a validation of the algorithm is performed with a static experiment analysis and a simulated measurement to calculate detour lengths. In these chapters 4 and 5, first findings towards LOS/NLOS and detour characterizations are stated. In chapter 6, the planning and execution of the kinematic experiment is described and is followed by the main part of this work, the analysis of signals in the context of LOS/NLOS, C/N$_0$ and OMC in chapter 7: kinematic experiment analysis. A conclusion in chapter 8 summarizes the findings of this work and gives an outlook on possible improvements and focuses in the future.

1.2 State of the Art

The work is based on a paper by Betaille et al. (2013) in which the authors use a simplified 3D model of the urban environment to determine which satellites are visible based on the relation between user position on the street, building heights and satellite position. The term "Urban Trench Model" is also defined here, referring to a geometrical street model instead of the frequently used term "canyon" for downtown tall buildings in mega cities. Betaille et al. (2013) states that a positioning method based on visible LOS satellites only has a better accuracy than standard methods. But if LOS only are not sufficient, corrected NLOS satellite signals based on building reflections are used for a still improved positioning.

Besides this work, there are multiple other approaches that make use of 3D mapping aided (3DMA) positioning methods such as the so-called Shadow Matching (Groves, 2011). It also takes advantage of a 3D city model to predict satellite visibilities and then compares it to the measured satellite visibility to determine the user's position.

Instead of only focusing on excluding NLOS signals, some works propose to use NLOS observations in a constructive sense by determining the signal delay. Kumar and Petovello (2014) develops a Signal Delay Matching algorithm which matches observed path delays with predicted path delays based on a 3D city model. Ben-Moshe et al. (2011) even constructs an own 3D model by utilizing the user's position and LOS information to satellites and compares the algorithm with findings from Simultaneous Localization And Mapping (SLAM), resulting in positive effects for positioning accuracy.

Hsu (2017) also proposes a method in which only the elevation angle of the incoming satellite signal and the distance to the reflecting building surface is taken into account for describing the delay. By doing so, the authors try to avoid the application of a 3D model due to computational load.

What has been neglected in the works mentioned above is the behavior of different receivers, especially those of geodetic receivers compared to high-sensitivity receivers. This is an aspect which is focused on in this thesis by deploying multiple geodetic and high-sensitivity devices. Furthermore, a sophisticated 3D CityGML model has been applied which contains more information compared to the simplified model used in Betaille et al. (2013). Besides, an emphasis is put on the comparison between determined detour lengths, i.e. additional path lengths, and Observed Minus Computed (OMC) values calculated from a precise IMU-supported reference trajectory.

2 Background

In this chapter, some background information is presented related to the topics that are dealt within this thesis. First, we take a look at the city model of Hanover that is used to detect whether a satellite is visible or not. Secondly, some basics concerning aspects of a Global Navigation Satellite System (GNSS) are introduced that are relevant for the studies of this work.

2.1 City Model

A city model is the core aspect of this thesis to distinguish between satellites in line-of-sight and non-line-of-sight. Therefore, a CityGML model is used as it is easy to handle and free for use.

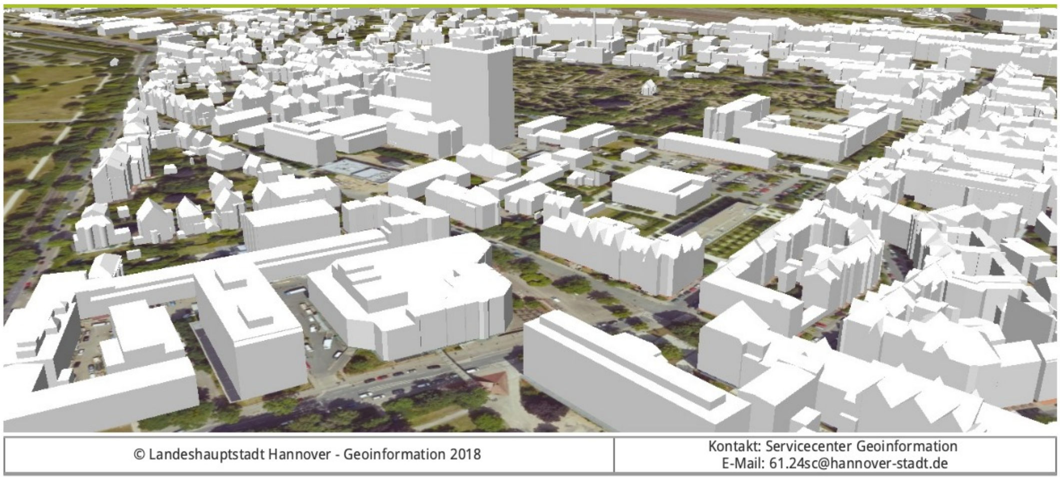

© Landeshauptstadt Hannover - Geoinformation 2018
Kontakt: Servicecenter Geoinformation
E-Mail: 61.24sc@hannover-stadt.de

Figure 2.1: Visualization of the Hanover 3D CityModel (Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation, 2018). The ground is visualized through pictures of images flights.

2.1.1 CityGML Hanover

To model the city of Hanover a 3D city model provided by the geoinformation department of Hanover is used (Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation, 2017). Figure 2.1 shows an extract from the model provided by the website of `hannover-3d.de` by Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation (2018). It pictures the provincial capital in a digital and three-dimensional way. There are two levels of detail (LoD). LoD1 includes more simple block graphics in which the building heights are taken from the mean of eaves and ridges, the height accuracy is approximately ±5 m. In LoD2, eaves and ridges are depicted both together with a standardised roof shape. The building height accuracy here is about ±1 m, see also table 2.1 for some key information about the CityModel.

The data has been recorded in 2010 during a laserscan flight and has then been processed to around 130,000 buildings, however small buildings with a ground area less than 15 m^2 and huts in garden plots have been left out. The coordinates are given in UTM ETRS89, the absolute heights are based on Deutsches

Table 2.1: Key information about CityGML Hanover (Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation, 2017).

Aspect	Value
Version	CityGMl 2.0
Levels of detail	LoD1 and LoD2
Reference system	ETRS89 (UTM)
Height reference system	DHHN92
Building height accuracy LoD1	±5 m
Building height accuracy LoD2	±1 m

Haupthöhennetz 1992 (DHHN92) and reassured by survey height reference points. Therefore, it can be assumed that the absolute coordinates are within an accuracy level of centimeters (cf. Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation (2017)).

Because the accuracy of LoD1 is not sufficient for the purposes of this work, LoD2 is used. The shape of the roof is an important information for the detection whether a satellite is in line-of-sight or not. Moreover, the geometry between satellite, user and the flared roof is one of the main aspects in this work.

The provider also released a map in which all the 1 km x 1 km tiles are plotted over the corresponding position in the city of Hanover. Every tile has an ID and is provided as a single GML file.

Figure 2.2 shows a visualisation of the provided geometries. This specific data is taken from a small GML test file that pictures the police department Hanover with an extent of approximately 120 m x 140 m. With the help of images (c) and (d) of the real situation taken from Google Maps, it can be noticed that trees are not included and small objects are neglected.

Yet roofs are pictured in a detailed way as well as the form of the building, e.g. notice the small spire attached to the house in the lower left corner. This alone is described by many polygons to show its round character.

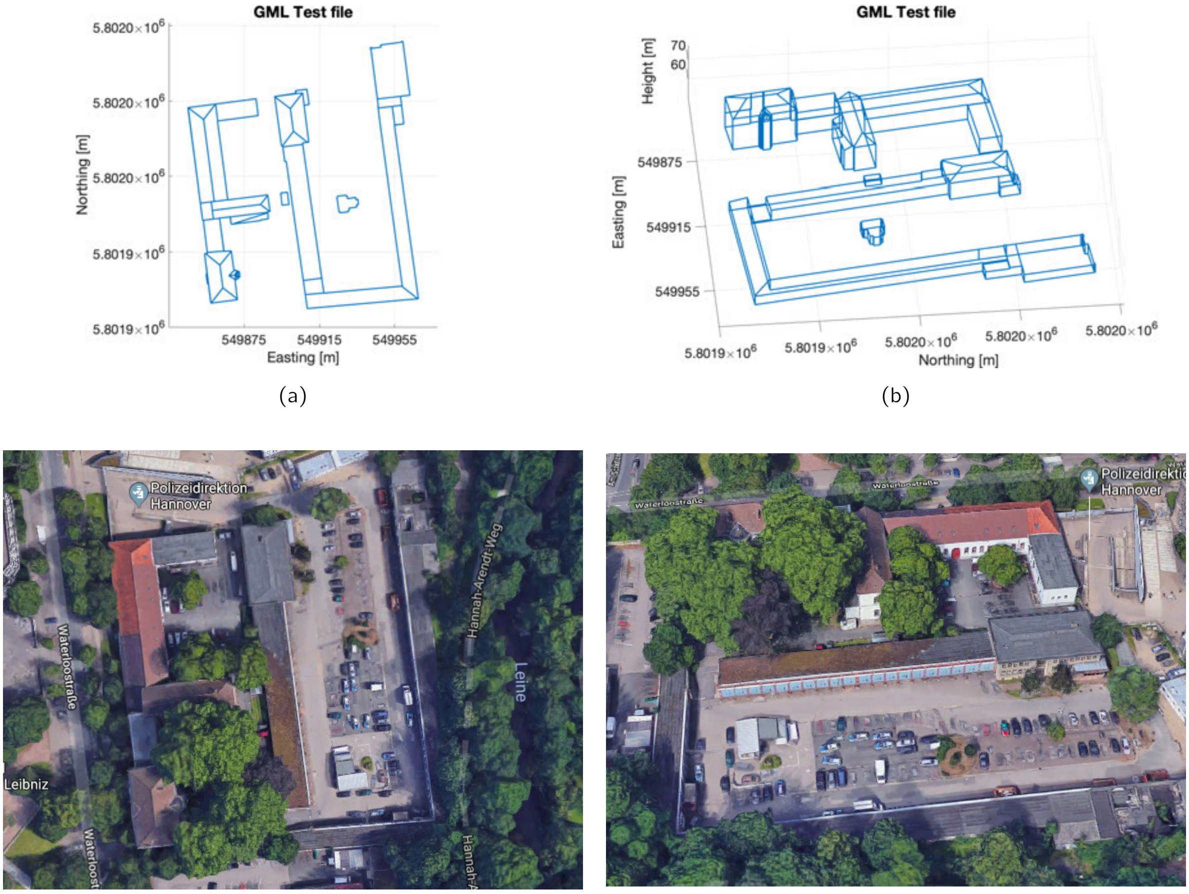
What might be an error due to lack of timeliness of the data is the small building in between the garages. It cannot be found on the satellite image on the parking lot.

All in all, the accuracy LoD2 is sufficient for the purposes of this work. Small errors can be neglected as many other objects from the real surroundings (trees, cars etc.) are not even existent in the City-Model.

2.1.2 GML file format

The data for the CityModel are provided in the Geography Markup Language (GML) format. GML is an XML-based format (Extensible Markup Language) developed by the OpenGIS Consortium (OGC). The format is specialized for geographic information (Gröger et al., 2012).

A GML file contains text, just like XML, so that both human and machines are able to read it. Inside of it, it shows a list of features which are divided into properties and geometries. Properties include name, type and value description whilst geometries include basic elements such as points, lines, curves, surfaces and polygons (cf. Ron Lake (2019)). In our case in which we deal with buildings, the main geometry contained in the provided GML files are polygons described by a list of points. To have a complete set of polygon edges, the first vertex is repeated at the end of the list.



(a) (b)

(c) (d)

Figure 2.2: Visualisation of CityGML Hanover test file with Matlab (top view in (a) and side view in (b)) and comparison with real situation (top view in (c) and side view in (d)) taken from Google Maps (Google Maps, 2019). Coordinates in UTM.

Listing 2.1: First lines of CityGML file `5480_5804.gml`.

```
<CityModel xmlns:bldg="http://www.opengis.net/citygml/building/2.0" xmlns:gml="http://www.opengis.net/gml"
    ↪ xmlns="http://www.opengis.net/citygml/2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    ↪ xsi:schemaLocation="http://www.opengis.net/citygml/building/2.0␣http://schemas.opengis.net/citygml/
    ↪ building/2.0/building.xsd">
  <gml:boundedBy>
    <gml:Envelope srsName="urn:adv:crs:ETRS89_UTM32*DE_DHHN92_NH" srsDimension="3">
      <gml:lowerCorner>547987.15 5803995.42 50.639</gml:lowerCorner>
      <gml:upperCorner>549008.08 5805011.59 126.75</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <cityObjectMember>
    <bldg:Building gml:id="UUID_3103730e-a6a7-49b0-bc05-95442b8ead4c">
      <gml:name>UUID_3103730e-a6a7-49b0-bc05-95442b8ead4c</gml:name>
      <bldg:roofType>3200</bldg:roofType>
      <bldg:boundedBy>
        <bldg:WallSurface gml:id="UUID_1ef9a1a9-8bfe-49aa-a716-8c8c41feaa84">
          <bldg:lod2MultiSurface>
            <gml:MultiSurface srsName="urn:adv:crs:ETRS89_UTM32*DE_DHHN92_NH" srsDimension="3">
              <gml:surfaceMember>
                <gml:Polygon>
                  <gml:exterior>
                    <gml:LinearRing>
                      <gml:posList>548428.73 5804294.95 54.191 548428.73 5804294.95 60.51 548428.7 5804295.13
                          ↪ 60.51 548428.7 5804295.13 54.191 548428.73 5804294.95
                       54.191</gml:posList>
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:Polygon>
              </gml:surfaceMember>
            </gml:MultiSurface>
          </bldg:lod2MultiSurface>
        </bldg:WallSurface>
      </bldg:boundedBy>
```

The listing above shows the top section taken from a GML file of the Hanover data set, more specifically the tile of Hanover Nordstadt. The first few lines contain information about the file itself, followed by information about the geometric extent that is covered. It is stated that the corners are about 1 km apart in each x- and y-direction while the maximum difference in height is about 76 m.

The first object in this list is a building with information about the specific ID, roof type, coordinate and height system. Subsequently, we get information about the polygons that are part of the building, namely the coordinates of the points framing the polygon. With the help of these, we can extract the spatial information of the object and depict it.

2.2 Computational Geometry Algorithms

In this section, functions related to the topic of computational geometry that are implemented in this work are introduced.

As the city model involves a series of geometric shapes, lines and objects, computational geometry algorithms are used to get information about their spatial relations.

2.2.1 Ray-Box Intersection

For the ray-box intersection test, the algorithm proposed in Williams et al. (2005) is used. It is based on axis-aligned bounding boxes with known values for the box minimum and maximum coordinates as well as the origin O and direction D of the ray.

As of simplification, the workflow of the algorithm is shown for 2D in figure 2.3.

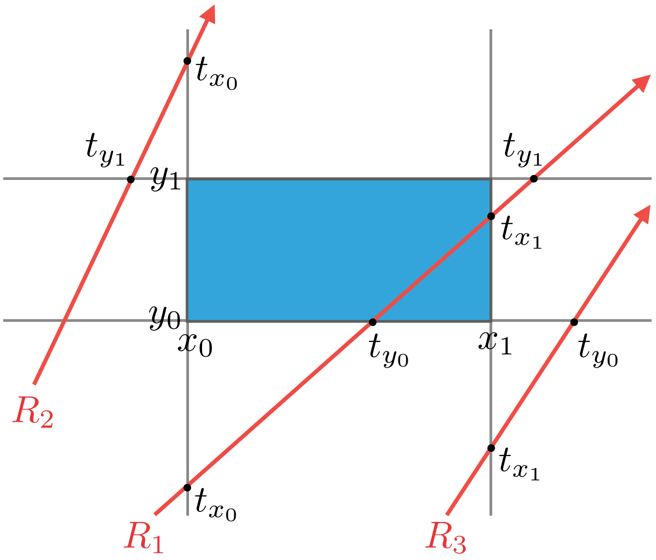


Figure 2.3: Workflow of the ray-box intersection. Ray R_1 intersects the box, R_2 and R_3 only pass but not intersect.

Taking a ray $R(t) = O + t \cdot D$, the intersection time points for each of the bounding planes x_0, x_1 as well as y_0, y_1 of the box can be calculated using

$$t_{x_i} = \frac{x_i - O_x}{D_x} \tag{2.1}$$

$$t_{y_i} = \frac{y_i - O_y}{D_y} \tag{2.2}$$

where i stands for either the upper or lower side of the box. Note that not all of the intersection points necessarily have to lie on the box borders.

If we now compare the order of all points t_{x_i} and t_{y_i}, it becomes obvious that the lower borders ($i = 0$) of the box have to be intersected before the upper borders ($i = 1$) in order for the ray to cross the box as it is the case for R_1 in figure 2.3. Here, t_{x_0} comes before t_{y_1} as does t_{y_0} before t_{x_1}. For rays R_2 and R_3, the opposite is the case (t_{y_1} comes before t_{x_0} for ray R_2, t_{x_1} comes before t_{y_0} for ray R_3,) and hence those rays are not intersecting the box.

If a ray has a negative slope, the definition of the lower and upper borders can be switched. Additionally, Williams et al. (2005) proposes improvements for numerical problems such as a slope near zero to enhance the performance speed.

2.2.2 Ray-Triangle Intersection

For the purpose of calculating the intersection between a ray and a triangle, the *Möller-Trumbore intersection algorithm* named after its inventors is used (Möller and Trumbore, 2005). It is a fast algorithm as it does not need to compute the plane equation spanned by the triangle.

A ray $R(t)$ is defined by its origin O and normalized direction D so that

$$R(t) = O + tD \tag{2.3}$$

while a triangle is defined by the three vertices V_0, V_1 and V_2 it is bounded by, cf. figure 2.4.

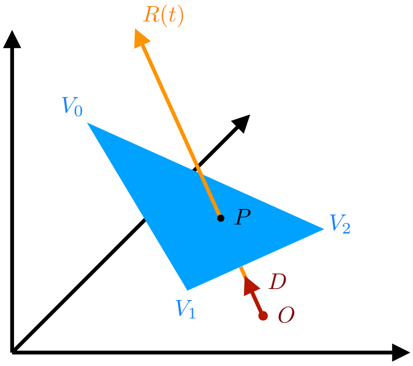


Figure 2.4: A triangle and a ray with their parameters.

For the algorithm, barycentric coordinates are used which for the case of a triangle is a triplet of u, v and w with properties so that

$$u, v, w \geq 0 \quad \text{and} \quad u + v + w = 1\,. \tag{2.4}$$

Every point P can now be described by a unique convex combination of V_0, V_1 and V_2 which means

$$P(u, v, w) = uV_0 + vV_1 + wV_2 \tag{2.5}$$

but since the triplet of barycentric coordinates only has a degree of freedom of 2, we can also write

$$P(u, v) = uV_0 + vV_1 + (1 - u - v)V_2\,. \tag{2.6}$$

To find the intersection point of the ray with the triangle, we set

$$R(t) = P(u, v) \tag{2.7}$$

which yields

$$O + tD = uV_0 + vV_1 + (1 - u - v)V_2 \tag{2.8}$$

and is equal to

$$\begin{bmatrix} -D & V_0 - V_2 & V_1 - V_2 \end{bmatrix} \begin{bmatrix} t \\ u \\ v \end{bmatrix} = O - V_2\,. \tag{2.9}$$

In equation 2.9 we can see that the barycentric coordinates (u, v) as well as the distance t from the origin can be found by solving the linear system. The barycentric coordinates can afterwards be translated back to cartesian if needed.

To use the algorithm inside a function, the user sets the input parameters of the origin O, the direction D and the vertices V_0, V_1 and V_2. The function then uses the linear equation 2.9 to compute if and where the intersection takes place. The output parameters are a flag for intersection (1 for intersection, 0 for no intersection) and, if desired, the intersection point coordinates either in barycentric or cartesian form.

The big advantage of this algorithm is that only one calculation has to be done inside the function which is the linear equation 2.9. There is no actual plane that has to be set up and no check if a point lies within a polygon. This reduction of calculation steps lowers the runtime tremendously.

2.2.3 Point-In-Polygon Test

To test if the intersection point Q lies within a specific polygon with vertices $P(1), ..., P(n)$, a ray casting algorithm is used, see Haines (1994).

The algorithm shoots a ray in any direction, for simplicity, the x-direction is used. Starting from the point itself, it now counts the intersections of the ray with the edges of the polygon. If the resulting number is even, the corresponding point is on the outside of the polygon. If the number is odd, the point is on the inside of the polygon, see figure 2.5.

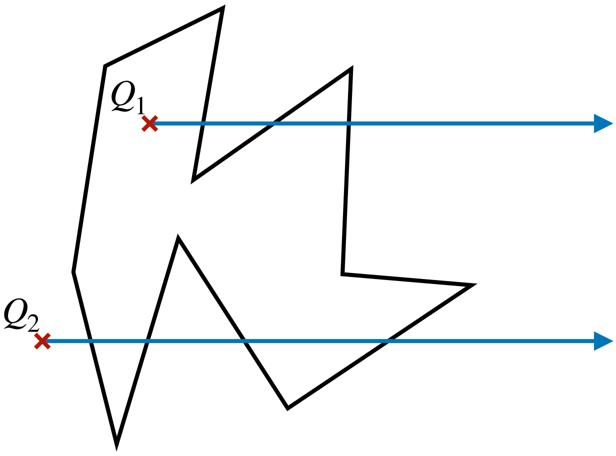


Figure 2.5: The ray from Q_1 is intersecting three times with edges, the ray from Q_2 four times. Q_1 lies within the polygon, Q_2 not.

In cases in which the ray intersects the vertices directly, we assume that those vertices lie slightly above the ray. This means that vertices on top of a maximum peak are counted as two intersections, vertices at the bottom of a minimum peak are counted as zero and vertices along a line are counted as one. Hence, the outcome of even or uneven number of intersections will still be correct to determine whether or not a point lies within the polygon.

2.3 GNSS Aspects

In this section, a brief overview is given over aspects with respect to GNSS. This includes information about the satellite-user geometry, signal reflection and determination of the user's position.

2.3.1 Line-Of-Sight (LOS)

The Line-Of-Sight (LOS) vector represents the direct line between user position and satellite and can be either given in earth-centered, earth-fixed (ECEF) coordinates like the satellite or in topocentric measures.

To compute the LOS vector, both coordinates of the station s and satellite k have to have the same coordinate reference. For ECEF (index e) this is

$$LOS^e = \begin{pmatrix} X^k \\ Y^k \\ Z^k \end{pmatrix} - \begin{pmatrix} X_s \\ Y_s \\ Z_s \end{pmatrix} \tag{2.10}$$

and for topocentric coordinates (index t) with origin at the station $S = \begin{pmatrix} \varphi \\ \lambda \end{pmatrix}$ with geographical latitude φ and longitude λ, LOS^e has to be rotated. This leads to

$$LOS^t = R_e^t \cdot LOS^e \tag{2.11}$$

with

$$R_e^t = \begin{pmatrix} -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ -\sin\lambda & \cos\lambda & 0 \\ \cos\varphi\cos\lambda & \cos\varphi\sin\lambda & \sin\varphi \end{pmatrix}. \tag{2.12}$$

Taking LOS^t, the elevation ϵ and azimuth α can be calculated to

$$\epsilon = \arctan\left(\frac{LOS^t(3)}{\sqrt{LOS^t(1)^2 + LOS^t(3)^2}}\right) \tag{2.13}$$

$$\alpha = \arctan\left(\frac{LOS^t(2)}{LOS^t(1)}\right) \tag{2.14}$$

2.3.2 Signal Reflection and Multipath

A satellite signal is an electromagnetic wave propagating in a right-handed circular polarization (RHCP) which means that the electric field vector $\boldsymbol{E}$ varies with a constant amplitude. If an electromagnetic wave now meets a transition of two media, it gets partly reflected. This is the case if the signal traveling trough air meets a surface, e.g. a wall. A rough surface will cause diffuse reflection while a smooth surface creates a specularly reflection (Hofmann-Wellenhof et al., 2008).

A reflection in satellite positioning might occur if the receiver is located between high buildings and the direct LOS is blocked. Signals that arrive at the receiver will then have a longer pathway than the direct signal would have, causing a positioning error. If direct and indirect waves reach the receiver, there will also be positioning errors which is called multipath. In some cases, the errors may be of a magnitude so that the usage of GPS might be limited or even interrupted (Leick et al., 2015).

The magnitude of the error caused by multipath depends on the following parameters (Mansfeld, 2004):

- power ratio between direct and indirect wave
- range difference between direct and indirect wave
- directional receiving pattern of the receiving antenna
- signal processing within the receiver

Strong reflective surfaces provoke a change of the polarization direction from right-handed to left-handed. Therefore, in order to reduce multipath, an antenna that is designed especially for RHCP is beneficial.

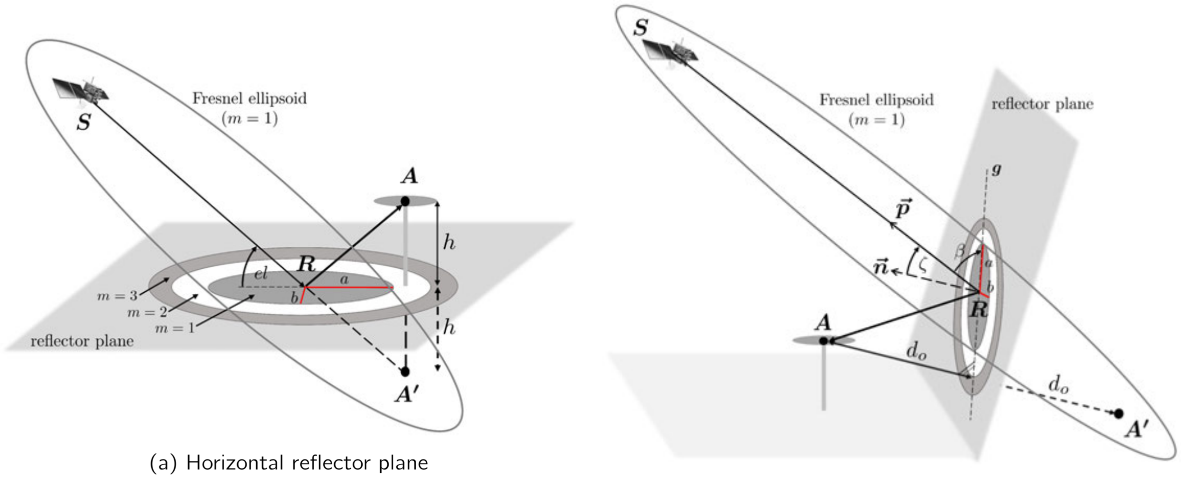
To mitigate multipath effects by signal and data processing, techniques such as smoothing carrier phases, signal-to-noise ratio dependencies and data combinations have been investigated. Possible are also techniques within the receiver technology (Hofmann-Wellenhof et al., 2008).

2.3.3 Fresnel Ellipsoid And Reflection

In a homogenous medium, electromagnetic waves propagate as a straight line and follow the laws of Fermat's principle at the interface of two different media (law of reflection). Taking GPS, the satellite emission aperture angle is ±13.9° which means that modeling electromagnetic wave reflections as a discrete point is not sufficient. Instead, Fresnel zones are used (Rost, 2011).

A Fresnel ellipsoid is a rotation ellipsoid with its focal points based on the satellite position ***S*** and the mirror point ***A′*** of the receiving antenna at point ***A*** mirrored at the reflector plane, see figure 2.6. The intersected area is called Fresnel zone. The whole area inside the Fresnel zone has an effect on the reflection process.

Reflection points and Fresnel ellipses



(a) Horizontal reflector plane

(b) Arbitrarily oriented reflector plane

Figure 2.6: Fresnel zones as intersection of Fresnel ellipsoid with reflector plane (Zimmermann et al., 2018).

For horizontal reflector planes, the Fresnel zone depends on the satellite elevation angle ϵ, the carrier wavelength λ and the height of the antenna above ground h. As in Zimmermann et al. (2018), the semi-major and semi-minor axes a and b of the Fresnel zone are given by

$$b = \sqrt{\frac{\lambda h}{\sin\epsilon} + \left(\frac{\lambda}{2\sin\epsilon}\right)^2} \tag{2.15}$$

$$a = \frac{b}{\sin\epsilon} \tag{2.16}$$

where the direction of the semi-major axis is defined by the satellite-antenna orientation.

To get the coordinates of the reflection point $\boldsymbol{R}$ in the horizontal Fresnel zone, we define the line g between the satellite $\boldsymbol{S} = \begin{pmatrix} x_S \\ y_S \\ z_S \end{pmatrix}$ and the mirrored reflection point $\boldsymbol{A'} = \begin{pmatrix} x_A \\ y_A \\ -z_A - 2h_g \end{pmatrix}$ with h_g being the height of the ground as

$$g : \vec{\boldsymbol{x}} = \boldsymbol{S} + \kappa(\boldsymbol{A'} - \boldsymbol{S}) \tag{2.17}$$

which can be inserted into the ground plane $E : z = h_g$ so that

$$\kappa_{\boldsymbol{R}} = \frac{h_g - z_S}{2h_g - z_A - z_S} \tag{2.18}$$

and

$$\boldsymbol{R} = \boldsymbol{S} + \kappa_{\boldsymbol{R}} \vec{\boldsymbol{SA'}}. \tag{2.19}$$

With higher elevation, $\boldsymbol{R}$ moves closer to the antenna point and the Fresnel zone gets smaller.

For arbitrarily oriented reflector planes, the elevation angle ϵ cannot be used directly to compute the semi-major axes. Instead, the incidence angle ζ of the satellite signal with respect to the reflector plane (cf. figure 2.6b) has to be determined first to

$$\zeta = \arccos\left(\frac{\vec{\boldsymbol{n}} \cdot \vec{\boldsymbol{p}}}{\|\vec{\boldsymbol{n}}\| \cdot \|\vec{\boldsymbol{p}}\|}\right) \tag{2.20}$$

where $\vec{\boldsymbol{p}}$ denotes the vector between the reflection point $\boldsymbol{R}$ and $\vec{\boldsymbol{n}}$ the normal vector of the reflector plane. For the angle β between the reflector plane and signal ray, this leads to

$$\beta = 90^\circ - \zeta \tag{2.21}$$

instead of using the elevation angle ϵ.

With the distance d_0 between the antenna and reflector plane, the semi-major axes can now be derived to

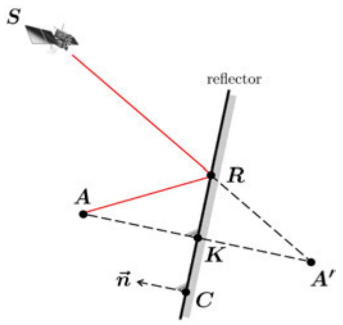


Figure 2.7: Points of interest along the reflector plane (Zimmermann et al., 2018).

$$b = \sqrt{\frac{\lambda d_0}{\sin\beta} + \left(\frac{\lambda}{2\sin\beta}\right)^2}, \tag{2.22}$$

$$a = \frac{b}{\sin\beta}. \tag{2.23}$$

The reflection point in this case is located at

$$\boldsymbol{R} = \boldsymbol{A'} + \frac{(\boldsymbol{C} - \boldsymbol{A'}) \cdot \vec{\boldsymbol{n}}}{(\boldsymbol{S} - \boldsymbol{A'}) \cdot \vec{\boldsymbol{n}}}(\boldsymbol{S} - \boldsymbol{A'}) \tag{2.24}$$

with (2.25)

$$\boldsymbol{A'} = \boldsymbol{A} + 2\,(\boldsymbol{K} - \boldsymbol{A}), \tag{2.26}$$

$$\boldsymbol{K} = \boldsymbol{A} + \frac{(\boldsymbol{C} - \boldsymbol{A}) \cdot \vec{\boldsymbol{n}}}{\vec{\boldsymbol{n}} \cdot \vec{\boldsymbol{n}}}\vec{\boldsymbol{n}} \tag{2.27}$$

where $\boldsymbol{C}$ refers to an arbitrary point on the reflector plane and $\boldsymbol{K}$ is the projected antenna point from $\boldsymbol{A}$ onto the reflector plane along the normal vector $\vec{\boldsymbol{n}}$, cf. figure 2.7.

The detour length which constitutes the additional length caused by the reflection can now be computed by subtracting

$$d_l = \overline{\boldsymbol{SR}} + \overline{\boldsymbol{RA}} - \overline{\boldsymbol{SA}}. \tag{2.28}$$

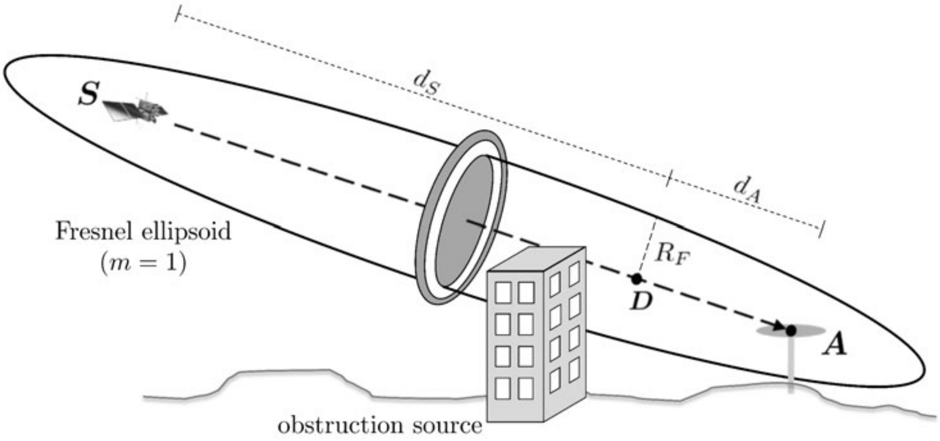


Figure 2.8: Fresnel ellipsoid in the context of LOS transmission (Zimmermann et al., 2018). Obstructions may lie within the ellipsoid diffracting the signal.

Fresnel ellipsoid for signal characterization

In the context of LOS transmission, Fresnel ellipsoids can also be used to characterize the type of signal propagation (Hannah, 2001). The rotation ellipsoid is then spanned between the antenna point $\boldsymbol{A}$ itself and the satellite $\boldsymbol{S}$ as shown in figure 2.8.

At any given point $\boldsymbol{D}$ along the signal path, the radius R_F of the Fresnel ellipsoid can be determined by

$$R_F = \sqrt{\frac{\lambda d_S d_A}{d_S + d_A}} \tag{2.29}$$

where d_S represents the distance between $\boldsymbol{S}$ and $\boldsymbol{D}$, d_A denotes the distance between $\boldsymbol{A}$ and $\boldsymbol{D}$.

If an obstruction source is larger than the Fresnel zone at the respective point, it will lead to a complete blocking of the signal. If it is smaller than the Fresnel zone, the signal can get diffracted.

Hereby, the amount of diffraction and subsequently the reduction of signal strength depends on the area of the Fresnel zone that is blocked (Hristov, 2000). The bigger the area, the higher the diffraction effect. However, until a certain point, the diffraction factor forms a wave, resulting in a non-monotonic curve, see figure 2.9. This alternating diffraction factor will cause the signal strength to oscillate for example when the satellite is setting over the edge of a building.

2.3.4 Single Point Positioning

The single point positioning (SPP) method is used to determine the receiver position from code pseudorange measurements. These pseudoranges can be modeled at an epoch t as

$$R_S^k = \rho_S^k(t) + c\left(\delta_S(t) - \delta^k(t)\right) \tag{2.30}$$

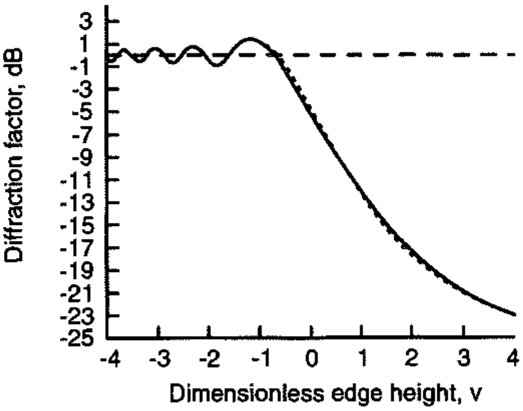


Figure 2.9: Diffraction factor from a knife-edge hill versus dimensionless height v. $v < 0$ means less than half of the Fresnel ellipsoid is covered, $v > 0$ more than half. The dashed and solid line represent two different approaches to determine the curve (Hristov, 2000).

where R_S^k denotes the measured code pseudorange between station S and satellite k and $\rho_S^k(t)$ is the geometric distance between the two. $\delta_S(t)$ denotes the receiver clock offset and $\delta^k(t)$ the satellite clock offset w.r.t. GPS time (Hofmann-Wellenhof et al., 2008). c is the constant speed of light.

$\rho_S^k(t)$ can be written as

$$\rho_S^k(t) = \sqrt{(X^k(t) - X_S(t))^2 + (Y^k(t) - Y_S(t))^2 + (Z^k(t) - Z_S(t))^2} \tag{2.31}$$

where the coordinate components of each satellite $X^k(t), Y^k(t), Z^k(t)$ and receiver $X_S(t), Y_S(t), Z_S(t)$ are introduced in earth-centered, earth-fixed (ECEF) coordinates at an epoch t.

To estimate the unknown receiver position and clock bias

$$\boldsymbol{x} = (X_S, Y_S, Z_S, c\delta_S)^T \tag{2.32}$$

with observed pseudoranges

$$\boldsymbol{y} = \left(R_S^{k=1}, R_S^{k=2} \dots R_S^{k=m}\right)^T \tag{2.33}$$

in a single point positioning approach, there is usually an iterated weighted least square estimation applied, e.g. within the software *RTKlib* (Takasu, 2013). This is done by solving for

$$\hat{\boldsymbol{x}}_{i+1} = \hat{\boldsymbol{x}}_i + \left(\boldsymbol{H}^T \boldsymbol{W} \boldsymbol{H}\right)^{-1} \boldsymbol{H}^T \boldsymbol{W} \left(\boldsymbol{y} - \boldsymbol{h}(\hat{\boldsymbol{x}}_i)\right) \tag{2.34}$$

with the measurement equation

$$\boldsymbol{h}(\hat{\boldsymbol{x}}) = \begin{pmatrix} \rho_S^{k=1} + c\delta_S - c\delta^{k=1} + I_S^{k=1} + T_S^{k=1} \\ \rho_S^{k=2} + c\delta_S - c\delta^{k=2} + I_S^{k=2} + T_S^{k=2} \\ \vdots \\ \rho_S^{k=m} + c\delta_S - c\delta^{k=m} + I_S^{k=m} + T_S^{k=m} \end{pmatrix} \tag{2.35}$$

and its partial derivative matrix $\boldsymbol{H}$, differentiated with respect to each of the unknowns. I_S^k stands for the ionospheric impact between station S and satellite k, T_S^k for the tropospheric.

The weight matrix $\boldsymbol{W}$ is designed in a way where

$$\boldsymbol{W} = \mathrm{diag}\left(\frac{1}{\sigma_{k=1}^2}, \frac{1}{\sigma_{k=2}^2}, \dots, \frac{1}{\sigma_{k=m}^2}\right) \tag{2.36}$$

with

$$\sigma^2 = F^k R_S \left(a_\sigma^2 + b_\sigma^2 \cdot \frac{1}{\epsilon_S^k} \right) + \sigma_{eph}^2 + \sigma_{ion}^2 + \sigma_{trop}^2 + \sigma_{bias}^2 \tag{2.37}$$

where

F^k satellite system error factor [ ]

R_S code/carrier-phase error ratio [ ]

a_σ, b_σ carrier-phase error factors [m]

σ_{eph} standard deviation of ephemeris and clock error [m]

σ_{ion} standard deviation of ionosphere correction model [m]

σ_{trop} standard deviation of troposphere correction model [m]

σ_{bias} standard deviation of code bias error [m]

cf. Takasu (2013).

3 Programming

In this chapter, the workflow of all the algorithms applied for this work are documented. First, the focus lies on line-of-sight/non-line-of-sight (LOS/NLOS) determination to decide whether a satellite signal comes from a satellite that is visible or not. This can be done in different modes which is why there is a subdivision between single point mode, trajectory mode and grid mode. Secondly, the reflection parameters and resulting detour lengths are calculated applying the formulas from Fresnel ellipse computations of section 2.3.3. Finally, an outlook is given to find points that could be improved.

3.1 LOS/NLOS Determination

The LOS/NLOS determination algorithm is the core of the programming part of this work. It will tell if a satellite is visible or not based on a given user position and time epoch and makes use of a 3D city model.

Generally, the algorithm is divided into three parts: settings, data preparation and the final LOS/NLOS calculation. In settings, parameters can be modified to meet the current aims. In data preparation, the CityModel and its polygons are edited to improve the algorithm's performance. Lastly, the LOS/NLOS calculation will apply a sequence of computational algorithms to determine if a satellite is in sight or not.

The basic algorithm structure is seen in figure 3.1. The output is a logical LOS matrix of size *number of timestamps* x *number of satellites* which will have flags of *1* if the respective satellite is visible at the corresponding epoch. If an NLOS matrix is desired, the LOS matrix is simply negated.

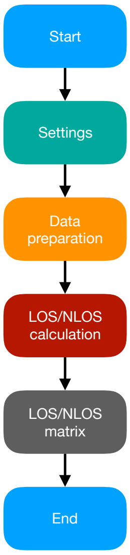


Figure 3.1: LOS/NLOS algorithm structure.

3.1.1 Settings

In settings, the user is able to set parameters for the position, size of the investigated area and time, depending on mode and aims.

In figure 3.2, the basic steps within the settings are shown. The chosen mode will determine the process of the algorithm. These three modes are eligible: trajectory, single point or grid mode.

In trajectory mode, a trajectory in UTM coordinates is requested as well as time stamps that match the coordinate points. The algorithm will treat each point of the trajectory separately to determine the LOS satellites.

In single point mode, a static position is analyzed in terms of visibility. The surrounding area and time extent are set as desired.

For grid mode, due to runtime and memory space only one epoch is analyzed. The output here, unlike other modes, is a matrix of the size of the grid stating the number of visible satellites. Important for the user to choose is the grid resolution as this will affect the runtime immensely. It is reasonable not to set the resolution too high since the number of visible satellites does not change between points that are set very closely and the accuracy of the city model is not at the same level.

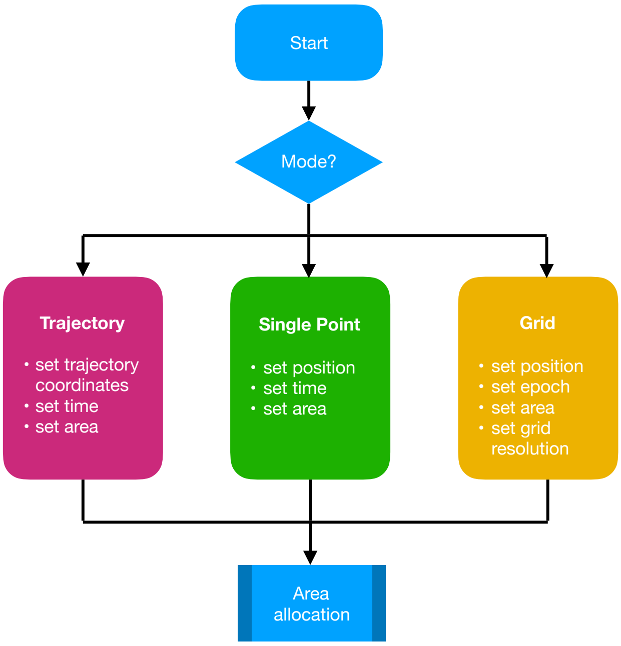


Figure 3.2: Algorithm structure for settings part in the very beginning of the program. The user is able to choose the processing mode.

As of experience, a grid resolution of one point per two square meters is the highest feasible one considering the accuracy of ±1 *m* of LoD2 (cf. section 2.1.1) and the fact that interpolation between points is applied afterwards. For a fast overview of the visibility situation in a specific area, the user should downgrade the resolution to get a faster result. Another reason to downgrade the resolution can be a big area since the runtime increases with the number of grid points that are treated.

Area Allocation After setting the above mentioned parameters, the area has to be allocated to the corresponding GML files.

The coordinate frame used in the CityModel GML files is UTM32 based on ETRS89 (Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation, 2017). The city of Hanover is separated into 215 GML files which each represent a so-called tile as seen in figure 3.3. All tiles are named after their position w.r.t. UTM coordinates, e.g. the file *5490_5803.gml* represents the region for all UTM coordinates from 549000 m Easting, 5803000 m Northing to 549999 m Easting, 5803999 m Northing.

To find the tiles the specified region lies in, the corner points of the area are extracted. Then the corresponding tiles are selected by reducing the coordinates of the corner points to the first four digits.

3.1.2 Data Preparation

Previous to the actual calculation whether a satellite is in sight or not, all the data needs to be prepared. This especially means organizing the CityModel in terms of reading, processing and ordering the CityModel. The preparation is important in order to reduce runtime and improve the program's performance.

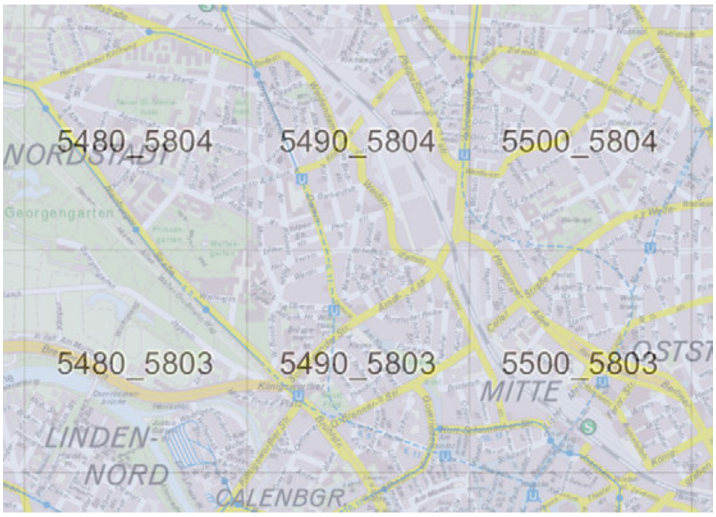


Figure 3.3: An extract of the overview for all tiles of Hannover. The dimension of each tile is approximately 1 km^2. The four digits represent the UTM coordinates in East and North (Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation, 2017).

Reading GML files

Reading a GML file is very similar to reading XML data which is the reason the CityModel is imported using a function that is based on *XML2struct.m* provided by Falkena (2012) on MathWorks. The provided function is itself based on Matlab's included function *readXML.m*. The output is from data type struct which allows the user to browse all the data that is written inside of the GML file with respect to the folder that it is allocated to.

In this step, all the tiles from the area allocation step in settings are read.

After reading the raw data, the output is a structure array named *CityModel* that contains all the information in a nested character. To get an idea how to navigate within the struct, the order is as follows.

Listing 3.1: Contents of CityModel in CityGML.

```
CityModel
        gml_colon_boundedBy
                gml_colon_Envelope
                        gml_colon_lowerCorner
                        gml_colon_upperCorner
                        Attributes
        Attributes
        cityObjectMember
```

The field `gml_colon_Envelope` gives information about the extent of the CityModel tile. Attributes always contain additional information like IDs. The field of interest for the purpose of this work now is `cityObjectMember`, in which all the buildings of the tile are stored in.

Listing 3.2: Contents of cityObjectMember.

```
cityObjectMember
        bldg_colon_Building
                gml_colon_name
                bldg_colon_roofType
                Attributes
                bldg_colon_boundedBy
                        bldg_colon_WallSurface
                        bldg_colon_RoofSurface
                        bldg_colon_GroundSurface
```

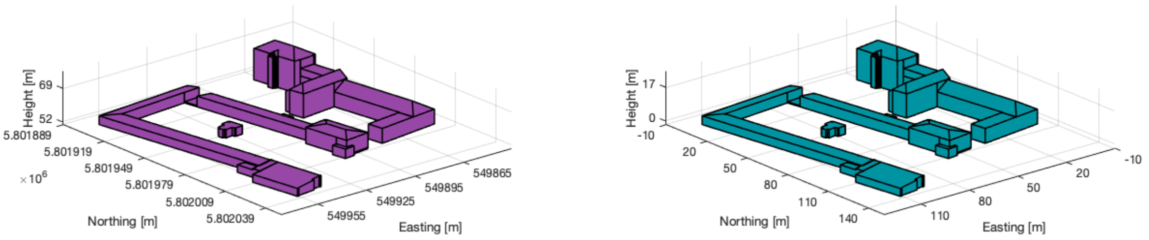
The field `bldg_colon_boundedBy` now contains all surfaces that describe the building. These surfaces can either be a wall, roof or ground. Every building consists of at least one of each. If we now go deeper into the surfaces we finally get the polygon coordinates.

The function *readGMLwithstruct.m* filters the structure of the CityModel in a way that the output is a cell that contains the buildings on its first layer and the surfaces of each building in the second layer.

After all tiles inside the area are read, an algorithm checks if a building from each tile is inside the specified bounding box. This is done by taking only the ground surfaces of the building: if a vertex of the ground surface lies within the x- and y-dimension of the area, the whole building is assigned to the area and hence is in further consideration of the LOS/NLOS calculation.

Transformation to local coordinates

The coordinate frame used in the original GML files is UTM32 based on ETRS89 together with heights referring to *Deutsches Haupthöhennetz 1992* (DHHN92) (Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation, 2017). For later calculations and faster comprehension within the model, the coordinates are reduced to the in the GML file preset *lowerCorner* of the bounding box with the lowest value in Easting, Northing and height. The lower corner hence becomes the local origin.



(a) Test file CityModel in given UTM coordinates.

(b) Test file CityModel transformed to local coordinates.

Figure 3.4: CityModel with different coordinate systems.

Box subdivision

In order to have less calculations and hence faster runtimes, the data i.e. the polygons are divided into a box structure. This later helps to handle less polygons at once when calculating intersections between rays and polygons.

To do this, we imagine the whole city model, i.e. all polygons, to be within a cuboid. The cuboid then is subdivided into smaller equally sized cuboids which we call boxes. A polygon that lies inside of a box is assigned to that specific box, however if that polygon has vertices beyond a single box, it is assigned to all boxes that have a vertex inside.

The result will be a three-dimensional cell with the size of the number of boxes that has a list for each box specifying which polygons are inside.

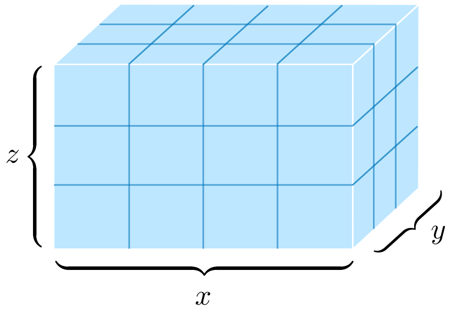


Figure 3.5: Model of box division. x, y and z represent the number of boxes in each direction.

To set the number of boxes in each direction, we consider the average number of polygons p_{box} a box should contain. This is then the parameter to control the computing time. The parameter p_{box} combined with the given total number of polygons p will lead to a first approximate total number of boxes B' which is also equal to the product of the number of boxes in each direction x, y and z.

$$B' = \frac{p}{p_{box}} = x \cdot y \cdot z \tag{3.1}$$

Furthermore, we want the number of boxes x and y to be in equal proportion to the extent δ_x, δ_y in these directions, taken from the maximum and minimum vertex of all polygons. The number of boxes z in the direction of height though can be fixed to a certain value because the extent here is known to a good approximate and does not change much from tile to tile.

This leads us to

$$\frac{x'}{y'} = \frac{\delta_x}{\delta_y} \Rightarrow y' = \frac{\delta_y}{\delta_x} \cdot x' \tag{3.2}$$

with preliminary x' and y' and inserted into 3.1 we get

$$B' = z \cdot x'^2 \cdot \frac{\delta_y}{\delta_x} \tag{3.3}$$

$$x' = \sqrt{\frac{B'}{z} \cdot \frac{\delta_x}{\delta_y}}\,, \quad y' = \frac{B'}{z \cdot e} \tag{3.4}$$

$$x = \lfloor x' \rfloor\,, \quad y = \lceil y' \rceil. \tag{3.5}$$

The reason for the *floor* and *ceil* operations is to round x and y to the nearest integer number but not letting the final

$$B = x \cdot y \cdot z \tag{3.6}$$

differ too much from its original B'.

After the number of boxes has been set, every polygon is allocated to the box it lies in. The decision is made depending on whether at least one of the vertices is within the box. Then the polygon ID is assigned to the respective box. The result of this computation step is a cell with the size corresponding to the number of boxes in each direction. Every entry inside of the cell is containing the IDs of the allocated polygons.

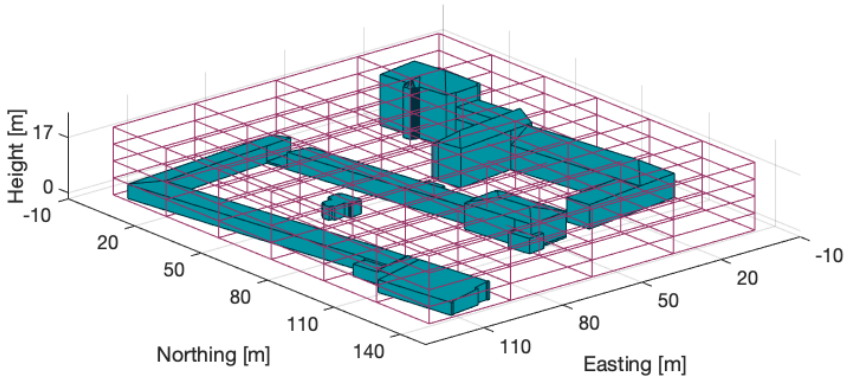


Figure 3.6: CityGML test file sectioned into boxes.

Polygon Triangulation

Another part of the data preparation is the triangulation of all polygons. This is done in order to have a faster runtime especially when calculating intersections of rays and polygons and also to have an easier handling with the polygons. With triangles, we do not need to think about convexity or the number of vertices of a polygon.

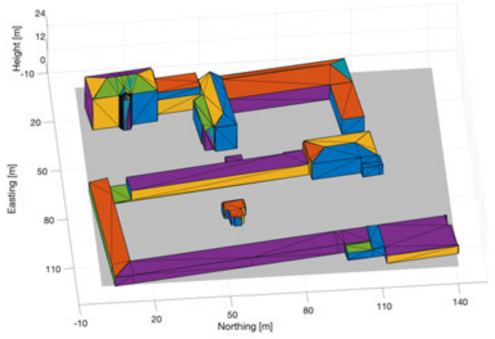


Figure 3.7: CityGML test file with triangulated polygons.

The triangulation is done with Matlab's *triangulation* function (MATLAB, 2019) where the input is a *polyshape* object that can be created using the *polyshape* function. Basically, the coordinates of a polygon are transformed to a polyshape which then can be triangulated.

Since those functions can only work in two dimensions, the vertices of the polygons have to be reduced to one plane. On which plane the triangulation should be executed is determined by finding the plane in which the 2D-coordinates are never identical. This ensures that all vertices are considered. Also, when using the function *polyshape* it has to be paid attention to the fact that the vertex alteration controlled by the *'Simplify'* property is set to false as well as the *'KeepCollinearPoints'* property is set to true. Otherwise, Matlab will remove vertices which later can result in holes in the 3D representation.

After the triangulation, the resulting connectivity list of the vertices can be used to link the vertices not only in 2D but in 3D. This is then used as the final triangulation, see figure 3.7.

3.1.3 LOS/NLOS Calculation

In the final stage, a satellite is marked either as visible or not visible. This is done by first getting the line-of-sight (LOS) vector and ray for every available satellite and a test if this ray is intersecting anything, see the schematic overview in figure 3.8. Since the CityModel has already been prepared, the test is done in two hierarchic steps to save time. First, the ray is tested in order to find which boxes it is intersecting. Secondly, the ray is tested to find intersecting polygons inside of the respective boxes.

If a ray intersects with any polygon in any box, it is not visible, i.e. NLOS.

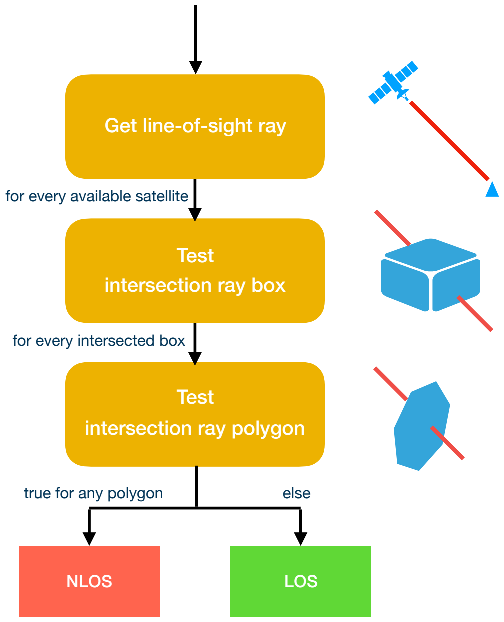


Figure 3.8: LOS/NLOS calculation process.

Line-Of-Sight

In order to test if a satellite is visible, the LOS vector has to be established at first. Therefore, we take the satellite coordinates derived from a broadcast file of a desired day to a specified time and transform the UTM coordinates of the GML file to ECEF coordinates. The resulting LOS vector in ECEF is transformed to a LOS vector in a topocentric sense where the azimuth and elevation of each satellite can be calculated (cf. section 2.3.1).

All satellites that have an elevation above zero are categorized as *available* as they show up over the horizon. Only these are considered for further studies.

For later computations, the LOS vector is also normalized to a unit vector to apply it as the direction D between user and satellite. The ray then forms itself by taking the origin O which is the user position and the direction D.

Intersection of Ray and Box

To reduce runtime, the hierarchical box structure now comes into action. As we do not want to test if an intersection takes place between a satellite ray and every polygon in the whole city model, we first only test which boxes the ray is intersecting.

For this purpose, the algorithm of Williams et al. (2005) in an implemented Matlab form is integrated (cf. section 2.2.1). As the author claims, the ray-box intersection algorithm is both efficient and robust and an improvement of Smits (2002) in terms of ray slopes that are near zero.

If the outcome for the intersection of a box is true, all polygons that have been assigned to that box are marked as *considered* so that in the next calculation step only this selection of polygons is taken into account.

Intersection of Ray and Polygon

The calculation of the intersection between a ray and a polygon is based on the triangulation that has been done during data preparation. This makes the computation easier and hence reduces the run-time.

For every triangle of a *considered* polygon, the intersection is tested. This is done by applying the function *TriangleRayIntersection.m* provided by Tuszynski (2018). It is based on the algorithm by Möller and Trumbore (2005) described in section 2.2.2.

A previously implemented point-in-polygon test that also needs plane calculations has been proven to be too slow as multiple consecutive computations have to be made instead of only one in the case of triangle ray intersection.

If any of the triangles is crossed by the ray, the test for the polygon ray intersection is set to true. And if the outcome of the intersection test is true for any of the polygons, the satellite is set to not visible (NLOS).

The test is repeated for each of the satellite rays.

3.1.4 Output

Satellite Visibility

For both single point mode and trajectory mode, the output for the satellite visibility will be a LOS or NLOS matrix marking all visible and non-visible satellites for each epoch.

In figure 3.9, visible satellites are colored green and and non-visible ones red for a single epoch in single point mode according to the visibility matrix. It can be seen that all of the red satellite rays cross a building. The PRN of the rays is noted in the legend of the figure to see which PRNs are in sight.

In grid mode, it is possible to create maps. If the user requests to have a map of the satellite visibility, all the calculation steps to determine the number of visible satellites from the single point mode are applied for every grid point. By doing so, a visibility matrix is created. The satellite visibility is then color coded and displayed for the requested area.

In figure 3.10, the satellite visibility is exemplarily shown for the test file with a resolution of one point every 2 m. The date and time has been set randomly to GPS week 1759 and GPS seconds of week 299754. Furthermore, 2500 points on the ground have been calculated to get a feasible density for the resulting heat map.

The resolution is still visible as the edges and interpolated areas are clearly detectable. Nevertheless, the heat map can give a good overview on which positions are heavily obstructed and which ones are barely affected.

What should be neglected in the heat map are the outer borders since it does not show outside buildings that could obstruct these, eg. position 50 m East, 140 m North in figure 3.10.

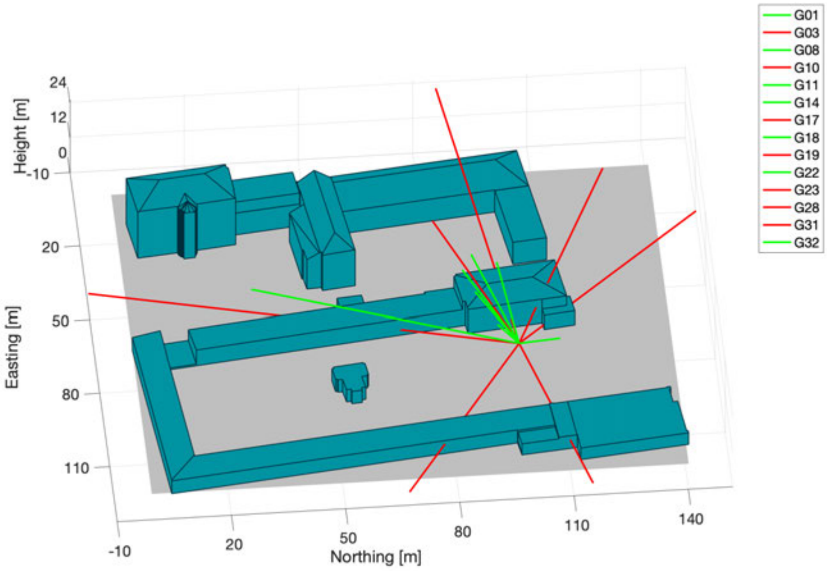


Figure 3.9: LOS (green) and NLOS (red) satellite rays within the CityGML test file.

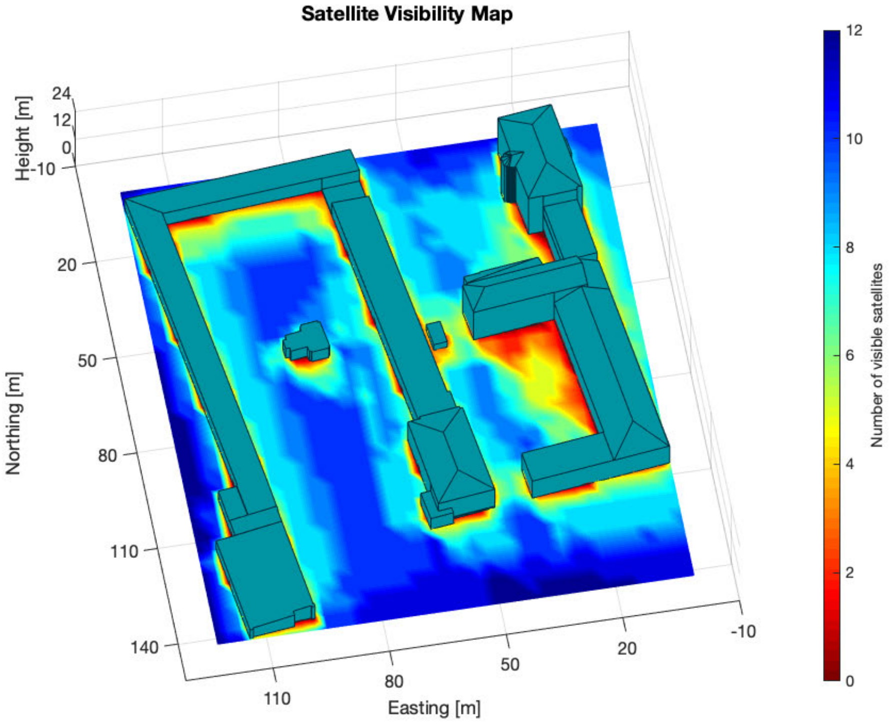


Figure 3.10: Color coded satellite visibility for the CityGML test file. Blue color tones refer to a high satellite visibility while red ones show a lower visibility.

DOP values

Instead of illustrating the satellite visibility, the program is able to show DOP values of LOS satellites only as well.

To achieve this, for every grid point the LOS values from the visible satellites are taken to compute a design matrix A and subsequently taking the DOP values from it. In the event where $A' \cdot A$ becomes near singular, the inverse of that product to get the cofactor matrix Q_x is not calculated so that a real DOP value is guaranteed. This is especially the case when less than four satellites are visible.

In figure 3.11, the GDOP map for the same point of time and with the same resolution as the visibility in figure 3.10 is shown. The mentioned events where a DOP calculation is not possible are not mapped and hence stay white. In order to distinguish low GDOP values in color, the colorbar has been limited to a constant maximum of 5 so that the resolution for low, good DOP values stays high. This means, all DOPs greater than this limit are colored in the same dark red.

Again, values at the outer borders should be neglected because of possible obstructions outside of the visualized area.

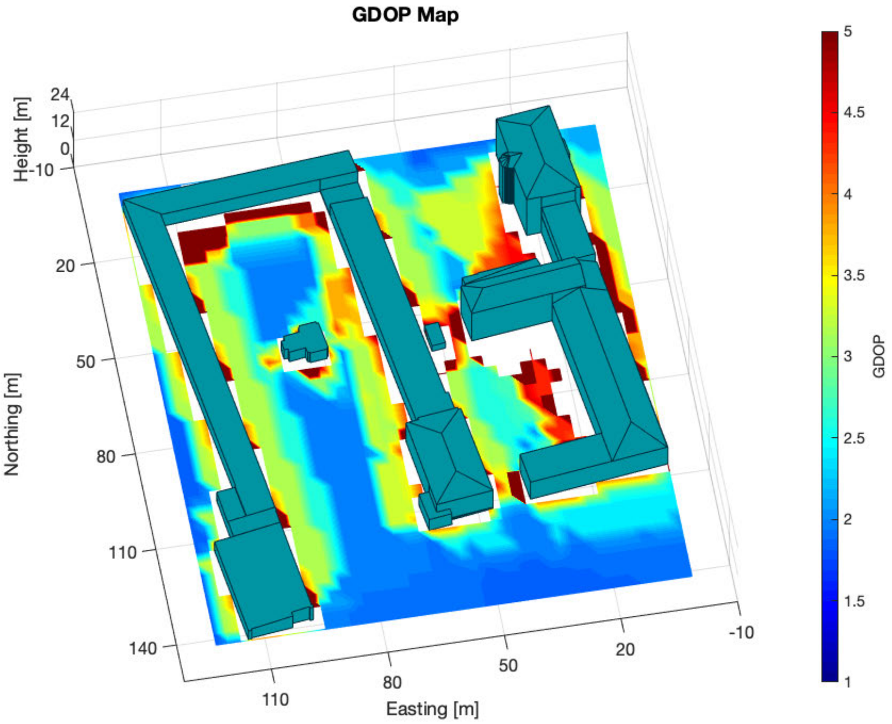


Figure 3.11: Color coded GDOP values for the CityGML test file. In white areas a calculation of DOP values was not possible because of an insufficient number of visible satellites.

3.2 Signal Reflection

To compute detour lengths of the multipath signal and to set up Fresnel ellipses, the identification of reflection points is required. One can distinguish between reflections on the ground and reflections on building planes which are oriented arbitrarily. Both cases are handled differently.

3.2.1 Horizontal ground reflection

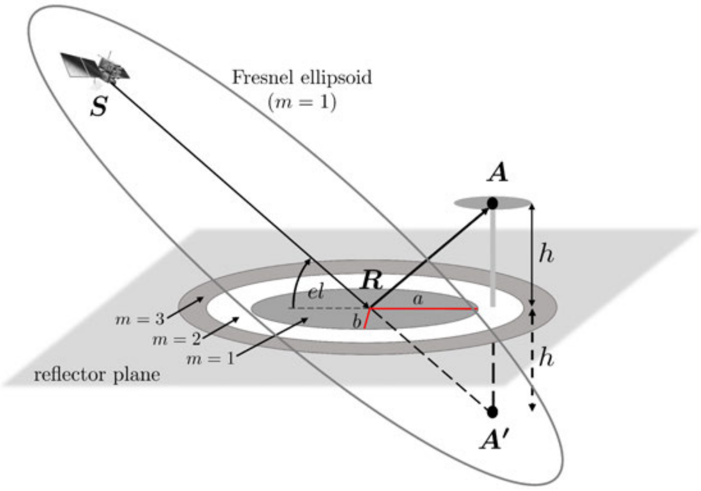


Figure 3.12: Horizontal ground reflection of a signal between satellite point $\boldsymbol{S}$ and mirrored antenna point $\boldsymbol{A}$ (Zimmermann et al., 2018).

In case of a ground reflection, the coordinates of the reflection point $\boldsymbol{R}$ can be determined directly from the coordinates of the antenna point $\boldsymbol{A}$ and satellite point $\boldsymbol{S}$ (cf. figure 3.12) using the formulas given in section 2.3.3. It then has to be tested if the line of sight between $\boldsymbol{S}$ and $\boldsymbol{R}$ as well as between $\boldsymbol{A}$ and $\boldsymbol{R}$ is free of obstruction using the function that has already been used in the programming of the Urban Trench Model (cf. section 3.1).

This concludes to a programming sequence of the following. For all satellite positions $\boldsymbol{S}$ do

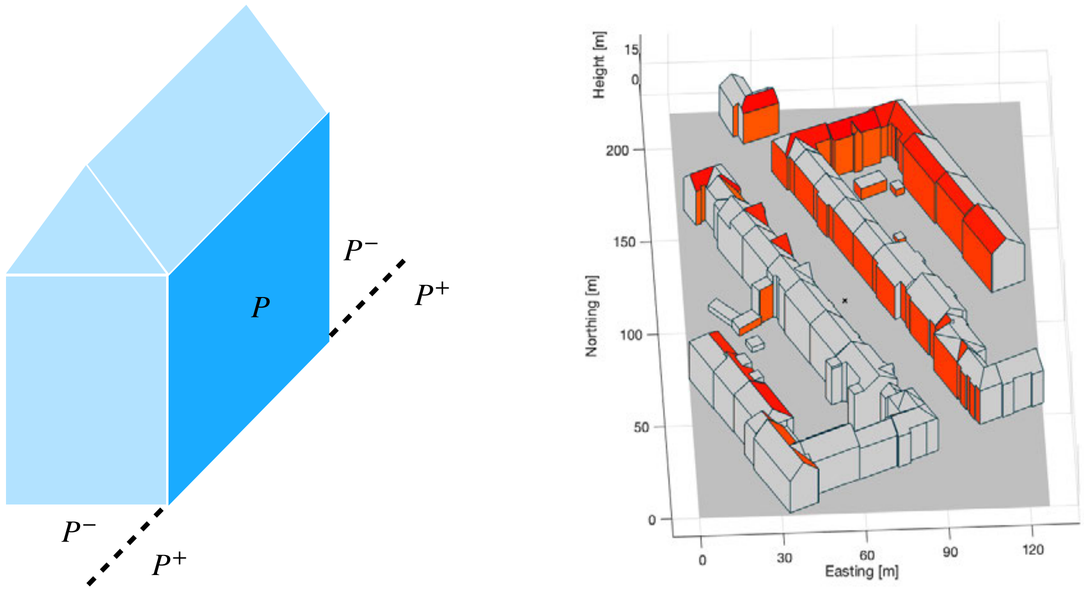
1. Compute coordinates of $\boldsymbol{R}$
2. Test if $\overline{\boldsymbol{SR}}$ is free of obstruction
3. Test if $\overline{\boldsymbol{RA}}$ is free of obstruction

Afterwards, the ellipse parameters and detour lengths can be determined.

3.2.2 Reflection on building planes

Reflections on building planes are more complicated to compute as the planes are arbitrarily oriented.

Focusing on this example, it first has to be tested whether a reflection on the plane P is possible at all. This means, both points of satellite and antenna have to be on the same outer side of the plane that is described as P^{+}, cf. figure 3.13a. In the application of the CityModel, a reflection can take place only on the outside of a building plane.



(a) Plane sides P^+ and P^-. In order for a reflection to happen, both satellite and antenna point $\boldsymbol{S}$, $\boldsymbol{A}$ have to be in P^+.

(b) Possible reflection planes for a simulated point. All red planes have at least one satellite and the user position (cross) in P^+ over the course of a day.

Figure 3.13: Plane sides P^+ and P^- schematically (a) and applied to the CityModel (b).

To test if both $\boldsymbol{S} \in P^+$ and $\boldsymbol{A} \in P^+$ the normal vector of every plane has to be determined first (cf. section 3.2.2). Then, the check is performed by calculating the dot product of

$$v_S = \vec{\boldsymbol{p}} \cdot \vec{\boldsymbol{n}} \tag{3.7}$$

$$v_A = \vec{\boldsymbol{q}} \cdot \vec{\boldsymbol{n}} \tag{3.8}$$

where $\vec{\boldsymbol{p}}$ denotes the vector from the reflection point $\boldsymbol{R}$ to $\boldsymbol{S}$ and $\vec{\boldsymbol{q}}$ denotes the vector from $\boldsymbol{R}$ to $\boldsymbol{A}$. If $v_S > 0$ and $v_A > 0$ which refer to acute angles between the vectors, both points $\boldsymbol{S}$ and $\boldsymbol{A}$ are in P^+.

As the coordinates of the reflection point $\boldsymbol{R}$ are calculated using the formulas described in section 2.3.3, a check on $\boldsymbol{R}$ is required to define if $\boldsymbol{R}$ lies inside the polygon that spans the plane P or not. This is obtained by applying the point-in-polygon test described in section 2.2.3.

Finally, we conduct the check if both $\overline{\boldsymbol{SR}}$ and $\overline{\boldsymbol{AR}}$ are free of obstruction.

The programming sequence for determining the reflection points on building planes is summarized in the following. For all planes P and for all satellite positions $\boldsymbol{S}$ do

1. Test if $\boldsymbol{S} \in P^+$ and $\boldsymbol{A} \in P^+$
2. Compute coordinates of $\boldsymbol{R}$
3. Test if $\boldsymbol{R} \in P$
4. Test if $\overline{\boldsymbol{SR}}$ is free of obstruction
5. Test if $\overline{\boldsymbol{RA}}$ is free of obstruction

Determining the normal vector of a 3D-polygon

Generally, the polygons of the 3D CityModel are oriented in a counter-clockwise order so that a normal vector of three consecutive points will point outward of the building. However, when determining the normal vector of a 3D-polygon by calculating the cross product of vectors to three points, some problems still might occur (cf. figure 3.14):

1. The three points might be collinear, resulting in an arbitrary normal vector.
2. Since the polygons are not necessarily convex the three points might surround an inward edge resulting in the normal vector pointing in the opposite direction.

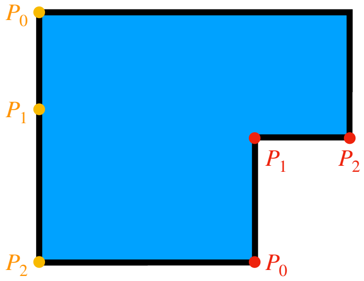


Figure 3.14: Fatal point choice for normal vector calculation even though the counter-clockwise orientation is applied. Yellow points P_0, P_1 and P_2 are collinear, red points P_0, P_1 and P_2 wrap around an inward edge.

To avoid these problems the normal vector has to be be determined using a well-shaped triangle consisting of three polygon vertices. This way, an edge can never point inward. Also, the Matlab inbuilt *triangulation* function does not triangulate collinear points.

What has to be paid attention to is the orientation of the polygon points in their 2D representation which is necessary for the triangulation. Additionally to the 3D to 2D conversion of the points as described in section 3.1.2, a test has to be performed to determine whether the points are ordered clockwise or counter-clockwise.

The test uses a simple formula that primarily is used to calculate a polygon's surface area (Bender, 2010),

$$A = \frac{1}{2} \sum_{i=1}^{n} y_i \left(x_{i-1} - x_{i+1} \right) \tag{3.9}$$

but can also be applied for the orientation test. x_i and y_i denote the 2D coordinates of polygon points P_i. If A is negative the points are ordered clockwise, if positive then counter-clockwise.

As the 3D polygons have a counter-clockwise point order, the 2D coordinates are mirrored if A is negative. Then, the triangulation algorithm will form triangles with points in the correct order.

Finally, the normal vector $\vec{n}$ can be computed using

$$\vec{n} = \overrightarrow{P_0 P_1} \times \overrightarrow{P_0 P_2} \tag{3.10}$$

with P_0, P_1 and P_2 forming the first, second and third vertex in a triangle.

3.3 Outlook

At some points during the programming work ideas came up that could improve the workflow but have not been implemented for time reasons. Nevertheless, they are stated here to consider them in the future.

3.3.1 More Efficient Polygon Representation

During the work with the GML CityModel it turned out that many polygons are represented inefficiently, i.e. containing collinear points. Storing collinear points as vertices for polygons is simply unnecessary and should therefore be sorted out in advance. Moreover, collinear points do not only use up memory space and cause additional runtime but can also result in problems e.g. for computing normal vectors or converting 3D polygons into a 2D representation.

Sorting out collinear points can be done by checking iteratively if three consecutive vertices are collinear. A stable check is done by setting up a matrix of those three points and calculating the rank of that matrix. If the rank is smaller than 2, the points are collinear (Csati, 2017). If the test outcome is true, the middle point will be deleted.

3.3.2 Application of a Hashmap

A hash map or hash table is a data structure that can link keys to its corresponding values. A hash function behind it will compute the respective index to the buckets where the specific values can be found. Hash maps are more effective than other data structures such as search trees in terms of space and searching cost (Cormen et al., 2001).

For the purpose of the ray intersection algorithm, a hash map could be implemented as a chained hash map with the first table linking the PRN numbers of the satellites to the box numbers they are intersecting and the second table linking the box numbers to the polygon IDs they contain (cf. table 3.1).

Table 3.1: Idea for application of chained hash tables.

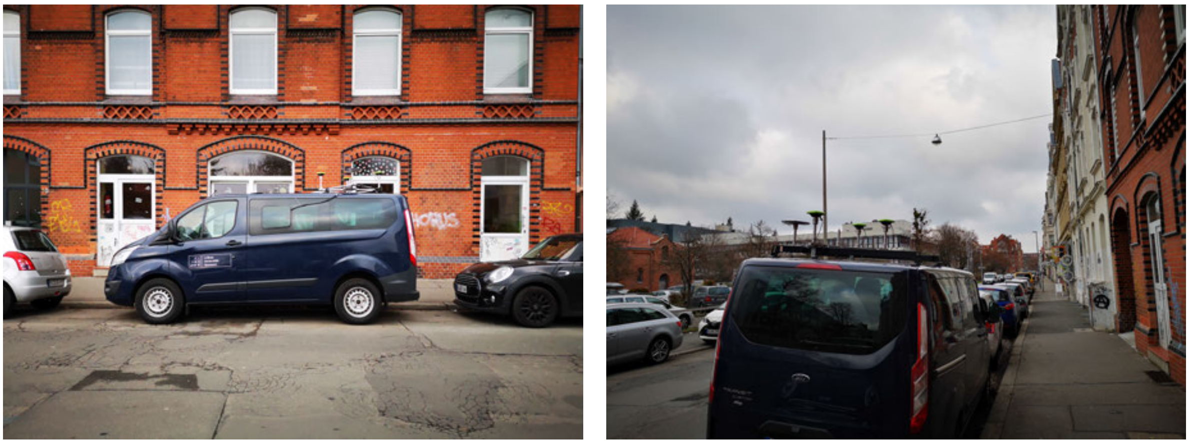
	Key	Value
First table	PRN number	Intersected boxes
Second table	Box number	Contained polygons

4 Static Experiment Analysis

To test the implementation and concept of chapter 3, a data set from Hartmann (2019) is studied and investigated, checking satellite visibilities and the performance in a static pre-analysis. Data has been collected under urban conditions with high potential of multipath and diffraction.

4.1 Experiment

In this experiment, geodetic grade (Javad) antennas are combined together with a geodetic (Javad) and consumer market (u-blox) receiver. The antennas are mounted on a calibrated metallic cross bar on a van which is parked at the side of Schneiderberg street across a parking lot. On the east side of the car apartment, buildings are located (see figure 4.1a) while the left side does not show any buildings close-by (see figure 4.1b). Both receivers have collected data for a time period of three hours on day of year (DOY) 348 in 2018.



(a) Apartment buildings in the east of the antennas

(b) Street of Schneiderberg, view to the north

Figure 4.1: Static experiment setup in the urban surroundings of Schneiderberg in Hanover (Hartmann, 2019).

4.2 Analysis of Satellite Visibilities

For the time the receivers recorded data, a satellite visibility analysis has been performed using the Urban Trench algorithm described in section 3.1.

The algorithm has been performed for an area of 250 m x 150 m around the user's position and for the same interval of one second as it is the measurement frequency itself. A dimension of this size is enough as apartment buildings outside this area would only come in reach of 10 ° elevation in which many satellites could not be tracked anyhow. The time interval has been chosen to match the observed data and the simulated satellite visibilities.

Figure 4.2 shows the user position and satellite relation at 11:16 a.m. GPS time. Generally, it is noticeable that satellites in the east cannot be seen because of the buildings while satellites in the west are visible

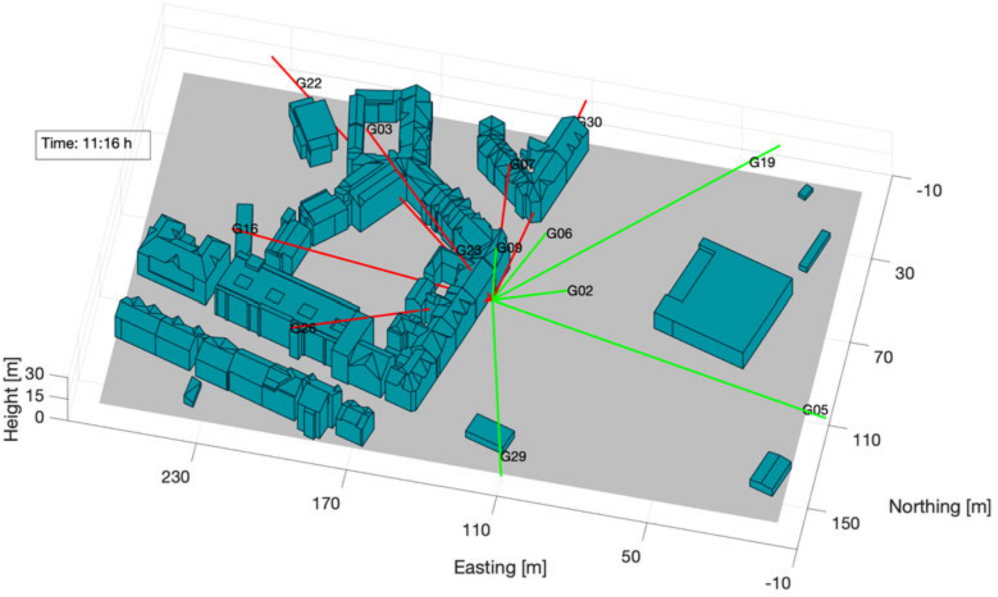
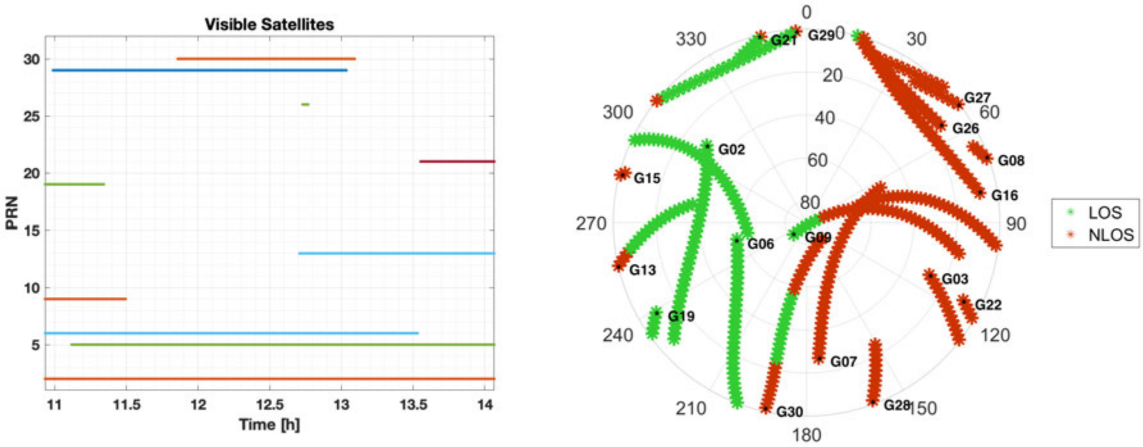


Figure 4.2: Satellite rays in LOS (green) and NLOS (red) for static experiment in Hanover Nordstadt at DOY 348 in 2018. Note that the view is towards south.

above a very low elevation due to the parking lot. And since the antenna is mounted on top of a van, all the other cars along the street should not affect the direct signal.



(a) LOS satellites PRN wise.

(b) Skyplot, distinguished between LOS and NLOS.

Figure 4.3: Satellite visibility graphs classified into PRN numbers and as a skyplot.

Figure 4.3 shows a general view over the visible satellites for the whole measurement period. The skyplot in 4.3b underlines the statement of visibility to the west and obstructions in the east. Note that satellite G30 is somehow specific as the LOS azimuth angle is very close to the street's azimuth angle. This then results in a tight decision whether the satellite is visible or not. It has to be reminded that the building model's accuracy is given as ±1 m which means that these close decisions could also be wrong.

4.3 Comparison with Observed Data

4.3.1 Number of visible and observed satellites

For the comparison between computed satellite visibilities and observed ones, the receiver outputs from both Javad Sigma and u-blox have been taken into account.

Table 4.1: Average, minimum and maximum number of satellites for static experiment.

	Average	Minimum	Maximum
# LOS satellites	4.7	3	7
# Javad observations	5.1	2	7
# u-blox observations	7.9	4	11
# satellites over horizon	11.3	10	13

In figure 4.4, the number of satellites calculated as visible is marked in blue, the observed number of satellites for Javad and u-blox in yellow and red respectively. To have an overview over the actual available number of satellites over the horizon, this number is colored in purple. In table 4.1, key parameters of the satellite visibilities are summarized additionally. Apparently, the urban condition reduces the total number of available satellites to an average of 42%. It is very obvious that the performance of GNSS in urban conditions therefore is highly reduced, even without thinking about reflections or diffractions.

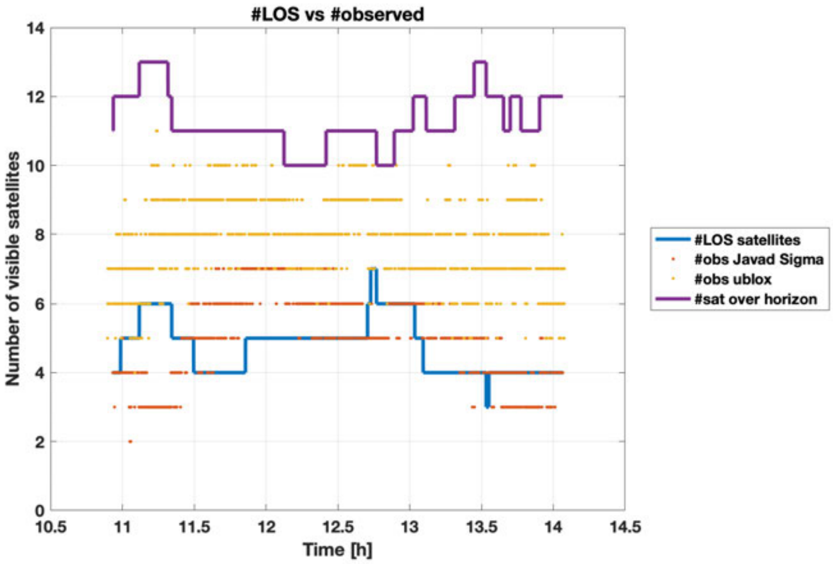


Figure 4.4: Comparison between observed and computed satellite visibility observed for the time of the static experiment.

Hartmann (2019) previously noted that mass market u-blox receivers track significantly more satellites than the geodetic receivers. The number usually varies around three more satellites that are tracked at the u-blox receiver but can also reach eight more as it is the case at 11:15.

This is due to the different tracking loops of the receivers. These need a certain signal strength to track the signal over noise and other interfering signals (Rao et al., 2013). The lowest C/N$_0$ the receiver is capable to track is the threshold for the tracking loop. The threshold for the Javad receiver is higher than the one of u-blox as the receivers use different carrier loop noise bandwidth parameters. If one applied the same parameters for Javad receiver, more signals will be detected as less signals fall below the threshold.

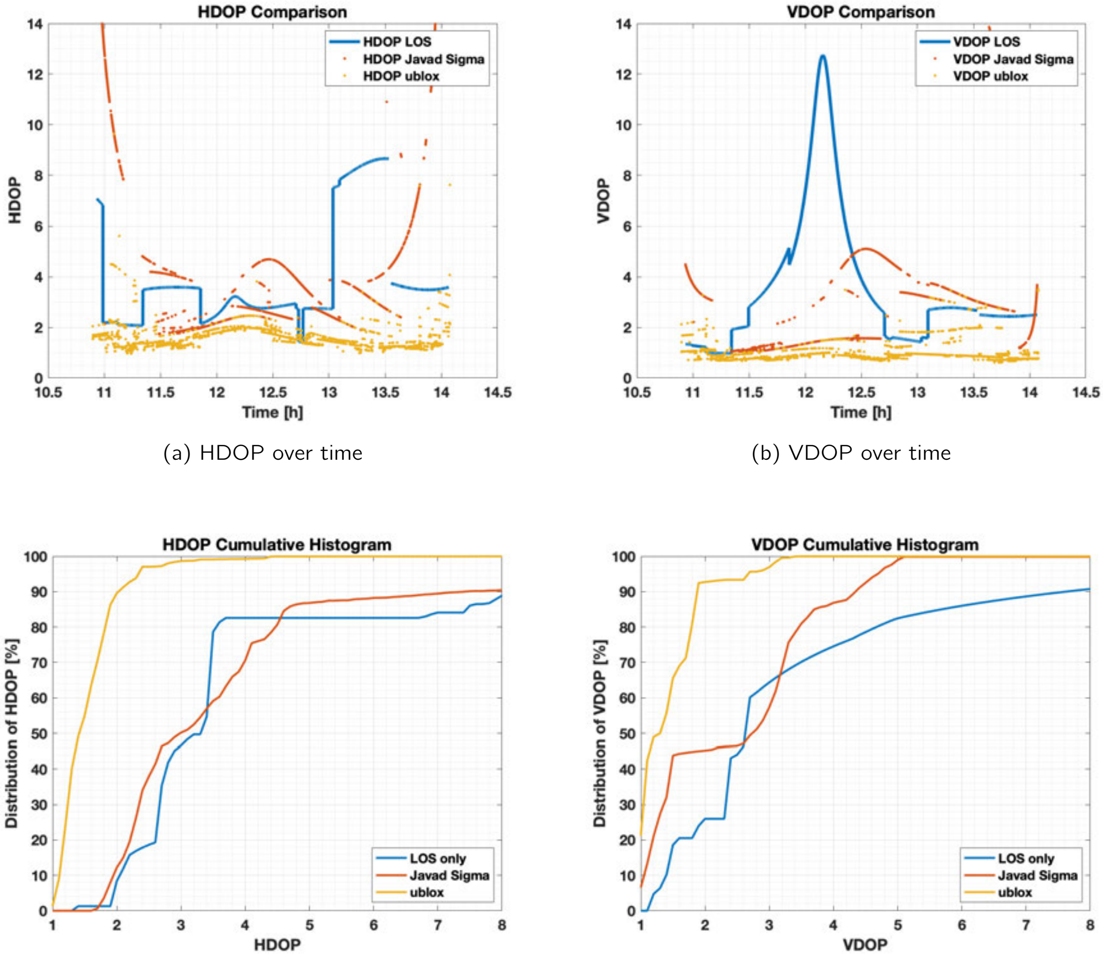
In comparison with the number of actually visible LOS satellites, u-blox always tracks an equal number or more. In some cases, it receives the signal of 6 more satellites than actually visible, the average difference between the numbers is $\bar{d} = 3.2$. At some epochs, the u-blox receiver even tracks all satellites over the horizon except for one (e.g. sometimes between 11:30 and 12:00 GPS time).

This is different for the Javad receiver. The number of tracked signals is mostly around the actual number of visible satellites, in average $\bar{d} = 0.4$. There even are times where less satellites are tracked than there are available. And even worse, between 11:00 and 11:30 as well as between 13:00 and 14:00 less than four satellite signals could be received which will result in problems when determining a position.

4.3.2 DOP Values

Figures 4.5a to 4.5d show a comparison of the respective Dilution of Precision (DOP) values for the observed satellites of Javad and u-blox receivers and for LOS satellites only. The time component of DOP is not relevant for this analysis. Hence, only the horizontal (HDOP) and vertical (VDOP) are analyzed here.

The urban condition has a significant impact on visibilities and therefore the LOS DOP reaches unfeasible high values at certain times. Furthermore, the DOP can increase suddenly by a high factor as satellites



(a) HDOP over time

(b) VDOP over time

(c) HDOP distribution

(d) VDOP distribution

Figure 4.5: Comparison of DOP values for observed satellites from the static experiment with DOP values for visible satellites only.

Table 4.2: Average, minimum and maximum HDOP and VDOP for the static experiment.

	Average	Minimum	Maximum
HDOP LOS	3.9	1.4	8.7
HDOP u-blox	1.7	1.0	54.7
HDOP Javad	4.8	1.7	53.9
VDOP LOS	3.5	1.0	12.7
VDOP u-blox	1.0	0.6	3.6
VDOP Javad	2.6	1.0	15.2

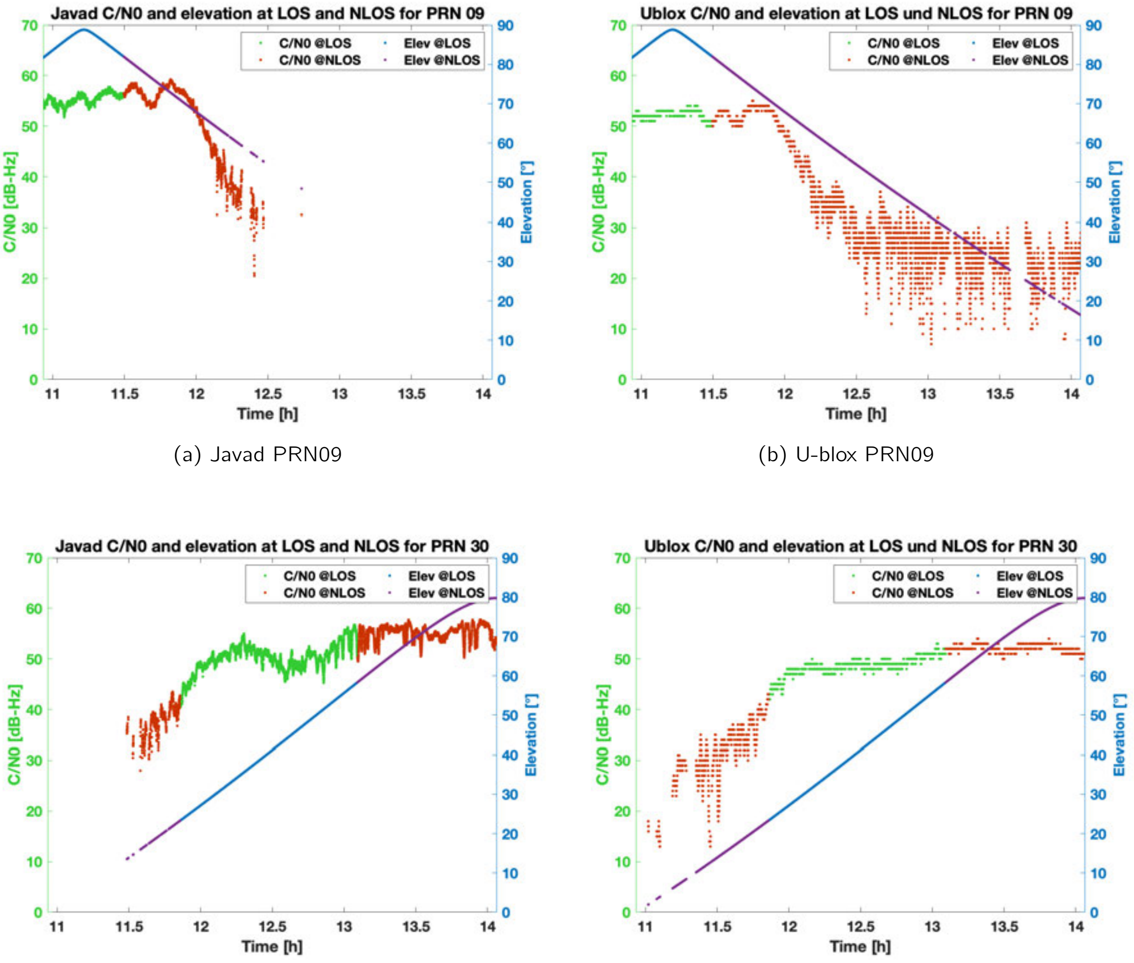
can become not visible even at high elevations. Such a jump appears in HDOP shortly after 13:00 when the number of visible satellites decreases to four (cf. figure 4.5a).

The HDOP values for the Javad receiver generally are in the same range as those of LOS only which corresponds to the findings in section 4.3.1 in which the number of tracked satellites is not too far apart from the number of LOS satellites. For VDOP, the Javad observations provide overall better VDOPs which can be especially seen in the corresponding cumulative histogram in figure 4.5d.

The cumulative histograms show in particular the dominance of the u-blox receiver in terms of low DOP values. Both HDOP and VDOP are significantly better, in both cases 90% of the values are below 2. In contrast to that, only 8% of the LOS only HDOPs reach a level lower than 2. The low DOP values directly anticorrelate to the high number of signals that are tracked throughout the measurement period.

The highest VDOP calculated for the u-blox receiver does not even surpass a value of 3.6 which is close to the mean of the actual VDOP of LOS only, cf. table 4.2. On the contrary, the mean value of u-blox VDOPs is 1.0 which can be considered as a quite good optimum.

4.3.3 Signal Strength



(a) Javad PRN09

(b) U-blox PRN09

(c) Javad PRN30

(d) U-blox PRN30

Figure 4.6: Comparison of C/N$_0$ values.

Comparison between receivers

Figure 4.6 exemplarily shows the signal strength C/N_0 for satellites PRN 09 and PRN 30 recorded with Javad (4.6a and 4.6c) and u-blox receiver (4.6b and 4.6d). These two satellites have been chosen because they both have LOS and NLOS phases.

The lines are color-coded for visible periods and non-visible periods . In all these examples, signals could be tracked in NLOS phases as well. The C/N_0 resolution is 0.25 dB-Hz for Javad and 1 dB-Hz for u-blox.

What can be further derived from the figures is that the lowest signal strength level the Javad receiver is able to process is around 30 dB-Hz. For u-blox, the lowest C/N_0 values go down to 10 dB-Hz.

This difference leads to big disparities in terms of tracking length. While PRN 09 is only tracked until 12:30 for the Javad receiver, u-blox records phase measurements for almost two more hours. PRN 30 is tracked by u-blox nearly since the satellite's rise over the horizon whereas Javad starts receiving its signal half an hour later.

For the period of 11:00 to 12:00 for PRN 09, the signal strength is oscillating. This is an indicator for multipath. Since the satellite is at a very high elevation during this period, it is possible that the reflection comes from the street surface.

Satellite PRN 30 shows continuously high C/N_0 values above 50 dB-Hz even during its second NLOS phase. As discussed in section 4.2, this satellite's azimuth is close to the street's orientation. The intersection decision of satellite ray and building is very close as the ray barely touches the apartment building. This means, either the satellite is actually directly visible and the NLOS determination is wrong because of the low building height accuracy, or the satellite is indeed diffracted but that does not reduce the C/N_0.

If the last aspect was true, this would mean that a rating on the signal quality based on the C/N_0 values only is critical as sometimes a closely diffracted signal does not reduce the signal strength significantly enough to be detected and then downgraded.

Oscillating Signal Strength at LOS/NLOS transition

If we look at the transition part between LOS and NLOS phases of the signal, oscillations in the signal strength are obvious as seen in figure 4.6 as well. Important to notice is that these oscillations start before the cut between LOS and NLOS and last on for a few minutes after the cut.

This matches the theory of the Fresnel ellipsoid that spans around the direct ray from the satellite as explained in section 2.3.3. As long as nothing cuts the ellipsoid, no diffraction is expected. But as soon as a building enters the ellipsoid, diffraction effects can be noticed. The more the Fresnel zone is interfered with, the more the signal is inhibited.

This oscillating signal strength coincides with the theoretical diffraction effects shown in figure 2.9 in section 2.3.3. Therefore, PRN09 is a very good example for diffraction caused by a sharp edge building.

4.4 Conclusion

In this chapter, the basic algorithm for LOS/NLOS determination has been validated with the result that the availability of LOS only satellites at the experiment site is very low. Only 4.7 satellites are in LOS in average. Therefore, NLOS observations are important to guarantee a positioning solution as LOS only satellites will not be sufficient at all times.

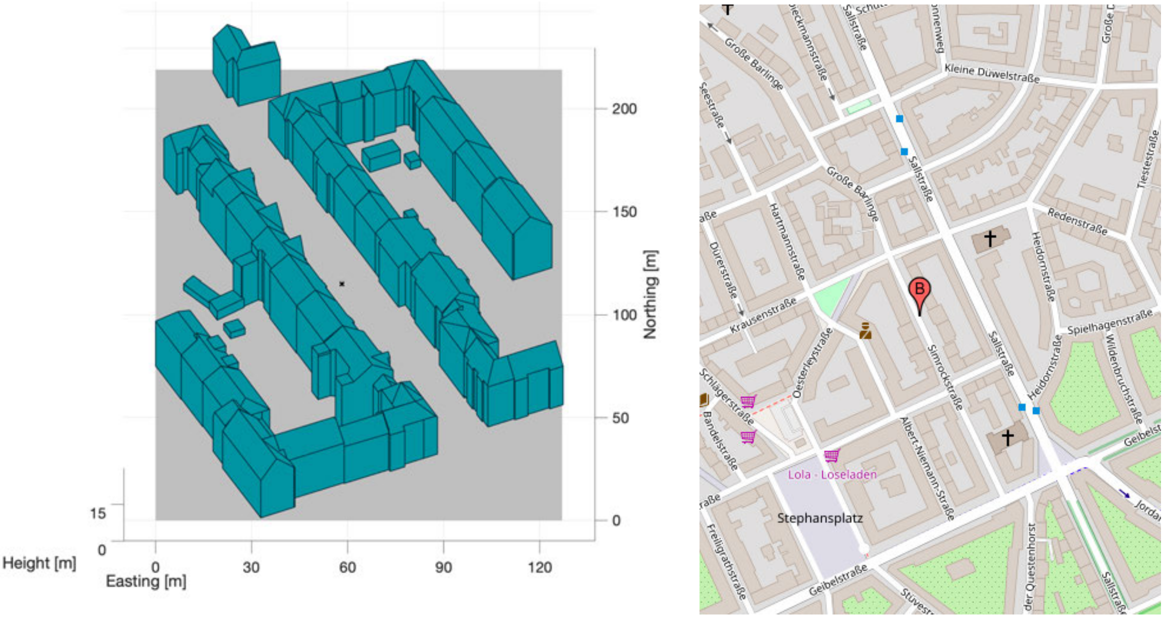
The number of LOS satellites compared with the receiver's recordings gives different results per receiver. The Javad receiver tracks a similar amount of signals as there are LOS satellites while u-blox records significantly more due to its high-sensitivity characteristic. Even NLOS satellites with low $C/N_0$are still recorded. This results in a considerably lower DOP value for u-blox.

Furthermore, the theoretically discussed diffraction effects have been verified as seen in the oscillating signal strengths before and after the blockade of the signal. This effect has to be kept in mind when further investigating LOS and NLOS signals as this distinction alone is not enough to characterize a signal.

5 Analysis of Signal Reflection in Static Mode

For the static analysis of signal reflections, a static point in Hanover Südstadt has been used to simulate a measurement for a whole day. In order to get further findings, the local geometry has been rotated ($\theta = 0°, 90°, 180°, 270°$) so that the effects of the street's orientation can be analyzed. Lastly, the resulting additional path length and the size of Fresnel ellipses are investigated.

5.1 Choice of Simulation Area



(a)

(b)

Figure 5.1: Overview over the local site of the simulation measurement: (a) Test area Simrockstraße for a simulated static analysis in a typical urban surrounding. User position is marked with a cross.
(b) Map of the simulated user position B situated in Simrockstraße in the district of Hanover Südstadt (OpenStreetMap contributors, 2019).

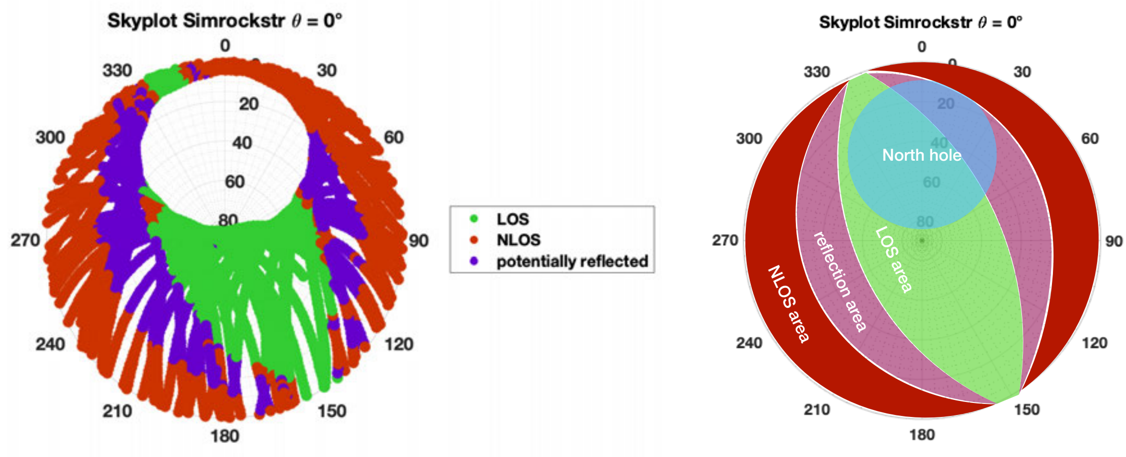
The experiment setup of chapter 4 is not suitable for a thorough reflection analysis as it does not offer many reflection surfaces. Due to the typical condition of an urban canyon, Simrockstraße in the district of Hanover Südstadt has been chosen, see figure 5.1. The simulated user position is surrounded by three to four-story apartment buildings in a street with approximate north/south orientation. Furthermore, the section of Simrockstraße offers roughly 165 m of consecutive urban trench, meaning that there is no street crossing in-between. This location gives the opportunity to analyze a simple but typical urban trench situation.

The time for this simulation has been set to the same DOY 348 in 2018 as in 4 but the extent has been expanded to a period of a whole day. The area dimensions have been set to 110 m in east/west and 200 m in north/south direction as these cover the whole street section.

At the northern end, the houses are approximately 23 m apart from each other while at the southern end, approximately 21 m, making the street a little wider in the north. The building walls have a height of roughly 16 m with rooftops 23 m above ground on the west side and 20 m on the east side.

5.2 Occurrence of Reflected Signals

To analyze the amount of possible reflections, the simulation has been performed using the reflection detection algorithm as explained in section 3.2.



(a)

(b)

Figure 5.2: Skyplots of Simrockstraße with actual reflection points and in a generalized form: (a) Skyplot of Simrockstraße, distinguished between LOS, NLOS as well as NLOS that might be reflected. (b) Generalized skyplot of Simrockstraße.

Figure 5.2a gives an overview over the satellite visibility of the user point as the distribution of simulated reflected signals around the station are visualized in a skyplot. The shape of the street can be seen quite clearly as the open ends of the road are in azimuth angles of 150° and 340° which corresponds to the field where LOS observations can be found. The highest buildings form an elevation angle of 60° from the user point in the west. Then, a transition area can be noticed between the LOS and NLOS area in which single reflections potentially take place. This area extends the LOS field of view up to 20 degrees in elevation. The lowest 20° to 30° of elevation are blocking all the signals apart from the open ends of the street, forming an NLOS area.

These zones can then be generalized as depicted in figure 5.2b. The width of the reflection and NLOS zones depends on the azimuth angle. If the azimuth angle is transverse to the street angle, the zones will be the widest. Because of the so-called „north hole“ there will be no observations in that specific area.

Figure 5.3 now shows the satellite visibility over time, illustrating the number of directly visible LOS signals and additionally taking the number of reflected signals into account. Generally, four satellites are directly visible in average. But taking also the number of potentially reflected signals into account up to twelve observations could be detected. While at some points somehow barely any reflections take place as it is the case shortly after 13:00, the number of additional signals can increase to six at 9:00. This number is quite impressive as it is higher than the average number of visible satellites and doubles the number of observations at the specific time.

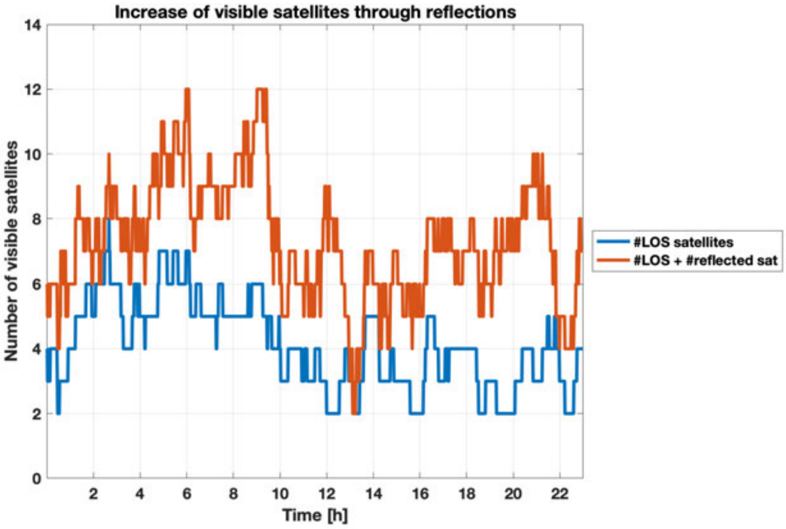


Figure 5.3: Satellite visibility of LOS only and additional reflected signals over time.

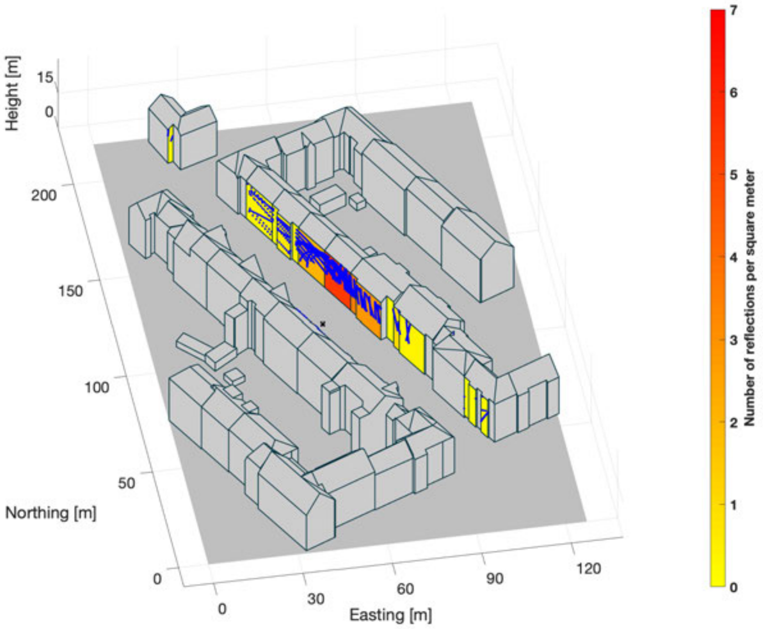


Figure 5.4: Reflection planes color-coded after their frequency of reflection together with the actual reflection points (blue).

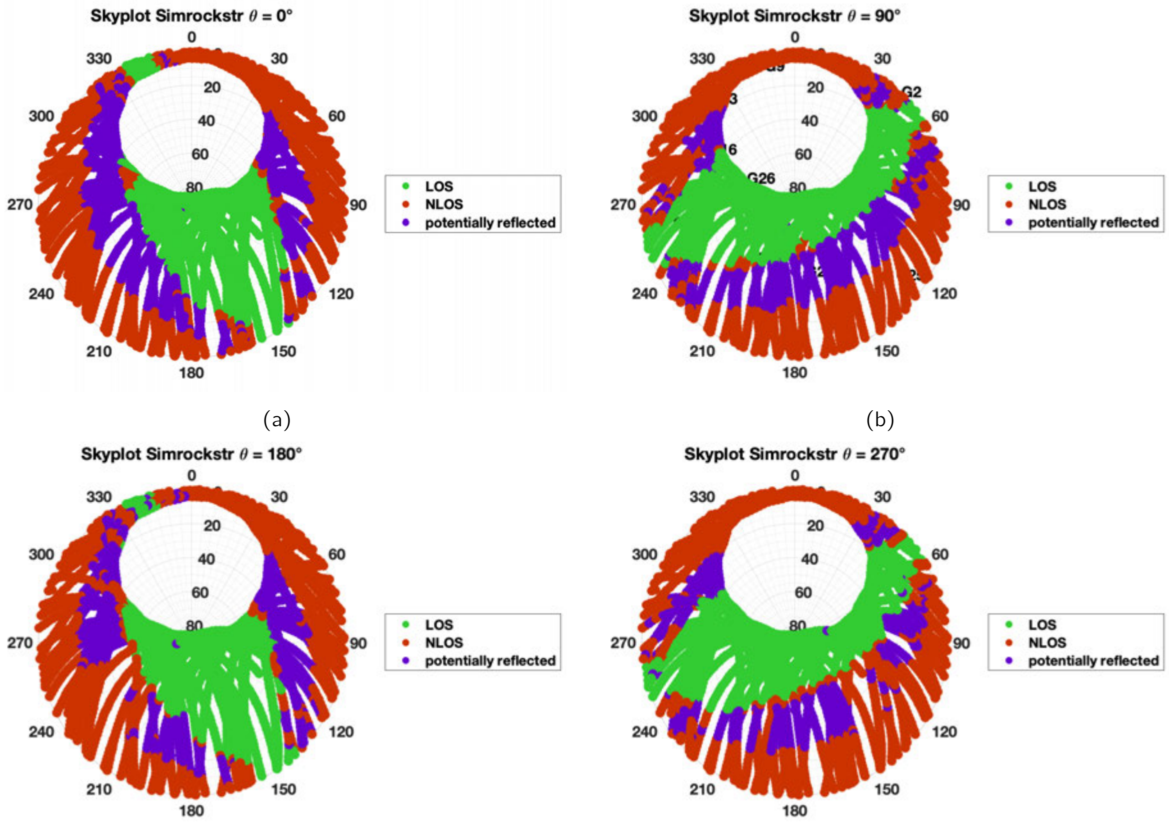
To see which surfaces actually are responsible for the reflections, figure 5.4 illustrates these as color-coded surfaces where a high number of reflections are colored in red and those with less reflections are colored in yellow. The blue dots show the reflection points on the surface.

What can be noticed is that as expected the surfaces in the closest vicinity have the highest density of reflection points. But up to a distance of roughly 50 m, surfaces are being used for reflections. In this example, all of the reflection surfaces are of type wall as the roof planes' angles are too flat. Furthermore, the shape of the reflected signal trace can be seen. Most of them just form a line starting slightly above antenna height and moving up towards the lower roof edge, some also form a curve. Towards the sides of the street, the traces mostly move in left and right directions.

5.3 Effect of geometry rotation

To see the effects of the local geometry orientation, the chosen region has been analyzed with four different orientations. The setup has been rotated around the user's position by an angle of $\theta = 0°$, 90°, 180°, 270° within the local xy-plane. This means that the global satellites still come from the same azimuth and elevations but the local geometry is changed.

The expected results will then show under which circumstances the most reflections take place. Or in contrast, they will show under which conditions the most LOS observations can be tracked as the orientation of the street will play a role as will the position of the "north hole".



(a)

(b)

(c)

(d)

Figure 5.5: Skyplots of the site rotated by four different angles θ.

Figure 5.5 shows the skyplot that is distinguished between LOS, NLOS and reflected signals for each of the rotations. The overall impression is that the before mentioned zones in figure 5.2b are simply rotated but with the effect that the north hole is covering different areas of the skyplot. The actual size of each zone is presumably the same, only some single points change color.

Table 5.1: Number of possible reflections and LOS/NLOS observations for each rotation.

Rotation angle	# possible reflections	#LOS obs	#NLOS obs
0°	4746	5675	11276
90°	3512	7168	9783
180°	4191	5592	11359
270°	3065	7163	9788

In table 5.1, the number of possible reflections together with the number of LOS observations and NLOS observations are stated for each rotation angle. Although the skyplots in figure 5.5 do not show significant differences between the rotations, the figures in table 5.1 do.

Possible reflections denote all mathematically possible single reflections of LOS and NLOS satellites but may include multiple for one satellite. It shows the potential for reflections of the local geometry. The highest potential for reflections therefore is obtained with rotation $\theta = 0°$ in which the street is oriented approximately north-south and the least possible reflections can be found for $\theta = 270°$ in which the street is approximately east-west oriented.

There are two reasons for the difference in those rotations. First, the position of the „north hole“changes as it covers different regions for every rotation. At $\theta = 270°$, more possible reflections are eliminated simply because there are no satellites who could emit a signal for the respective reflection surfaces. Secondly, there are more observations from east and west than from the north and south. Therefore, a street with open ends to east and west naturally does not offer as much reflection surfaces. Instead, a street where signals mostly get in transversally provides more surfaces applicable for reflections.

The number of possible reflections between rotations of 180° differs about 12% which can be explained by the fact that the street of Simrockstraße is not symmetrical.

The number of LOS observations directly anti-correlates with the number of possible reflections. Hence, there are more LOS observations in the east-west oriented street than in north-south. But the number of LOS observations is more stable if the local geometry itself is rotated by 180 degrees than the number of possible reflections which means that the building elevation is similar on both sides of the street.

5.4 Analysis of Detour Lengths

Figure 5.6 shows two exemplary reflections of satellite signals on a building surface. It gets clear that the length of the additional path which the reflected signal is taking (i.e. detour length), is highly dependent on the incoming angle. Satellite G30 has a low elevation and could only be reflected through the open end of the street. The resulting detour length is not as high as the one from satellite G08 which is coming from orthogonal to the street in a higher elevation. The reflection surface therefore lies on the opposite side of the street than the satellite itself which results in the signal having to also cover the distance back to the antenna.

Both azimuth and elevation of the incoming signal affect the reflection angle β between the satellite ray and the building surface which directly correlates with the resulting detour length.

What can be noticed further is that for the same azimuth angle, the farther away the reflection point and its surface, the longer the detour, which is only natural. Therefore, in narrow streets, overall smaller detour lengths can be expected than in wide roads.

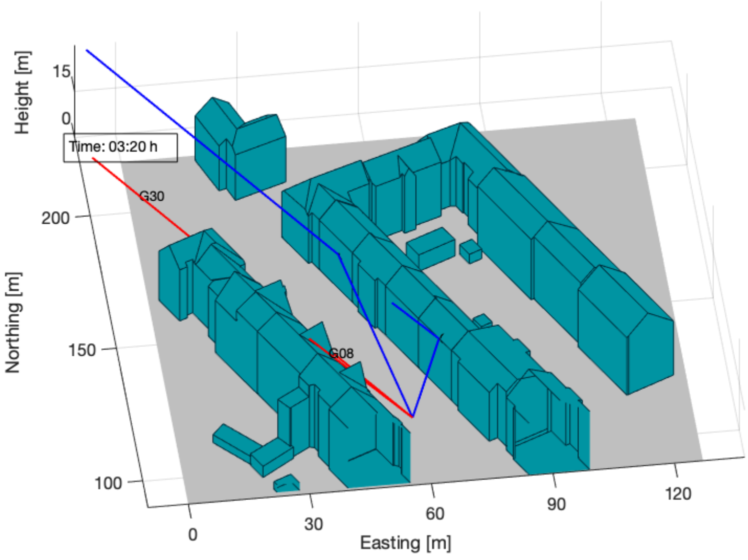
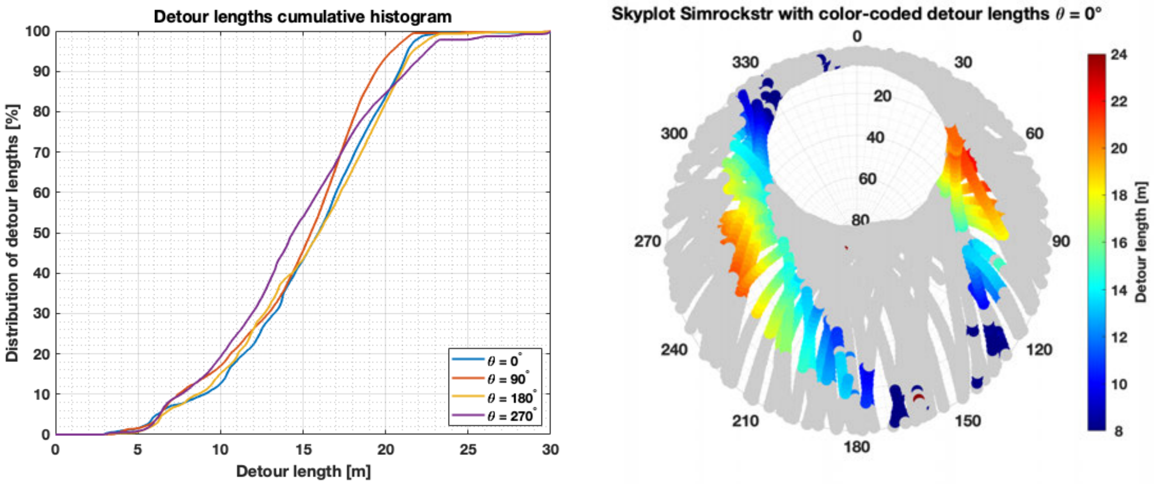


Figure 5.6: Exemplary reflection for G08 and G30 in Simrockstr. The direct signals in red are blocked but the reflected blue rays reach the user point through a reflection on a building surface.



(a)

(b)

Figure 5.7: (a) Detour lengths in a cumulative histogram for four different rotations. (b) Color-coded detour lengths projected onto skyplot. Grey dots show non-reflected signals.

Figure 5.7 shows the absolute detour lengths for the simulation in Simrockstraße in a cumulative histogram and skyplot. The histogram shows that the rotation does not effect the distribution of the detour lengths at all. For all rotations, the detour lengths are except for very few outliers in between 5 and 23 meters, equally spread. This means although the actual number of reflected signals is different especially between rotations of 90 degrees, the distribution of their magnitude is still the same.

In the skyplot of figure 5.7b, the relation between the angle of the incoming signal and the resulting detour length can be seen. Signals coming from the open ends of the street have a smaller detour than the ones coming in transversal to the street. At the same time, for each azimuth angle it is evident that the lower the elevation the larger the detour length.

This reassures the more theoretical findings of Betaille et al. (2013) where the detour lengths have been analyzed using a more simple city model only setting the width of the street together with the building heights. Figure 5.8 shows their skyplot with color-coded detour lengths but not only for single reflections but also for double and triple ones.

Since the CityModel of Simrockstraße uses geometry close to reality, the shape of reflection areas is not as perfect as in the simple model of Betaille et al. (2013). Some oriels or pillars will de facto occlude reflections. But still, figure 5.7b shows a confirmation applied to more realistic conditions.

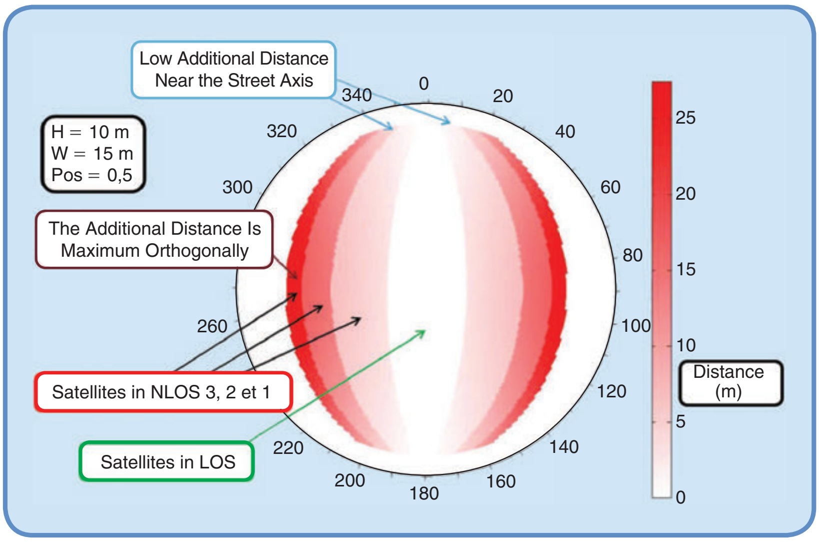


Figure 5.8: Detour lengths depending on azimuth and elevation (Betaille et al., 2013).

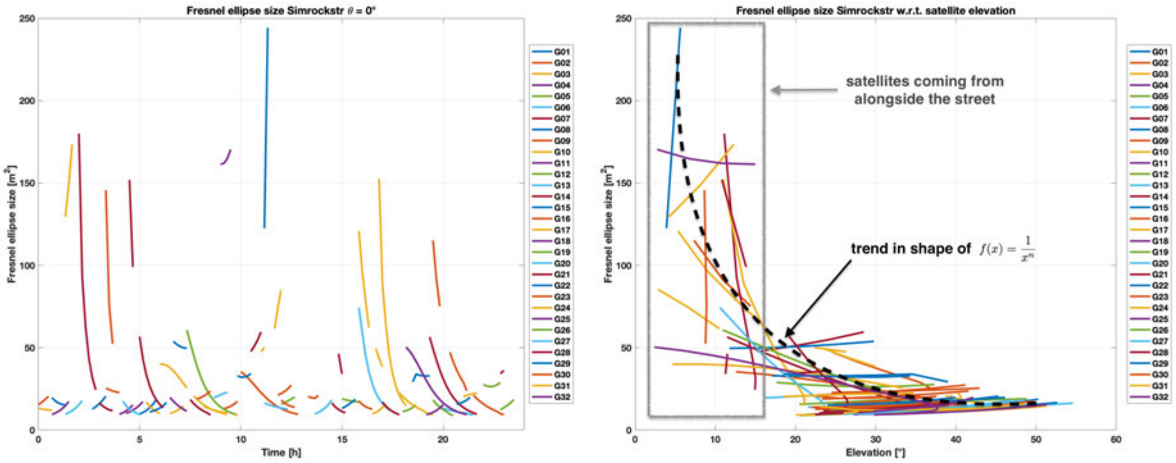
5.5 Analysis of Fresnel Ellipses

As stated in section 2.3.3, the whole area of a Fresnel ellipse has an impact on the signal reflection. Therefore, it is useful to take a look at the Fresnel ellipses of the simulated reflections in this chapter.

Figure 5.9 displays the resulting Fresnel ellipse sizes taken from the calculated semi-major axes a and b as explained in section 2.3.3. The ellipse size A_{ell} is determined by

$$A_{ell} = \pi ab. \tag{5.1}$$

Area sizes of up to almost 250 m^2 are reached in the simulation with a mean value of 29 m^2. The ellipse formed by the same satellite signal ray can change its size rapidly, noted by the almost vertical outlier markers in figure 5.9a. Considering a Fresnel ellipse size of over 150 m^2, the assumption of a point-wise



(a) Fresnel ellipse size over time.

(b) Fresnel ellipse size w.r.t. satellite elevation.

Figure 5.9: Fresnel ellipse sizes for the simulation in Simrockstraße.

reflection is very unfavorable since many different textures or uneven surfaces can occur inside of the Fresnel ellipse area.

Nevertheless, the CityModel can at least give an idea about the reflection distances. As it does not contain surface textures, a further look on the impact of the Fresnel area surfaces is not possible.

In figure 5.9b, the Fresnel ellipse sizes are displayed w.r.t. the satellite elevation. The whole picture forms a graph in a shape of $f(x) = x^{-n}$. Most satellites reflect on the building surfaces facing the front which means they need a certain elevation of approximately 25° to be able to reach the building's front surface. but some satellites also come from an azimuth angle close to that of the street, forming a glancing intersection. This incidence angle results in a large Fresnel ellipse but also in a quickly changing size. Therefore, the markers in a whole for all satellites shape a trend of $f(x) = x^{-n}$.

5.6 Conclusion

In this chapter, the reflection algorithm has been validated in a simulation. Additionally, the coordinates of reflection points, the detour lengths and the size of the resulting Fresnel ellipse have been computed.

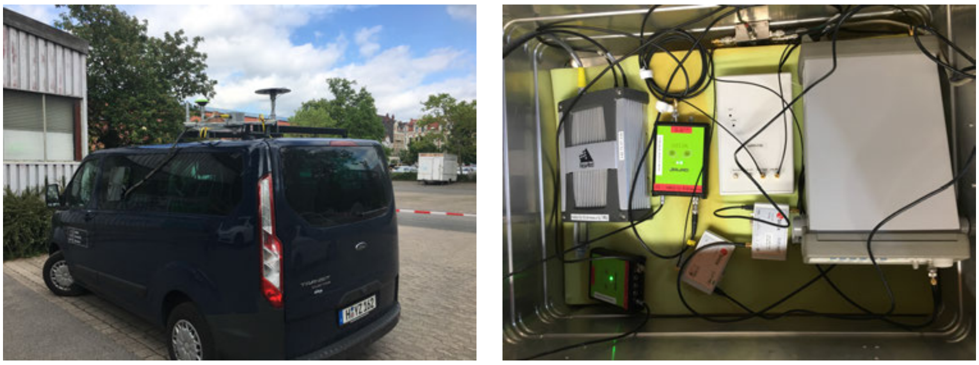
After computing reflections over the course of a day, the resulting reflection areas could be identified. Depending of the location of the "north hole" compared to the street's orientation, the amount of reflected observations differs. The resulting additional path length through reflection depends on the azimuth and elevation of the respective satellite. The highest detour length is achieved when the satellite's azimuth is perpendicular to the street's orientation and the elevation is as low as possible.

The size of the Fresnel ellipse is also dependent on the satellite signals's incidence angle. The area is largest if the azimuth angle is almost parallel to the street's orientation and the elevation is as low as possible. In those cases, the question rises if a single-point reflection is an adequate assumption for a signal reflection characterization. As everything inside the Fresnel ellipse, i.e. different surface structures, textures, roughnesses, contributes to the reflection process, a large Fresnel ellipse will make the actual signal reflection unpredictable.

6 Kinematic Experiment

In May 2019, additionally to the static analyses, a kinematic experiment has been planned and executed. The objective of the kinematic experiment is to test the LOS/NLOS algorithm as well as the reflection correction under real conditions in the urban area of Hanover. It is especially of interest if NLOS observations can be corrected at all and if so of what magnitude the effects show within the positioning.

This is then tested as well in the context of different antenna/receiver set-ups. The main focus here is to compare a low cost, high sensitive installation with a geodetic costlier one.



(a) Vehicle for test drive

(b) Equipment set-up

Figure 6.1: Kinematic experiment vehicle and equipment.

6.1 Design

The planning of the experiment comprises the choice of the test area, the set-up of measurements gear and the computation and selection of the reference trajectory.

6.1.1 Choice of Test Area

To test the Urban Trench Model, a test area is selected that is matching the CityModel as best as possible. This means, since no trees are modeled within LoD2, an area should be chosen that does not have many trees in the public streets or any other possible obstruction that is not marked in the model. Furthermore, a district with deep trenches and close buildings should be the object for the testing so that the expected effects are higher.

All these requirements are fulfilled by streets within the Südstadt district in Hannover. Along streets close to Sallstraße, the buildings have a block-like character and create deep trenches, especially narrow streets like Große Barlinge. Besides, trees are rare so that the CityModel fits sufficiently. The speed of the vehicle will vary between complete stand stills at road crossings and around 50 km/h on the bigger street of Sallstraße.

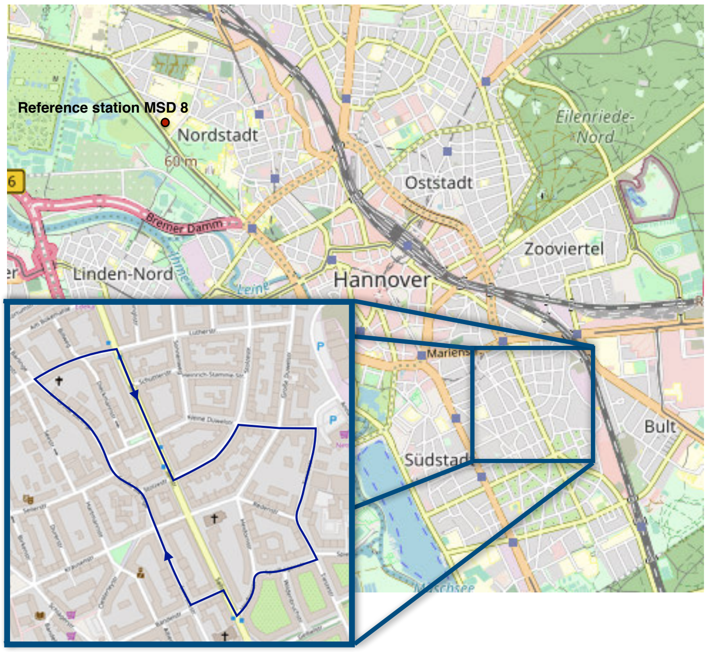
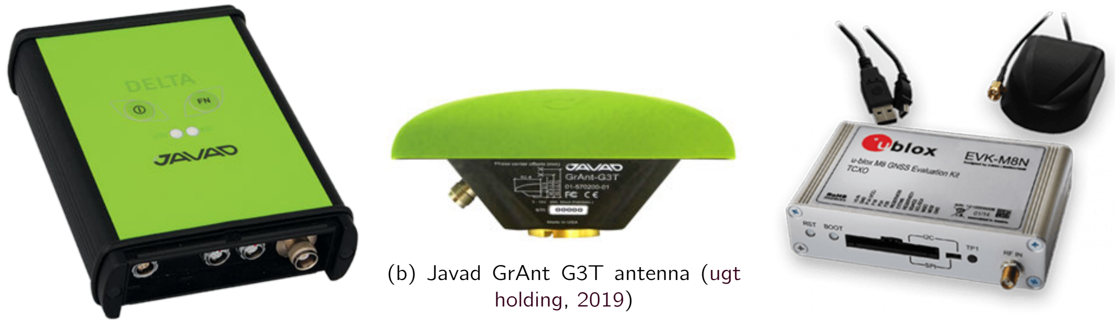


Figure 6.2: Test area Südstadt with test route along streets close to Sallstraße (OpenStreetMap contributors, 2019). Reference station MSD8 is marked with a red dot.

The chosen district together with the driven trajectory can be found in figure 6.2. Arrows show the direction in which the route was taken.

6.1.2 Set-up



(a) Javad TRE_G3T Delta (Javad GNSS Inc., 2019)

(b) Javad GrAnt G3T antenna (ugt holding, 2019)

(c) u-blox NEO 8MT with antenna ANN-MS (u-blox AG, 2019)

Figure 6.3: Equipment from kinematic experiment: two types of receivers and antennas.

The experiment focuses on two different antenna/receiver constellations: one low cost, high sensitivity receiver together with an antenna of the same brand and a geodetic high end receiver with a geodetic antenna. The geodetic equipment is chosen from the brand Javad (figures 6.3a and 6.3b) while the high sensitivity gear is provided by u-blox (figure 6.3c).

However, since the experiment was conducted together with a receiver clock modeling test, more equipment has been put together (see figure 6.4).

1. Six receivers:

- Novatel SPAN
- Javad TRE_G3T Delta, ID 0081
- Javad TRE_G3T Delta, ID 0082
- ublox NEO 8MT, ID 0867
- ublox NEO 8MT, ID 1771
- ublox NEO 8MT, ID 1779

2. Two antennas:
   - Javad GrAnt G3T
   - ANN-MS

For the receiver clock modeling experiment, an atomic clock *Micro Semi Miniature Atomic Clock SA.35* with short term stability has been added and connected to Javad receiver 0082 as well as to ublox receiver 1771 with a signal wave generator (SWG) in-between. This was done to compare a low-cost and a geodetic receiver with and without clock modeling but is not further investigated in this work.

All receivers except for ublox 1779 have been connected to the geodetic Javad GrAnt antenna via splitter. The cable connectors for Javad receivers and antenna as well as for the splitter are of type TNC *(Threaded Neill-Concelman)* with *F* standing for female and *M* for male connectors. Ublox receivers and antenna are connected through SMA *(SubMiniature Version A)* cables, hence requiring TNC/SMA adapters.

The Novatel SPAN receiver was additionally connected to an IMU (iMAR FSAS) for the computation of the reference trajectory. The connection is important so that the IMU data is available in GPS time.

As reference station, point MSD8 on the measuring rooftop of IfE was taken where a LEIAR25.R3 antenna was installed together with the same Javad receiver model TRE_G3T Delta, ID 0993, as in the vehicle. The measurement there was conducted independently from those obtained from the vehicle.

6.2 Execution

The experiment was held on May, 22nd 2019. After setting up the equipment and logging start of the reference station MSD8, the actual measurement with the vehicle commenced on a parking lot next to Schneiderberg in Hanover Nordstadt at 11:45 GPS time. All of the receivers were started and data was recorded throughout the drive from Schneiderberg to the test area in Südstadt and further around the city which was meant for analyses of the receiver clock modeling.

The test route (cf. figure 6.2) was run six times with the exact same path taken five times. The first round turned out to be a little different than the other rounds and was therefore neglected in the analysis.

Unfortunately, ublox receiver 1779 stopped recording before the drive in the test area due to technical problems. In the further process, there is no data available from this receiver from the test area. All the other receivers recorded data with a logging interval of 1 second throughout the measurement.

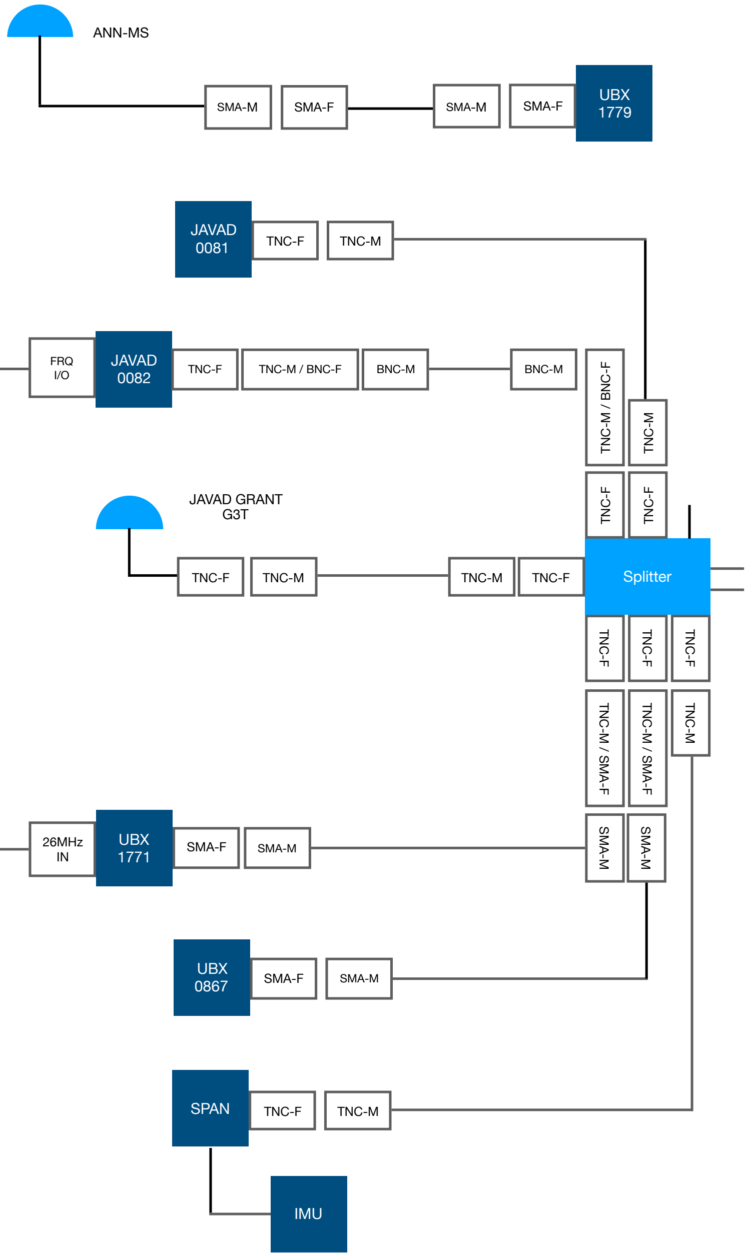


Figure 6.4: Connection diagram for the experiment. Blue objects are part of the measurement equipment, blank rectangles represent connectors.

6.3 Reference Trajectory

In order to achieve a precise reference trajectory, the information of an additional station has to be considered as well. As reference station, point MSD8 on the measuring rooftop of IfE was used (see figure 6.2). The solution for this station was obtained with a ionosphere-free linear combination.

This setup allows the calculation of a reference trajectory for the entire test drive from multi GNSS observations combined with IMU data in a tightly-coupled relative positioning. The lever arm lengths between antenna and IMU are calculated separately and introduced to the computation parameters. The recorded data together with the known coordinates of the reference station and precise satellite orbit and clock information (ephemerides: Code-MGEX) are imported into the post-processing software TerraPOS 2.4.3 (Terratec AS, 2019), which is then used to calculate the reference trajectory. The frequencies used are

- GPS (G): L1, L2, L5
- GLONASS (R): G1, G2
- Galileo (E): E1, E5a, E5b

Table 6.1: Key figures for comparison between trajectories.

#	Rec	GNSS	IMU	Clk	El Cutoff	N fixed	#PhaseObs	$\sigma_h[m]$	σ_v [m]
1	SPAN	G, R	1	0	5	88.2	32423	0.041	0.049
2	Javad 82	G, R	0	1	15	58.0	27855	0.415	0.504
3	Javad 81	G, R, E	1	0	15	72.9	42597	0.038	0.047
4	Javad 82	G, R, E	1	1	5	72.7	43007	0.038	0.046
5	Javad 82	G, R, E	1	1	15	72.9	42586	0.038	0.047

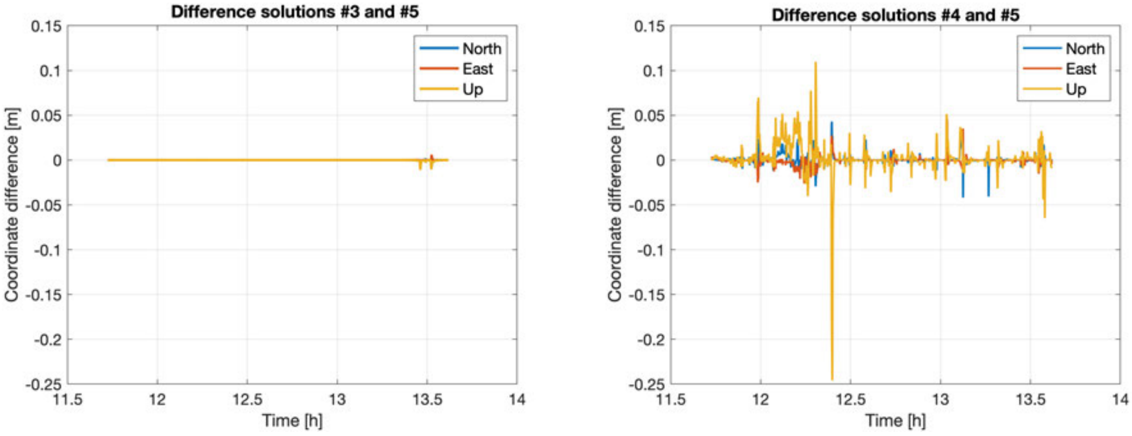
Different combinations and calculations have been tested to find the best solution for the reference trajectory. Ublox receivers have not been considered for the reference trajectory in the first place. In the following, both Javad receivers have been tested with different parameters of GNSS systems, IMU setting on or off and elevation cutoff angle. The outputs for percentage of fixed ambiguity N, number of phase observations and horizontal and vertical standard deviations have then been analyzed. These figures can be found in table 6.1.

For the SPAN receiver (#1 in table 6.1), there are no Galileo observations available. This has a direct effect on the number of phase observations which is about 25% lower than for solutions with Galileo observations. This effect can also be seen for #2 Javad receiver 82 without Galileo. For this combination, also the percentage of solved ambiguities is the lowest and it shows that the usage of an IMU is reasonable to get an accurate solution.

For combinations #3, #4 and #5, the figures for fixed ambiguities, number of phase observations and the standard deviations are not significantly different which is why other factors have to be considered. First, version #3 is excluded as it is assumed that the clock information will provide a better solution. This is because the external atomic clock is more stable than the internal oscillator of the receiver and hence influences the receiver's tracking loops (Weinbach and Schön, 2011). However, the actual coordinate differences between the same settings except for the external clock embedding is not really noticeable as seen in figure 6.5a. The maximum absolute difference only reaches 5 mm in the up-component which is also one of only two peaks over the whole measurement process.

Secondly, combination #4 is excluded for the reason of high elevated buildings throughout the measurement which make it nearly impossible for the receiver to gain information from signals coming from below 15 degrees. Setting the cutoff angle to 5 degrees only will clear the way for reflected or diffracted signals that are disadvantageous for computing the reference trajectory. Still, the effect in the resulting coordinates is not negligible as seen in figure 6.5b. Over the course of the measurement, many peaks in the coordinate difference of 5 cm in the up-component are noticeable with outliers going up to 25 cm. This makes it very likely that the incoming signals below 15 degrees of elevations are highly affected by multipath causing the resulting position to differ.

To conclude, combination #5 has been chosen to represent the reference trajectory. It uses three different GNSS, takes additional supporting information from IMU and atomic clock and rejects observations below the cutoff angle of 15 degrees.



(a)

(b)

Figure 6.5: Comparison between solutions #3, #4 and the selected solution #5 (cf. table 6.1).

6.4 Measurement Data Preparation

In order to compare the measured data with the outcomes of the analyses from the next chapter 7, the logged files are processed with the IfE Matlab Toolbox version 6.1 and provided in the corresponding workspace.

The structure of the Matlab workspace is a division between observations, station parameters, satellite parameters and computed values. To compare observations and computations, the observed minus computed (OMC) outputs are analyzed.

Within the computations, all corrections and reductions that have to be applied to the observations are listed. These include

- **Satellite clock error and orbits**: taken from Center for Orbit Determination in Europe (CODE), multi-GNSS product for satellite clock, orbit and earth orientation parameters
- **Tropospheric Correction**: application of the *Global Pressure and Temperature Model version 2* by Boehm et al. (2007)
- **Ionospheric Correction**: taken from IONEX files provided by the IGS containing vertical total electron content and daily GNSS satellite differential code bias (DCB) values (International GNSS Service (IGS), 2019)
- **Satellite Phase Center Offset (PCO)**: taken from antenna calibration sheet provided by the National Geodetic Survey (NGS) (National Geodetic Survey, 2019)
- **Relativisitc effects**: Modeled according to Sagnac effect

Satellite Phase Center Variations (PCV) have not been applied as first kinematic attitudes would have been needed. However, the magnitude of a few millimeters is negligible small and will disappear in the measurement noise.

7 Kinematic Experiment Analysis

In this chapter, the kinematic experiment data is analyzed concerning LOS/NLOS observations, reflections and is investigated in the position domain as well. Therefore, the algorithms from chapter 3 concerning LOS/NLOS determination and reflections are applied.

The measurement data from the experiment is divided into its five rounds, each of which is analyzed separately and later compared to the others. The start and end point of each round is the crossing of Lutherstraße and Sallstraße.

7.1 Algorithm application

In order to separate LOS and NLOS observations and to search for reflection points, the respective algorithms from chapter 3 are applied in trajectory mode.

The area size is set dynamically to the maximum extent in both easting and northing of the reference trajectory plus additional 150 m so that the LOS/NLOS determination can be executed adequately. The reference trajectory is then used as position in order to determine satellite visibilities. The time is set to the corresponding receiver time of each epoch (1 Hz sampling rate) to have an accurate reference.

7.2 Satellite Visibility Analysis

Figure 7.1a shows the available satellites exemplarily for round 1 in a skyplot while figure 7.1b shows the satellites which Javad 82 receiver recorded.

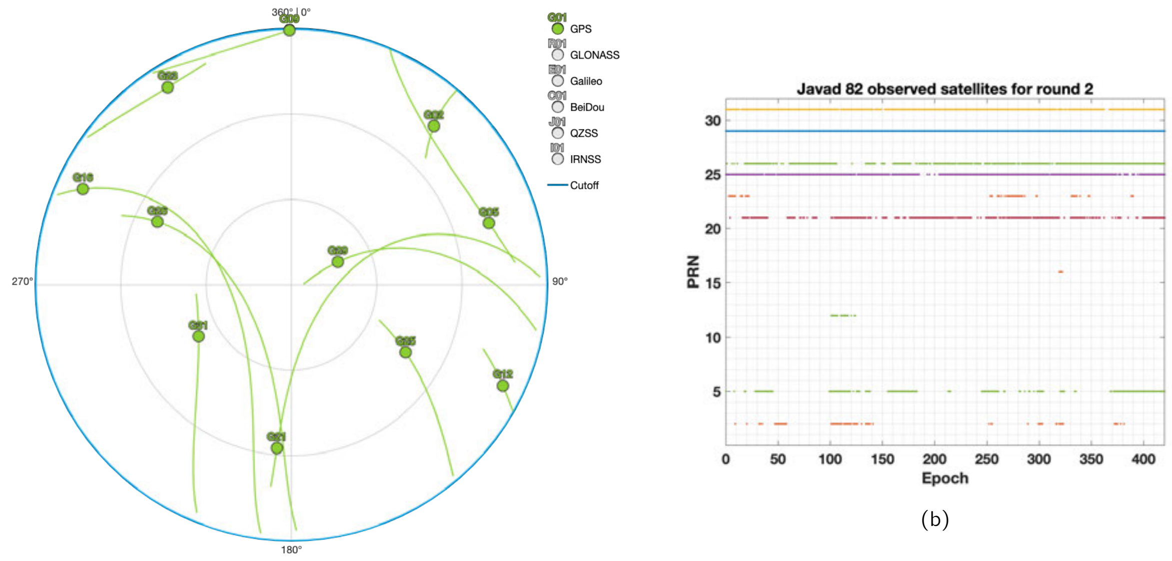
Satellite G29 is the highest elevated one and the only constantly recorded out of all eleven available satellites. Satellites with elevations between 30 and 60 degrees could be tracked but still contain data gaps and cycle slips, e.g. G26. Lower elevations cause the signals to be received only partially, some are visible only for a few seconds, e.g. G16.

The amount of recorded satellite signals depends on the receiver as already mentioned in section 4. Figure 7.2 shows the number of recorded signals for receivers Javad 82 (7.2a) and u-blox 1771 7.2b. As the green bars display the amount of received LOS signals and the purple ones show additional reflected NLOS signals, the red bars surpassing this limit seem to come from unknown sources.

First, it can be noticed that the number of LOS satellites varies highly throughout one measurement drive. With a maximum of 8 and a minimum of 2 satellites, the visibility differences are very high, moreover, it changes within seconds as the vehicle passes buildings and crossings. After the maximum of 8 satellites at epoch 110, the minimum of 2 is reached only a few seconds after.

The number of reflected satellites also varies between zero and four signals at one epoch. This is due to the fast changing local geometry where surfaces sometimes are more likely to reflect and sometimes less.

Comparing the two receivers, it is to be noted that Javad barely receives more signals than there are present through direct LOS or reflection. Only in the last epochs where the vehicle was standing still at a



(a)

(b)

Figure 7.1: Available and observed satellites in round 1 of the experiment route. (a) Skyplot of all available GPS satellites at the start of round 1, forecasting their track for the next hours (Trimble, 2018). (b) Observed satellites in round 1 for Javad 82 receiver, PRN-wise representation. Epochs are 1 second apart.

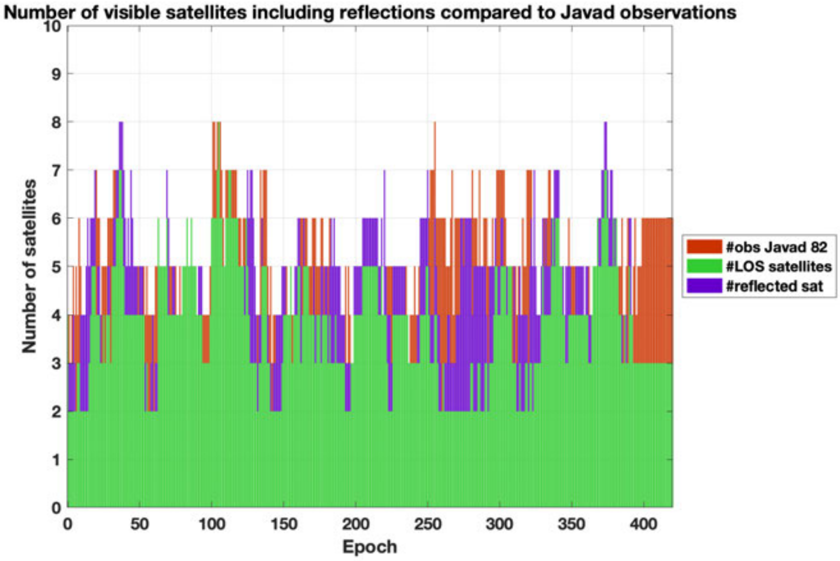
crossing, we see some red bars. This is completely different for the u-blox receiver. It consistently records two to four more satellite signals than it could be reconstructed which means it also tracks signals that must be highly diffracted or reflected multiple times. Whether these signals give a benefit in positioning and hence whether this high sensitivity characteristic is advantageous in this situation or not has to be investigated.

The difference in the number of satellites again affects the DOP values, as already mentioned in the static analysis in chapter 4. Figure 7.3 compares HDOP (Fig. 7.3a) and VDOP (Fig. 7.3b) for the two receivers Javad and u-blox as well as the theoretical LOS only DOP. What is different from the DOP values in the static part is the higher variance which is of course because of the fast changing geometry. But also, the difference in magnitude between Javad and u-blox receiver is not as big. Both receivers mostly lie in between HDOP 1 and 2 as well as VDOP 0.5 and 1.5. However, the DOP level of u-blox is still constantly lower and does not provide outliers like Javad. Even though the surroundings are altering throughout the measurement, the u-blox VDOP has a standard deviation of only 0.01 with a mean value of 0.58. Javad's mean VDOP and standard deviation with 1.26 and 0.80 are a lot higher.

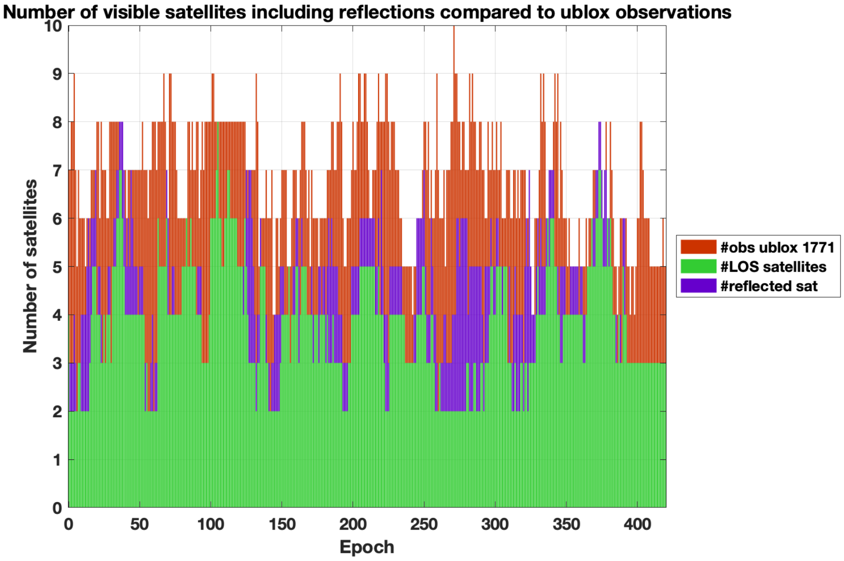
The LOS only DOP values show a higher level and also variation, keeping the DOP values in unstable higher regions. Just like in the static case, it can be stated that the LOS only observations are not enough for a confident positioning.

Figure 7.4 takes into account the local geometry when analyzing the number of LOS satellites, theoretically. It does not involve actual measurements from receivers and hence only gives information about the impact of the surrounding buildings on the satellite visibility. It seems that long, narrow streets reduce the number of visible satellites tremendously, e.g. the road section in the south west area. Other similar sections such in the north west area however sometimes even show six visible satellites.

What can be recognized quite clearly is that most of the road crossings provide a better visibility, especially in contrast to the neighboring road section. Taking exemplarily the crossing at (540 m, 630 m), the number of LOS satellites jumps from two to six in a very short pathway. This happens because after the last

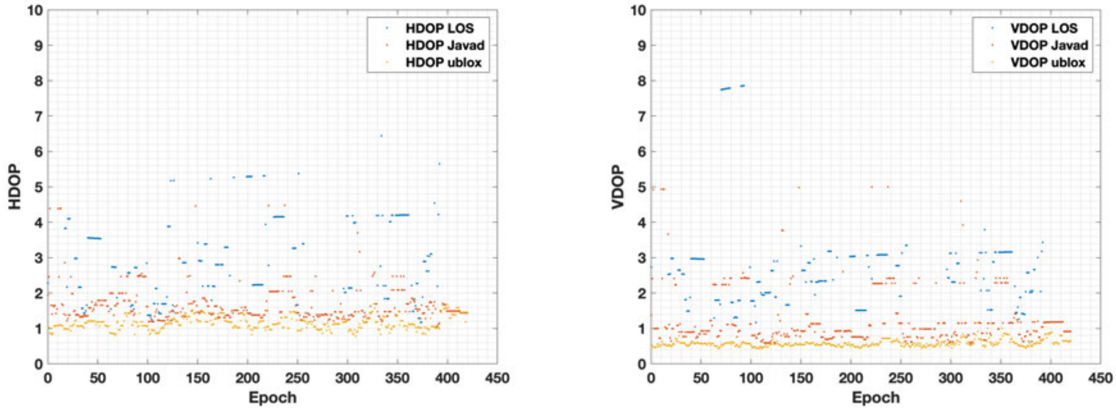


(a) Satellite visibility compared to Javad 0082 observations.



(b) Satellite visibility compared to u-blox 1771 observations.

Figure 7.2: Satellite visibility comparison between number of directly visible satellites (LOS), additional reflected signals and actually observed number of signals in round 1 of the experiment drive.



(a) HDOP

(b) VDOP

Figure 7.3: DOP comparison between LOS only, Javad and u-blox observations for round 1 of the experiment route.

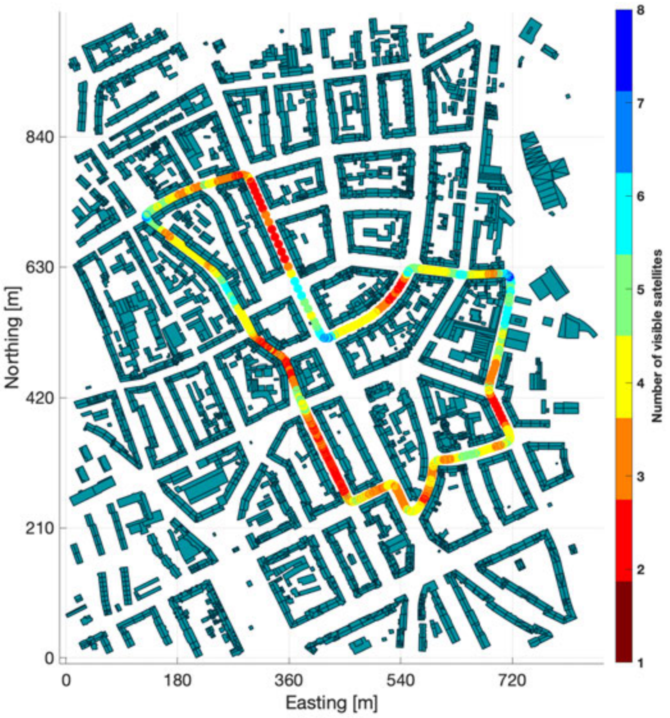


Figure 7.4: Number of LOS satellites in round 1 of the experiment drive.

building's edge all the lower elevated satellites from west and east direction, which is perpendicular to the road orientation, are in sight again.

A further analysis on classified road sections follows in section 7.6.

7.3 Reflection Analysis

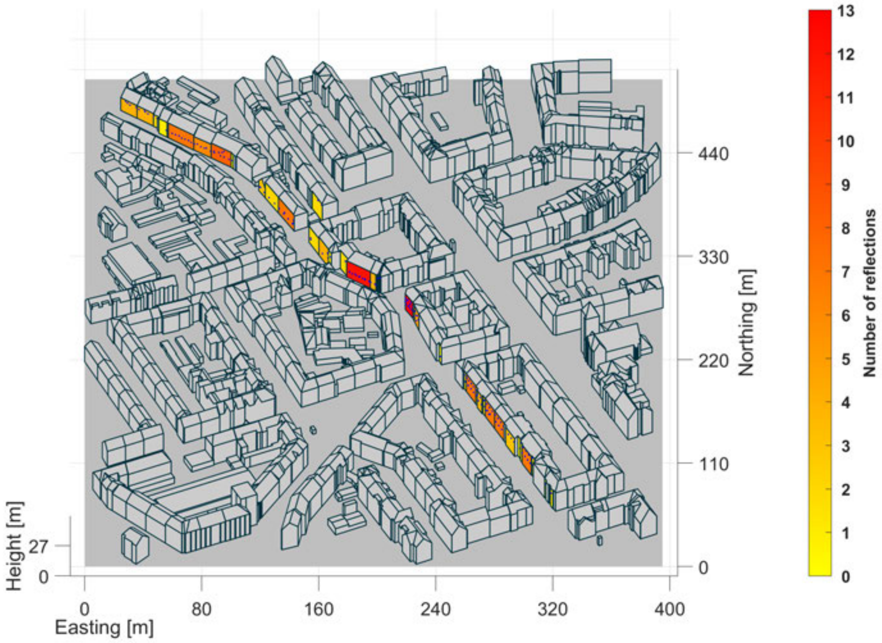


Figure 7.5: Number of reflections per polygon for round 1 in a close-up.

To evaluate the algorithm performance in a kinematic state, figure 7.5 shows a close-up of Simrockstraße and Große Barlinge with reflection points on color-coded building surfaces. For each round, around 400 possible reflection points are determined. Some of those are displayed in the figure.

It can be seen that the vertical building front surfaces in this case are the only ones where reflections could take place as there are no reflections on building rooftops. Furthermore, because of the kinematic mode, the reflection points form a horizontal line over the course of several neighboring buildings. This is because the satellite geometry situation does not change significantly between observations and neither does the geometry of the buildings change significantly. Therefore, a similar constellation offers the same reflection geometry.

The amount of reflection points on one surface depends on the speed of the passing vehicle, the orientation of the surface but also on the size of the polygon. Therefore, there are surfaces next to a crossing that contain more reflection points than those where the vehicle only passed by.

A more unexpected reflection occurred on a building that is not located in the same street where the vehicle drove at (170 m, 380 m). The coincidence that this reflection building is higher than the one between the vehicle and that building made this possible. Therefore it is important to not only take the buildings from one street into account.

7.4 Code Observed Minus Computed (OMC)

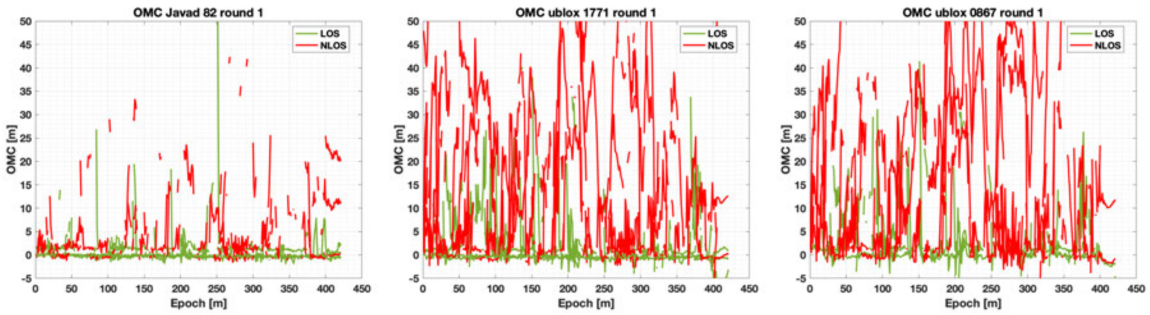
The Observed Minus Computed (OMC) variable describes the difference between the observed range to the satellite and the computed distance including all corrections and reductions mentioned in section 6.4. Therefore, the only remaining parts are the receiver clock error and errors that cannot be modeled, such as multipath or diffraction. To isolate those parts, the receiver clock error has to be eliminated first.

For receivers Javad 82 and u-blox 1771, due to the attached atomic clock, a linear model can be applied. This is done by estimating the drift with a linear fit over all observations in an elevation higher than 50 degrees, ensuring that those data is not impaired. The drift is then subtracted from all OMC, converting it to constant mean.

As receiver u-blox 0867 does not have a clock attached, the receiver clock error cannot be modeled. Instead, the satellite with highest elevation PRN29 is assumed as best fitting the clock error and hence subtracted from the other observations.

The OMC results with subtracted receiver clock error for each receiver are shown in figure 7.6. The graphs are color-coded for LOS observations (green) and NLOS observations (red). It can be clearly seen that most of the outliers are NLOS signals and those lines which form a zero mean are LOS only. Some observations have a constant offset of a few meters. The characteristic that most of the outliers are NLOS is the same for every receiver but it has to be noted that both u-blox receivers show a lot more outliers than the Javad receiver. This coincides with the findings of section 4.3.1 and 7.2 in which it is stated that the u-blox receivers track more satellites of NLOS character.

Moreover, the OMC from u-blox 1771 and 0867 show similar characteristics. Although the magnitude is not always the exact same which can be due to of the different receiver clock elimination methods. However, they show the same patterns, e.g. low OMC scattering between epochs 260 and 300 or lower outliers between epochs 350 and the end. Therefore, we focus on receivers Javad 82 and u-blox 1771 to analyze as both are connected to an atomic clock.



(a) Javad 82

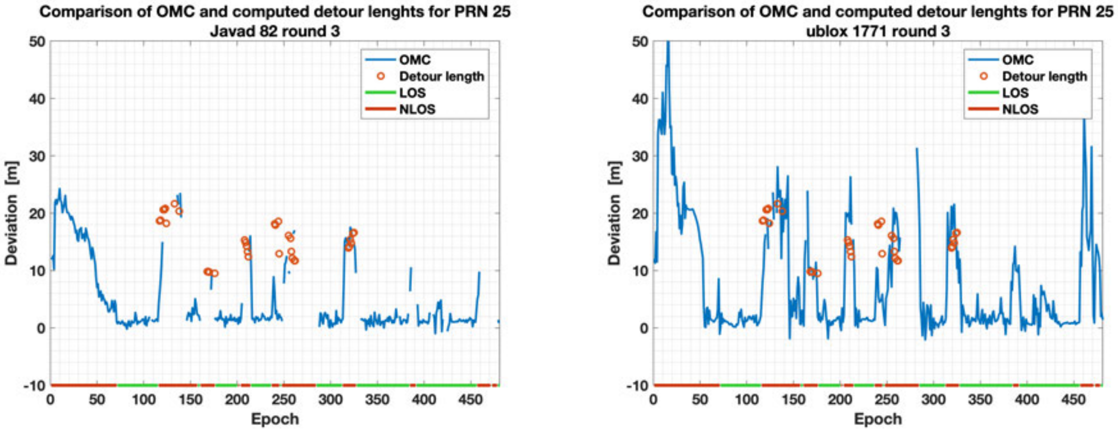
(b) u-blox 1771

(c) u-blox 0867

Figure 7.6: Observed Minus Computed (OMC) for three receivers for all satellites in round 1 over time with color-coded LOS and NLOS.

In figure 7.7, the OMC and computed detour lengths are displayed for satellite G25 only. And if we take a look at the OMC for individual satellites and compare it with the determined detour lengths in figure 7.7, it can be seen that for a satellite that is occasionally not visible the detour lengths match the OMC in a margin of around 5 m quite well. In some sections though, a detour was not detected and hence not applied. Epochs 0 to 50 and 460 to the end represent the crossing at the start in which the vehicle was not moving. Still, the OMC is varying up to 50 m from the true range. At this point, reflections might have taken place at other objects than available in the CityModel or are affected by multipath which was not modeled.

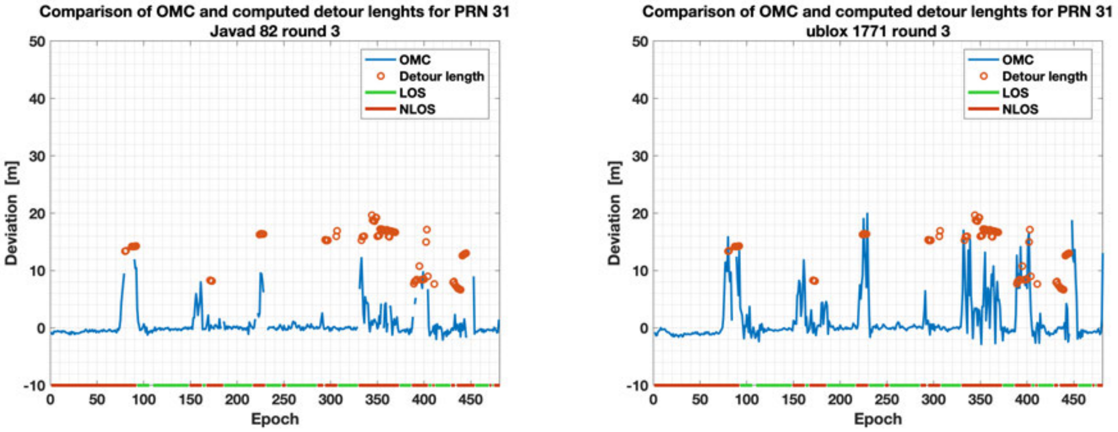
By comparing receivers Javad (Fig. 7.7a) and u-blox (Fig. 7.7b), we can see that the NLOS observations of u-blox have a higher variance in the OMCs. This is also due to the fact that there are simply more NLOS observations for u-blox. But still, the outliers of u-blox scatter more, e.g. between epochs 130 and 140 up to around 15 m. For the Javad receiver, the detour lengths fit slightly better as u-blox OMC mostly are higher than Javad OMC. Also, the error in the beginning of the round is smaller for the Javad receiver, making it more likely that multipath took place as the Javad receiver has built-in multipath reduction tools (Javad GNSS Inc., 2019).



(a) Javad 82

(b) u-blox 1771

Figure 7.7: OMC and computed detour lengths for satellite G25 in round 3. The bar at the bottom shows LOS and NLOS sections.



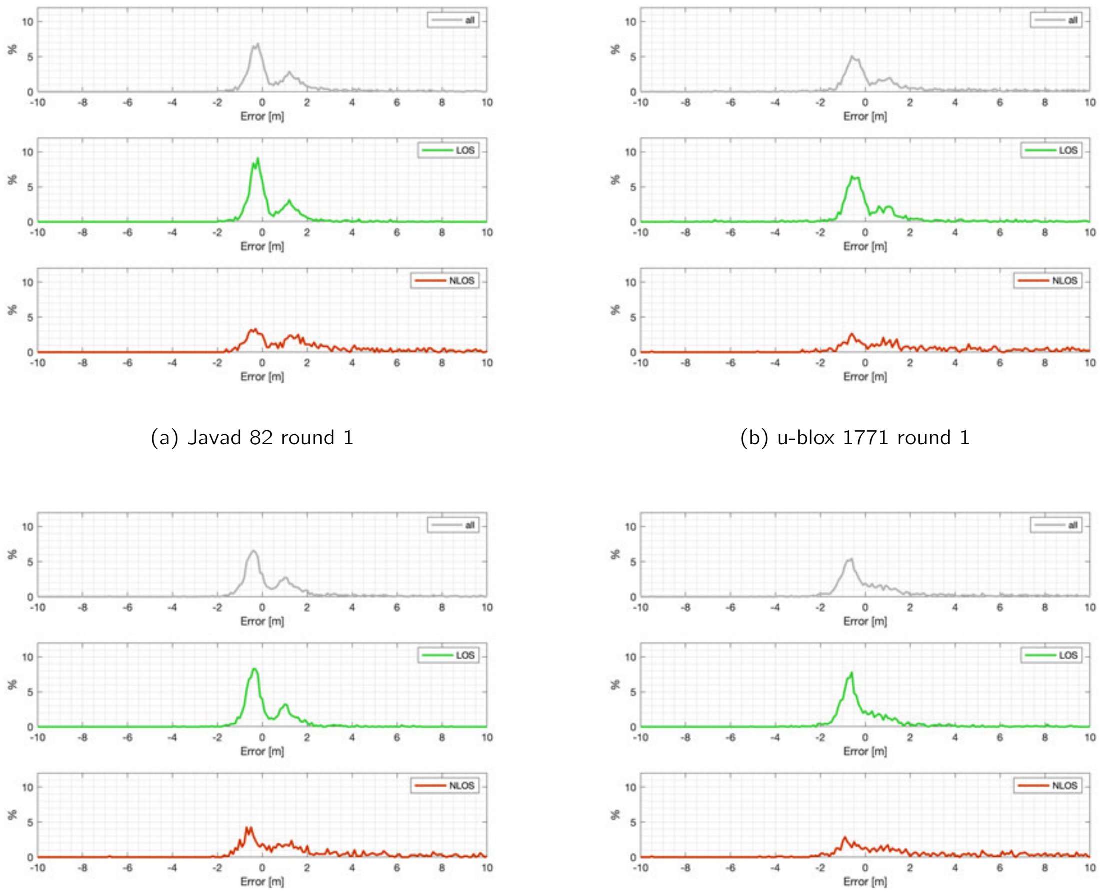
(a) Javad 82

(b) u-blox 1771

Figure 7.8: OMC and computed detour lengths for satellite G31 in round 3.

Figure 7.8 shows the OMC and detour lengths for satellite G31. G31 has a higher elevation than G25 (48° compared to 36° in average) and appears in the west instead of the east. For this satellite, the determined detour lengths fit better for u-blox (Fig. 7.15b) than for Javad (Fig. 7.15a). The reason is that the OMC for Javad throughout the measurement are lower, not only lower than the u-blox OMC but also lower than the detour lengths. This will result in a correction that is too high.

Facing the u-blox receiver, although the noise is still high at outliers, it is obvious that the magnitude fits better to the detour lengths. Especially during epochs 390 to 400, the detour lengths hit the OMC values on point. But just like for the Javad receiver, many detours are too high, resulting in an overcorrection.



(a) Javad 82 round 1

(b) u-blox 1771 round 1

(c) Javad 82 round 2

(d) u-blox 1771 round 2

Figure 7.9: Distributed OMC for Javad and u-blox receiver in round 1 and 3 distinguished between LOS and NLOS observations.

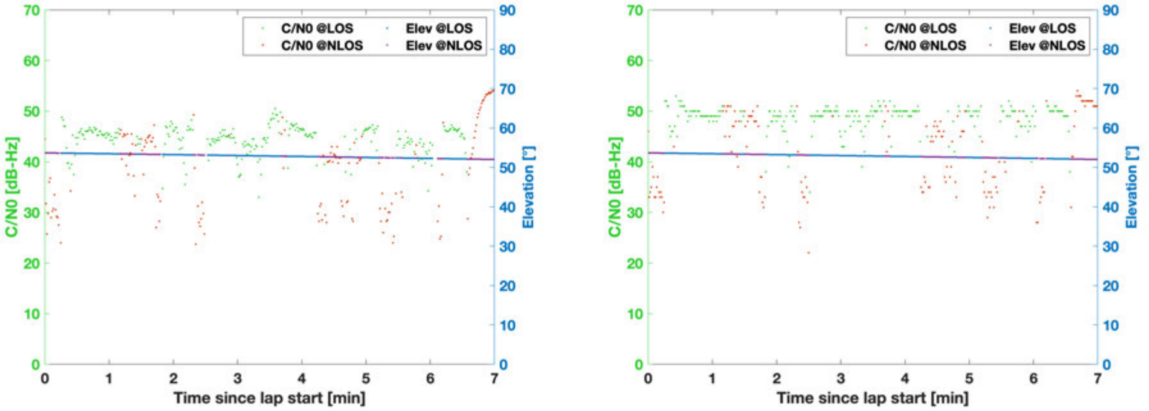
In order to analyze the OMC for LOS and NLOS observations separately, figure 7.9 gives an overview of these differentiations. It displays the distributed OMC for all satellites in round 1 and 2 but distinguishes between observation types LOS and NLOS.

As expected, the OMC for LOS observations have the highest percentage of errors under 1 m compared to the other observation types. The OMC curves for all observations and LOS are quite similar, this is due to the fact that most of the observations are of LOS characteristic as it will be later discussed in section 7.6.

What is obvious and also stated by Betaille et al. (2013) is that the error distribution of NLOS is the most arbitrary. Although peaks do occur in the range of up to 2 m of error, there are still a lot more observations in higher error ranges. This effect is repeated, as shown exemplarily by picturing the first two rounds.

The two receivers both show the same effect but with a different magnitude. As seen before in figures 7.6, there are more high outlier OMC for u-blox 1771 which is also because of the high-sensitivity characteristic of the receiver. This then results in a lower percentage for OMC with an error of less than 2 m as seen in figures 7.9b and 7.9d.

7.5 Signal-to-Noise Ratio Comparison



(a) Javad 82

(b) u-blox 1771

Figure 7.10: C/N$_0$ and elevation for satellite G31 in round 1 for both receivers.

To analyze the kinematic behavior of the signal strength, figure 7.10 exemplarily displays the C/N$_0$ values for satellite G31 for both Javad 82 and u-blox 1771 receiver in round 1 of the measurement drive. Additionally, the elevation is illustrated.

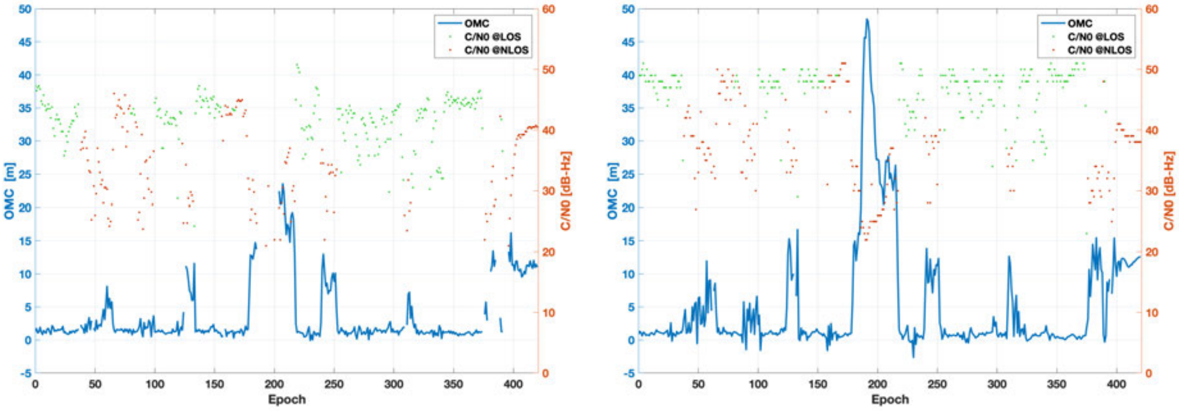
Just like the parameters that have been investigated before, the C/N$_0$ values as well show a high variation due to the fast changing surroundings. The nominal C/N$_0$ level seems to be approximately 45 dB-Hz for Javad 82 and 50 dB-Hz for u-blox 1771. Note the different sampling rate of 0.25 dB-Hz for Javad and 1 dB-Hz for u-blox.

It has to be noticed that all the outlier observations below 32 dB-Hz for Javad are of NLOS characteristic, marked with a red point. It is a similar picture for u-blox where only one observation lies below 38 dB-Hz that is LOS. But this fact cannot be turned around: not all observations with a nominal C/N$_0$ level are of LOS, especially around epoch 100 there are some NLOS signals with high C/N$_0$ for both receivers.

This means that concluding from C/N$_0$ to LOS or NLOS seems to be only possible in one direction, stating that significantly low C/N$_0$ values compared to the nominal level are probably NLOS. This can again be referred to the Fresnel ellipsoid from section 2.3.3 which explains that NLOS signals still can have a high C/N$_0$. In those cases, diffraction only takes place with a small extra path way which does not affect the signal strength much.

In figure 7.11, OMC and C/N$_0$ are displayed for PRN 25 for both receivers Javad 82 and u-blox 1771. It shows that high OMC correspond to low C/N$_0$ values. Every peak in OMC is accompanied by lower C/N$_0$ observations and vice versa. However, it cannot be inferred from an NLOS observation that the OMC at the same epoch is high or, as mentioned above, the C/N$_0$ is low. This is shown by high C/N$_0$ values in NLOS state at epochs around 70 for PRN 25 and around epoch 40 and 390 for PRN 5 where the OMC stays low.

At this point it seems reasonable to stick to C/N$_0$ when evaluating signal performance. The problem when relating between low C/N$_0$ and high OMC is that it is not proportional. For the Javad recorded PRN 25 it can be noticed that the same magnitude in OMC is reached with different levels of C/N$_0$. At epochs 130 and 240, the OMCs both peak around 10 m but the C/N$_0$ at the same time as these peaks varies around 20 dB-Hz for the first period and around 25 dB-Hz for the second. Furthermore, the C/N$_0$ values at epochs 205 to 215 scatter in the same region but with a significantly higher OMC of 15 to 10 meters.



(a) Javad 82

(b) u-blox 1771

Figure 7.11: C/N$_0$ and OMC for satellite G25 in round 1 for both receivers.

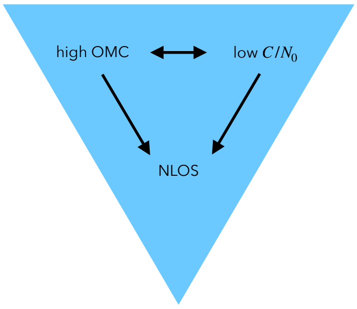


Figure 7.12: Conclusions that can be drawn between OMC, C/N$_0$ and NLOS observations.

Figure 7.12 summarizes which conclusions can be drawn between high OMC, low C/N$_0$ and NLOS observations. Note that the arrows pointing at NLOS are not pointing back, meaning a conclusion this way is not possible.

It should be added here that the fact that a satellite is of LOS character does not imply high C/N$_0$ or low OMCs either. Neither can a conclusion be drawn towards LOS observations: at epoch 70, low OMC values are in fact NLOS observations and low C/N$_0$ around epoch 340 are of LOS.

But since high OMCs and low C/N$_0$ seem to correspond, this could be helpful when adding detour lengths. A suggestion could be that detour lengths only should be applied when the C/N$_0$ value for the same epoch is low. In all probability, the OMC is then high, meaning the path length indeed has been recorded as too long. An applied detour length will make most sense in these occasions.

7.6 Classified Route Sections

For a classified analysis, three different street sections have been extracted to investigate different effects occurring in those sections. These classifications are crossings, streets in north/south direction and streets in east/west direction, see figure 7.13. With this classification, roughly 25% of the measurement data is collected at crossings, 50% at streets with north/south orientation and 25% at streets with east/west orientation.

At crossings, it is expected that more satellites are visible as less buildings are in close vicinity and the roads open the view for lower elevated satellites. And depending on the satellite constellation, visibilities can be different for streets with different orientations as well.

The barplot in figure 7.14 shows the number of observations in each section summed up for all 5 rounds for both receivers Javad 82 (7.14a) and u-blox 1771(7.14b). It is split into LOS and NLOS observations as well as the number of NLOS observations that could be corrected added onto the LOS observations. This shows the total number of "safe" observations as those are either directly visible or reconstructed.

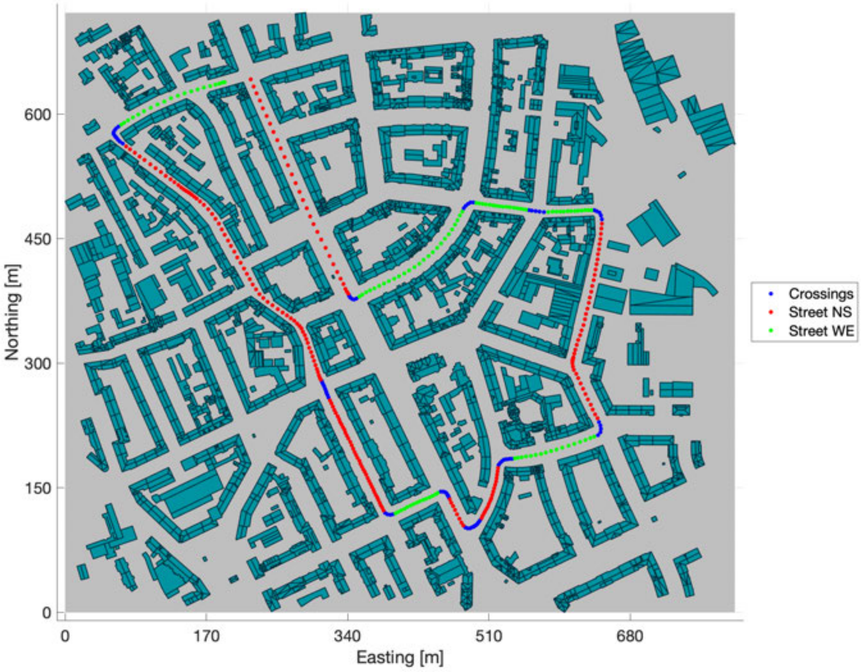
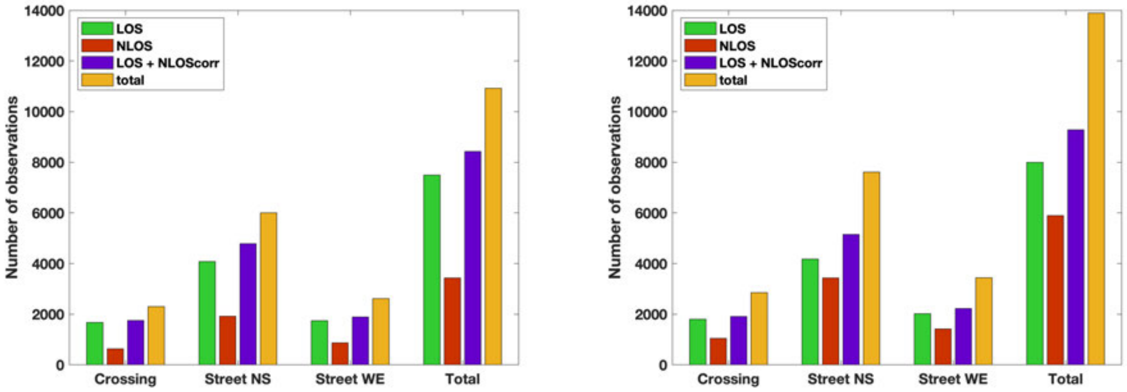


Figure 7.13: Classified route sections displayed onto the reference trajectory.



(a)

(b)

Figure 7.14: Number of observations in classified street sections for receivers Javad 82 and u-blox 1771.

Analyzing the total 14000 observations for u-blox and 11000 observations for Javad, most of them were collected in streets with north/south orientation. This is natural as the route was more north/south oriented in general. Crossings make up the shortest sections, resulting in the least number of observations for these classifications. Still, the absolute number does not differ much from the number of observations in east/west streets. This is due to the standing phases at crossings, where sometimes movements were halted within a crossing for more than one minute.

Table 7.1: Number of observations in classified street sections for receiver Javad 82.

Obstype	Crossing	Street NS	Street WE	Total
LOS	1674	4080	1741	7495
NLOS	633	1922	873	3428
LOS + NLOScorr	1749	4786	1893	8428
Total	2307	6002	2614	10923

Table 7.2: Number of observations in classified street sections for receiver u-blox 1771.

Obstype	Crossing	Street NS	Street WE	Total
LOS	1805	4180	2015	8000
NLOS	1045	3433	1422	5900
LOS + NLOScorr	1914	5149	2224	9287
Total	2850	7613	3437	13900

To go more into detail, tables 7.1 and 7.2 show the observation figures in detail. Indeed, the percentage of LOS observations at crossings is higher than in streets, making up 73% for Javad 82 and 63% for u-blox 1771. In streets, the orientation does not matter when determining the ratio of LOS observations for all satellites, making up 68% for Javad 82 and 57% for u-blox 1771.

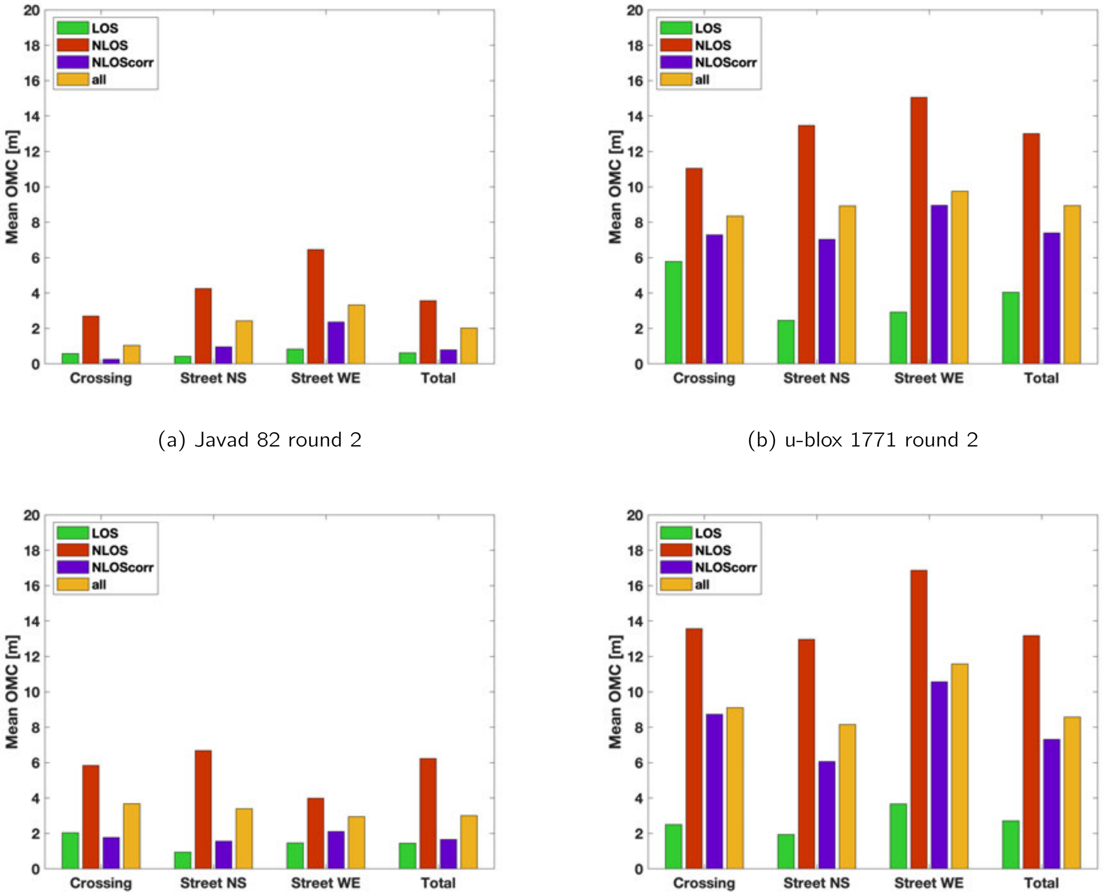
The number of NLOS observations that could be corrected shows an interesting picture. At crossings, the least amount and percentage of NLOS observations could be reconstructed, only 12% for Javad 82 and 10% for u-blox 1771. The amount is not much higher for streets in east/west direction, claiming 17% and 15% respectively. But for streets in north/south direction, the number is significantly higher. For Javad 82, it was possible to reconstruct 37% of NLOS observations here and 28% of u-blox 1771.

This must be due to the fact that there are overall more satellites in east/west direction whose signals then reflect on buildings along a street with north/south orientation. In east/west streets however, as satellites in the north are barely available and if so only in low elevations, reflections from the south would be the only possible option. At crossings, there are not many obvious reflection planes as in canyon-shaped streets, making reflections more random to occur.

When comparing the two receivers, it shows that the number of LOS observations does not differ much but the number of NLOS observations is the deciding factor. Almost every additional information the u-blox receiver is getting is of NLOS characteristic. Especially in streets with north/south orientation the higher number of NLOS is striking, reaching 45% of all received observations and 79% more than Javad is recording.

But these additional observations could not improve the number of „safe“ observations (LOS + corrected NLOS) much. Although u-blox 1771 overall recorded 72% more NLOS observations, the number of corrected NLOS is only 10% higher than Javad 82.

This means, the majority of additional NLOS observations cannot be reconstructed, making them likely to be highly disturbed and not beneficial for positioning. This is a strong counter-argument for collecting as much data as possible and high-sensitivity.



(a) Javad 82 round 2

(b) u-blox 1771 round 2

(c) Javad 82 round 3

(d) u-blox 1771 round 3

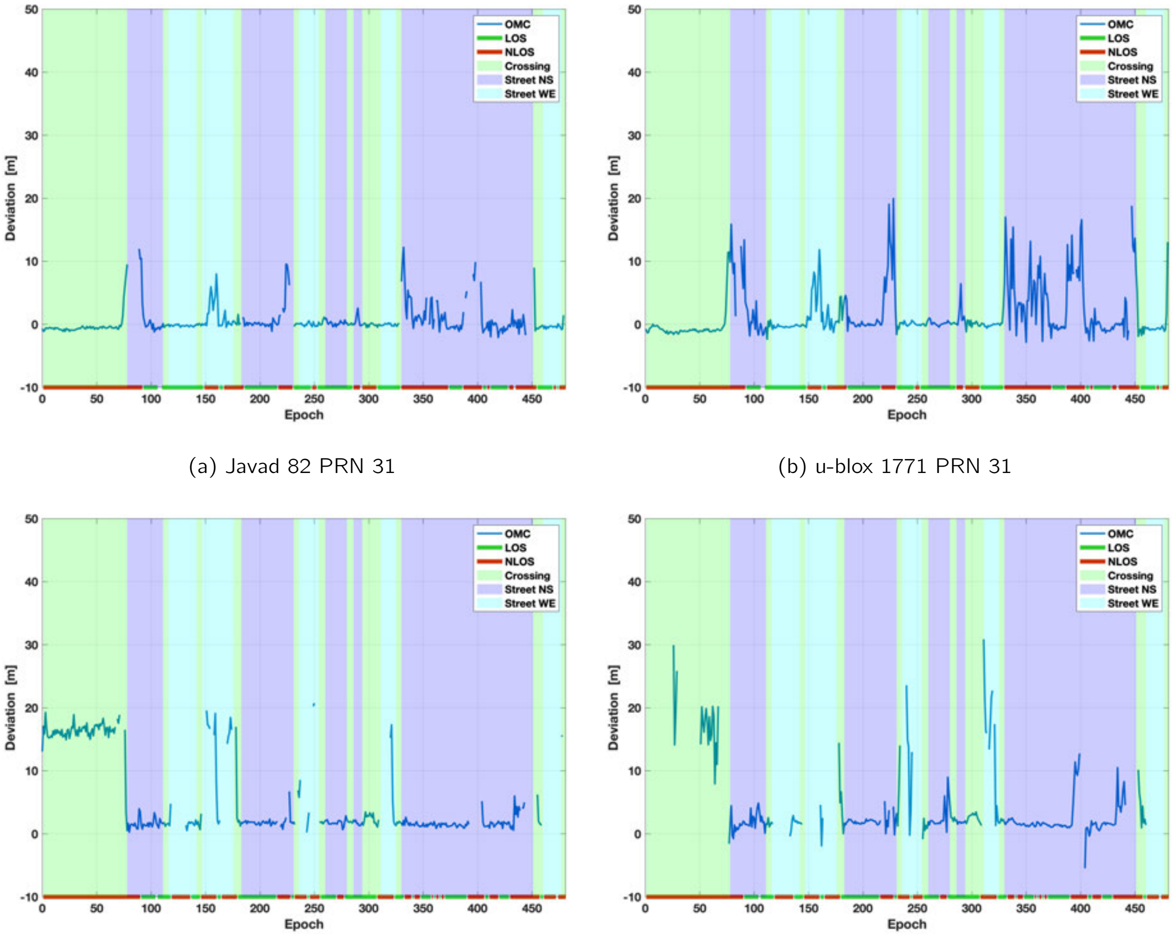
Figure 7.15: Mean OMC for classified sections for both receivers.

Figure 7.15 shows the mean OMC classified into above mentioned route sections again for receivers Javad 82 and u-blox 1771, exemplarily showing round 2 and 3.

As expected, the OMCs for NLOS observations are consistantly a lot higher than for LOS observations, no matter which receiver or which section of the route is viewed. The LOS observations show the lowest OMCs, especially for Javad 82 in round 2. Furthermore, in every case the OMC could be lowered through the correction of NLOS observations compared to the OMC for all uncorrected observations.

The difference between receivers is obvious as the u-blox OMCs are higher by a factor of 2 to 4, still the pattern between LOS and NLOS is the same. By looking at the different sections, it cannot be said clearly which regions are beneficial to achieve low OMCs. Crossings would have been expected to have lower deviations as more LOS observations could be achieved but both Javad and u-blox receiver show OMCs in similar magnitudes for every section.

But by looking at satellites individually, differences between sections are noticed. Figure 7.16 shows the individual OMC for satellites G31 and G21 in round 3 with the three classified sections displayed in the



(a) Javad 82 PRN 31

(b) u-blox 1771 PRN 31

(c) Javad 82 PRN 21

(d) u-blox 1771 PRN 21

Figure 7.16: OMC classified into sections for satellite G31 and G21 in round 3.

background. The bar at the bottom of the figure shows the LOS and NLOS phases for the respective satellite.

Satellite G31 is located at azimuth $\alpha = 232°$, elevation $\epsilon = 48°$ on average which can be considered as moderately elevated west, while satellite G21 is located at $\alpha = 184°$, $\epsilon = 39°$ which is in the south. Naturally, G31 has more LOS phases in east/west oriented streets while G21 in north/south and this corresponds to lower OMC values for these regions. Small exceptions have to be made, e.g. for G31 in the second east/west street as this is oriented almost perpendicular to the azimuth angle (see figure 7.13), resulting in more NLOS phases. Also, since the long north/south oriented street between epochs 330 and 450 is not a straight line, visibilities are changing within this section.

What has to be noticed are the sharp transitions between the sections, for satellite G31 when turning into a north/south oriented street and for satellite G21 when leaving said street. At these passes, the OMC drops or rises by sometimes more than ten meters. Furthermore, these transitions are not only because of a LOS NLOS transition. Surely, most of the high OMCs are in NLOS phases, but e.g. for satellite G21, the first NLOS phase continues until epoch 90 while the drop of OMC already takes place 10 epochs earlier when the road section is changed.

Hence, the overall classification parities as shown in figure 7.15 might be a result of adding all satellites from every angle, balancing the overall mean OMC.

7.7 Coordinate Impact

To see the impact the detour length has if not corrected, dx as part of the single point positioning process (see section 2.3.4) can be investigated. It is computed by determining

$$dx = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T \cdot \Delta\rho \tag{7.1}$$

with $\boldsymbol{A}$ denoting the design matrix as the epoch wise derivative of the distance ρ_A^k between station A and satellite k as

$$\boldsymbol{A} = \begin{bmatrix} -\frac{\Delta x_A^{k=1}}{\rho_A^{k=1}} & -\frac{\Delta y_A^{k=1}}{\rho_A^{k=1}} & -\frac{\Delta z_A^{k=1}}{\rho_A^{k=1}} \\ -\frac{\Delta x_A^{k=2}}{\rho_A^{k=2}} & -\frac{\Delta y_A^{k=2}}{\rho_A^{k=2}} & -\frac{\Delta z_A^{k=2}}{\rho_A^{k=2}} \\ \vdots & \vdots & \vdots \end{bmatrix} \tag{7.2}$$

all given in ECEF coordinates, cf. Schön (2015). $\Delta\rho$ denotes the calculated detour lengths for each satellite and epoch. The resulting three column matrix dx now represents the coordinate impact in each X, Y, Z direction and is rotated into topocentric coordinates to see the effects in north, east and up system.

The results from topocentric dx are then classified into four regions with respect to their horizontal position impact, see figure 7.17. This means that e.g. a blue marker will tell that the reflection causes the determined coordinate to be further north east than the actual position.

The outcome is shown in figure 7.18. The figure is subdivided into two different modes. First, all determined detour lengths are taken into account in figures 7.18a and 7.18b, representing Javad 82 and u-blox 1771 receivers. Second, only those detour lengths are taken into account that match the respective OMC value to an ϵ of 3 m, confirming in a way the computation. The value of ϵ is chosen based on the code noise and experience of the detour matches so far. For each figure, all five rounds are displayed with

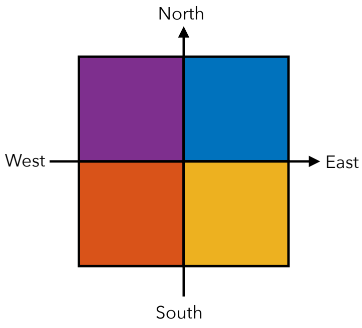
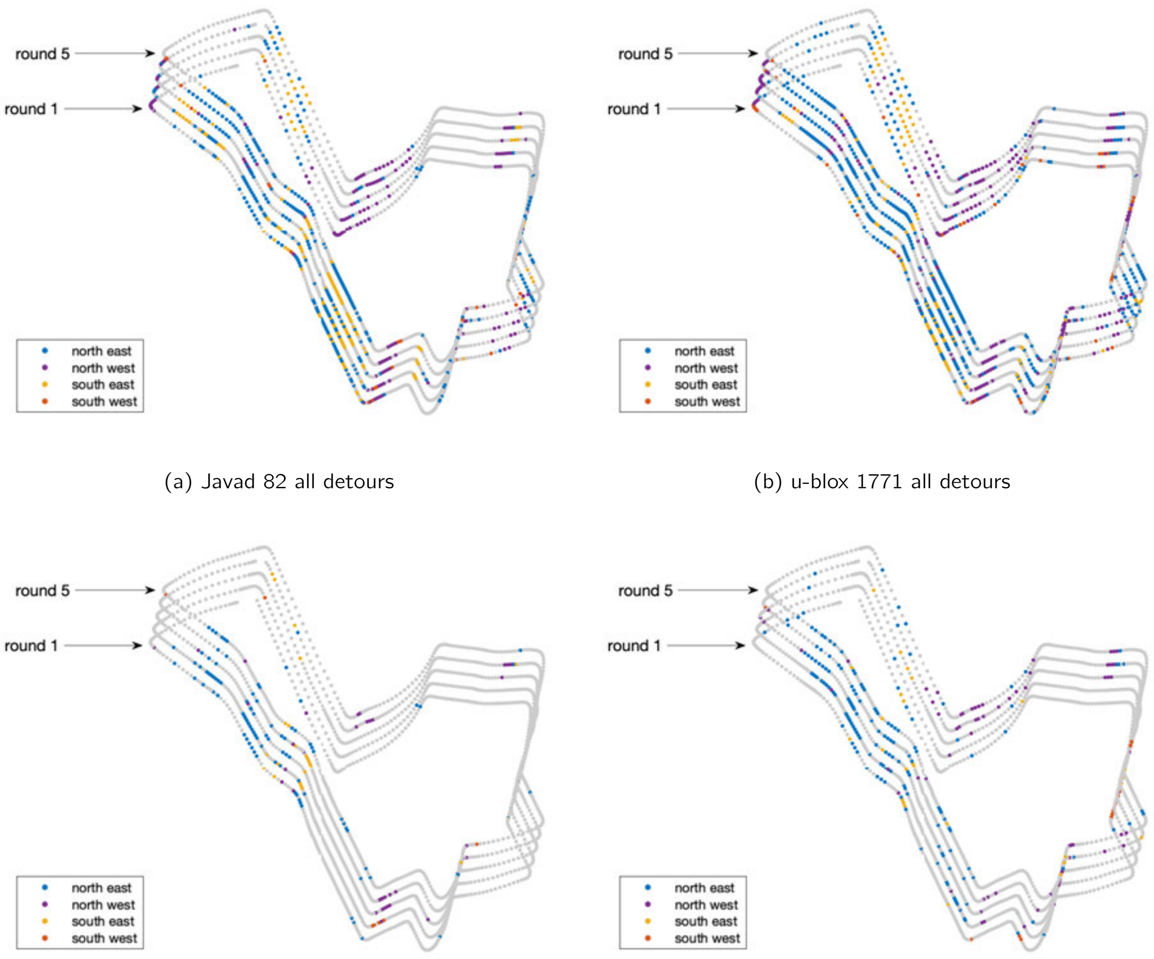


Figure 7.17: Legend for figure 7.18. Classified coordinate impact of detours.

a certain offset in north and east coordinate in order to be able to find repeatabilities.

In the case of all determined detour lengths, the effects correspond to the street classifications as of section 7.6. Also, the results for the two receivers match significantly well. For streets in north/south direction, especially for the long street in the west of the route, nearly all detours will lead to a shift to the east, displayed by blue and yellow markers. This is because of the satellite constellation at the time. All available satellites that had the possibility to post a reflection onto building surfaces in this region were coming from the west – resulting in an additional pathway to the east. It is useful to compare figure 5.2a from section 5 with the current satellite skyplot to see which satellites are probable for causing reflections.



(a) Javad 82 all detours

(b) u-blox 1771 all detours

(c) Javad 82 only detours so that $\Delta\rho$ - OMC $< \epsilon$

(d) u-blox 1771 only detours so that $\Delta\rho$ - OMC $< \epsilon$

Figure 7.18: Comparison of coordinate impact due to detour.

If present, the impact for streets with west/east orientation are repeatedly to the north west. As expected, roads with perpendicular orientation have impacts that are perpendicular as well. However, streets with east/west orientation are less likely to offer reflection surfaces for single reflections. There are more grey sections in streets with this orientation than in north/south sections. This has to have its cause in the satellite constellation for the northern hemisphere: with the so-called "north hole", there simply are less satellites available in the north, making reflections in east/west oriented streets less frequent. And if yet a reflection does occur, it has its impact to the north more likely as the reflected signal comes from the south. Furthermore, almost no impacts happen to the south west, which is also possible to correspond to satellite constellation thesis.

Figures 7.18c and 7.18d now just show an extraction of the figures above, displaying more of a projection for which regions of the experiment route the detour lengths algorithm worked well. Instead of the original approximate 400 detour lengths, the selection leads to only around 40 corrected observations for Javad and 60 for u-blox. This corresponds to 10% and 15% of the primary number of detour lengths. The part where the the most matches between OMC and detour lengths have been found were in the street of Große Barlinge in the north western part of the experiment route.

But in most of the other streets, the detour lengths do not match the OMC within ϵ, raising the question of reasons why only Große Barlinge seems to offer adequate results.

7.8 Reasons for Non-Matching Detour Lengths

As mentioned above, there are road sections in which the computed detour lengths match the OMC well but also many segments where the OMC do not match at all.

Possible causes for disparities might be

1. The model that is used is not accurate enough in some regions.
2. Vegetation or other obstacles prevent the reflection from happening on the actual calculated spot on the building surface.
3. Instead of a reflection, the signal is diffracted or affected by multipath.

All these reasons are looked at in detail in the following.

1. To evaluate the first reason i.e. buildings in reality are not modeled accurately enough, a look at the building's representatives is given.

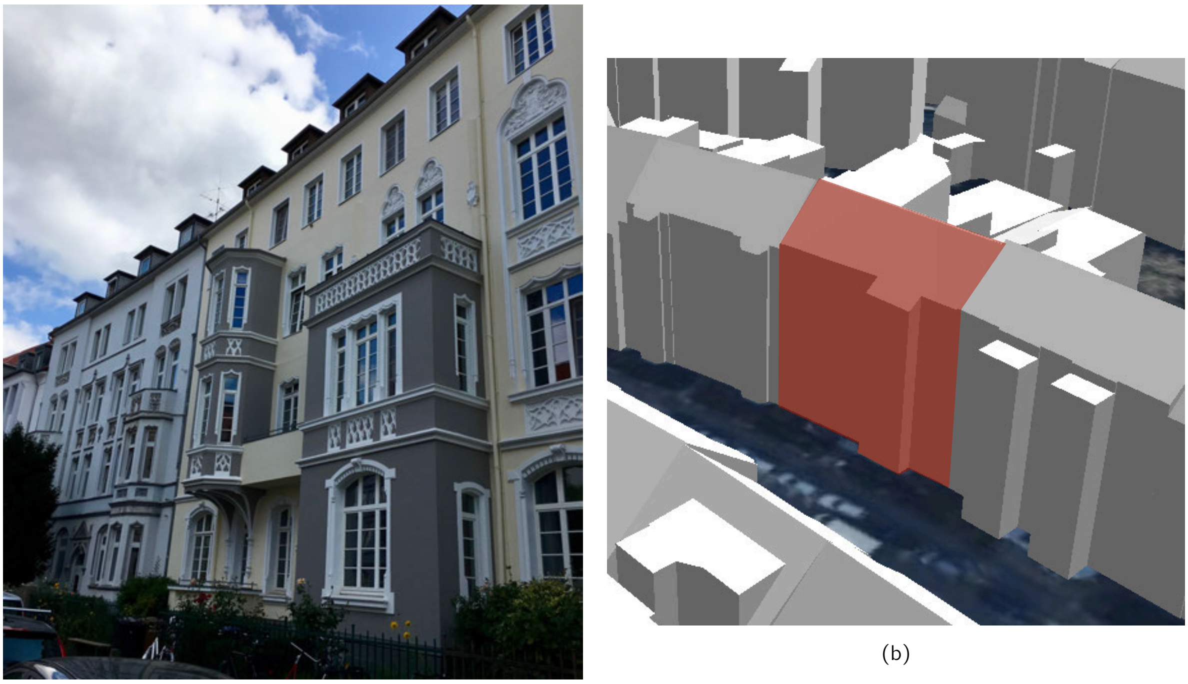
   Figure 7.19 shows an example of a building in Stolzestraße. The picture of figure 7.19a states it is a typical old building of Hanover of late 1800's early 1900's with details in the facade and two large oriels. However, these significant attributes are not depicted accurately in the CityModel. The marked building in figure 7.19b only indicates one oriel which does not have the actual extent metrics. This will lead to a deviation of a few meters when calculating the detour and comparing it to the actual pathway. Furthermore, since the oriel on the left of the building is shaped as a semicircle, reflections on this surface will deflect the signal in another direction, also resulting in errors. The balcony in between the two oriels is not modeled either.

   This building is just an example for many cases in Stolzestraße since the neighboring houses are similar. Therefore, the disparities in the CityModel compared to reality might substantiate the differences in OMC and detour lengths.

   Especially if we take a look at the street of Große Barlinge where the computed detour lengths and OMC do match quite well. This is seen in figure 7.20. The buildings in this street are a lot simpler: no oriels, sometimes balconies, but no heavy details. Due to the simplicity, the buildings match the model a lot better. This means, a correct application of the detour algorithm is more likely.

2. The accuracy of the model is not the only problem causing disparities. At many places, obstacles are to be found that do not appear in the model at all.

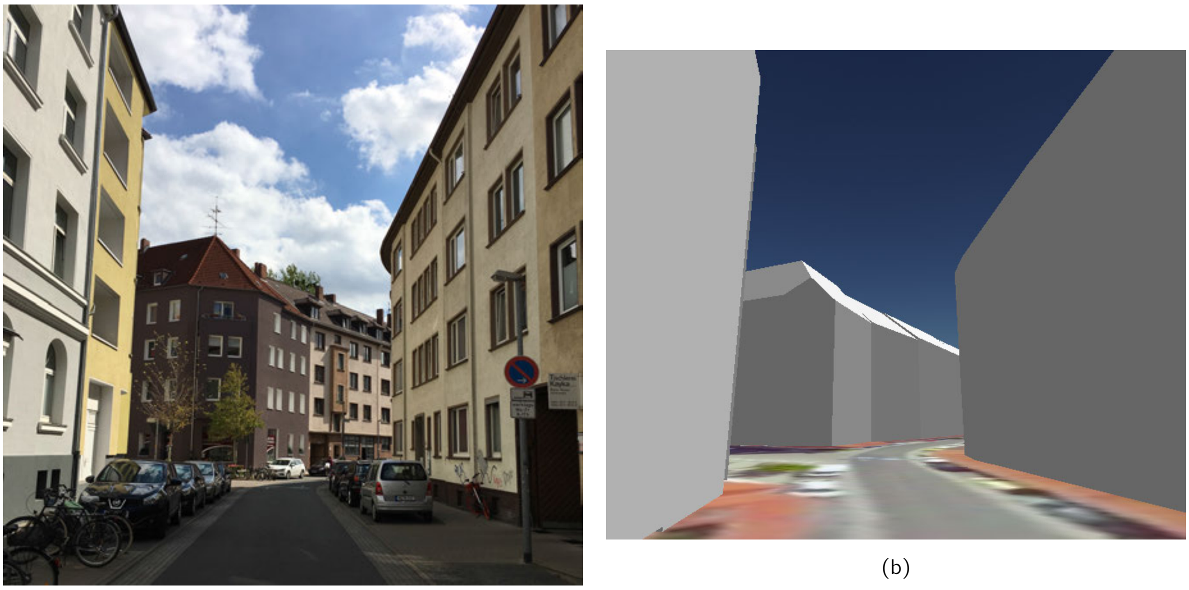
   Figure 7.21 shows two examples. In the first figure 7.21a, some large trees in Simrockstraße block the right side of buildings almost completely. If a reflection is found with the algorithm on that side, it is implausible that the reflection actually happened as straight as it has been computed.



(a)

(b)

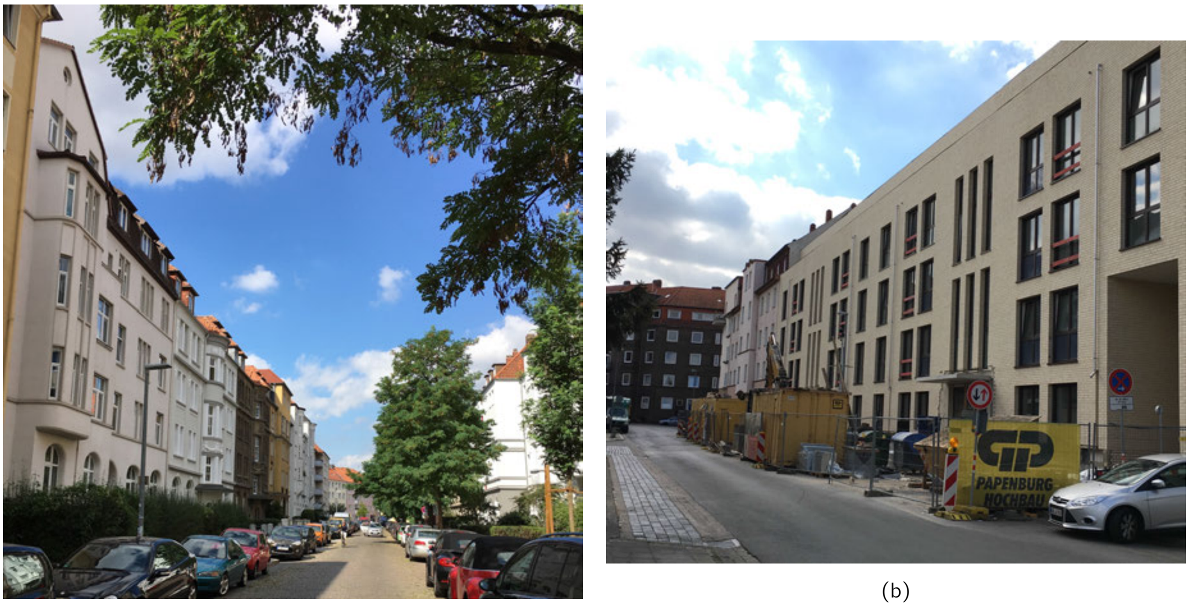
Figure 7.19: Comparison of buildings in Stolzestraße in the CityModel and in reality: (a) Stolzestraße no. 53 at site, (b) Stolzestraße no. 53 in CityModel (Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation, 2018)



(a)

(b)

Figure 7.20: Comparison of buildings in Große Barlinge in the CityModel and in reality: (a) View from Große Barlinge towards south at Stolzestraße crossing at site, (b) View from Große Barlinge towards south at Stolzestraße crossing in CityModel (Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation, 2018)



(a)

(b)

Figure 7.21: Obstructions in the streets of Simrockstraße and Tiestestraße: (a) trees in Simrockstraße, (b) construction site at Tiestestraße

The second figure 7.21b shows a construction site in Tiestestraße which is of course not displayed in the model. Also, constructions raise the question whether the model is up to date. To have a model as accurate as possible, changes in buildings have to be updated.

3. For the third reason we can find signs in a comparison between C/N_0 and OMC, e.g. in figure 7.22. If the C/N_0 stays high while the satellite is not visible, it is an indication that although the direct line-of-sight might not be available, diffraction will help the signal to bend around the blocking obstacle. It is also possible that the signal is affected by multipath. If a detour length at this time is then computed, it will have a wrong magnitude as the signal actually was not reflected but only diffracted, causing the additional pathway to be a lot shorter.

   Inside the marked box of figure 7.22, this phenomenon can be observed. Although the satellite is marked as not visible, the OMC stays relatively low, showing wave-like peaks approximately two meters of magnitude. A reflection surface could be found for a few epochs but since it is reasonable to assume that the signal is only diffracted instead of reflected, the computed detour length is around 15 m too high.

7.9 SPP Solution

In the last step, we perform an SPP solution with the help of RTKlib (Takasu, 2013) to see what an actual positioning solution will look like with manipulated observations.

As a preparation, the detour lengths $\Delta\rho$ with respect to satellite and epoch are subtracted from the original code observations, leading to a "corrected" solution. These are then investigated in RTKlib with settings of elevation cutoff of 15°, broadcast ionospheric corrections, Saastamoinen tropospheric corrections, precise satellite ephemeris and GPS only.

Figure 7.23 displays a comparison between a SPP solution with the original observations and the manipulated ones. As expected from the analyses before, in many occasions the detour lengths lead to a shift

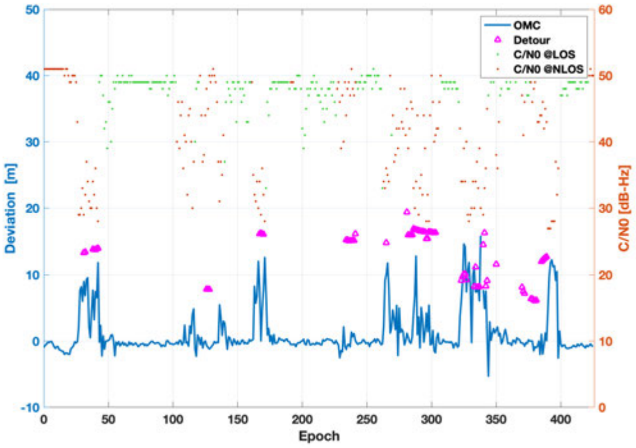
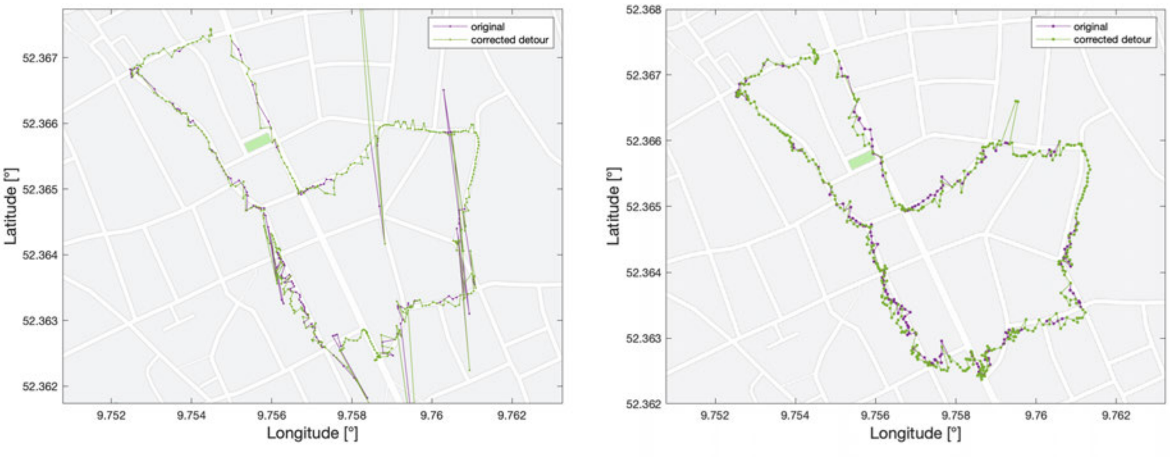


Figure 7.22: C/N$_0$, OMC and computed detour lengths for satellite G31 in round 2 for receiver u-blox 1771. Detour lengths are too high at epochs within the marked box.



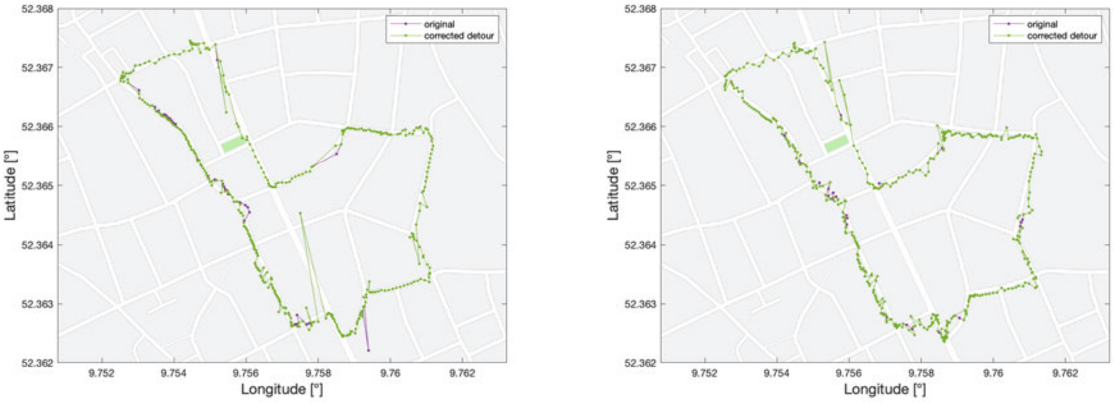
(a) Javad 82

(b) u-blox 1771

Figure 7.23: SPP trajectory comparison in round 1 with corrections from all computed detour lengths.

in the coordinates that is too far from the actual position. While the displacements are still moderate for u-blox in figure 7.23b with a few meters next to the roads, the solution for Javad 82 shows some huge outliers in figure 7.23a.

It concludes that the consideration of all detour lengths is simply overshooting as corrections are too high.



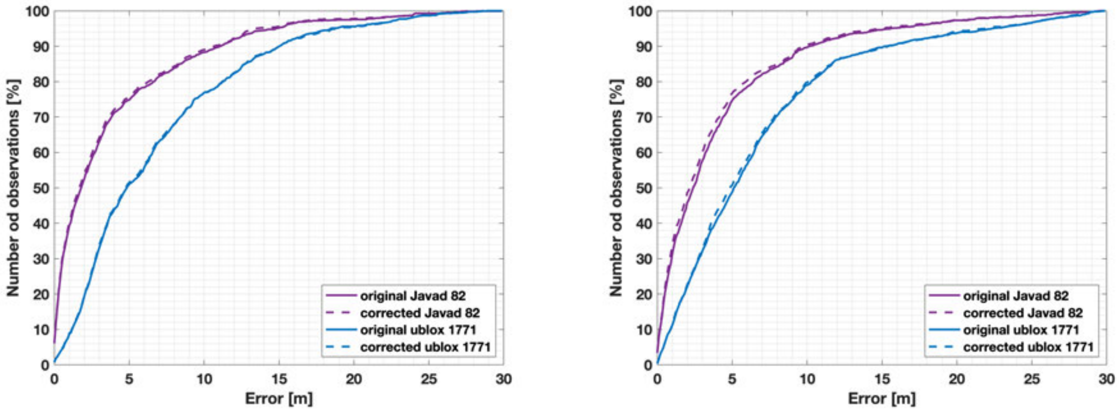
(a) Javad 82

(b) u-blox 1771

Figure 7.24: SPP trajectory comparison in round 3. Only detour lengths that match the OMC to ϵ are considered.

In figure 7.24, only detours that match the corresponding OMC to an ϵ of 3 m are considered and applied in the SPP solution. Obviously, only at some points the correction through detour lengths meet these requirements. However, at these specific points the positioning solution is indeed better than the unmodified version.

For Javad receiver in round 3 in figure 7.24a, an outlier in the south of the route is pulled back onto the street. Also, there is a span of five epochs in the middle section of the street in the west where all of those originally are positioned off the road but in the corrected version all lie on the street. Especially this case where for a period of a few epochs the positioning got better is a sign for the algorithm to have a positive effect.



(a) round 2

(b) round 5

Figure 7.25: Difference of SPP solutions to reference trajectory in a cumulative distribution.

However, because only some detour lengths are picked out and these have been validated before, the impact is expectedly low. This is confirmed by figures 7.25 which show a cumulative histogram of the error of the SPP solution compared to the reference trajectory. The fact that the corrected solution is slightly better for all rounds (rounds 2 and 5 are displayed exemplarily) is not surprising as the corrections were preselected. But it can be seen as well that due to the very few points where those are applied, the difference to the original is small, not even reaching 1% of improvement.

7.10 Conclusion

In this chapter, a thorough investigation has been carried out on the kinematic experiment. The satellite visibility, DOP values, reflection process and OMC have been analyzed also with respect to C/N_0, detour lengths and LOS/NLOS characteristics. Additional discoveries include those in the position domain.

The key findings are concerning the OMC in context of LOS/NLOS, C/N_0 and classified route section. Generally, NLOS observations have higher OMC values and these are distributed more randomly than the OMC values of LOS observations. However, the difference between the two receivers Javad 82 and u-blox 1771 is that due to the recording of many NLOS signals for u-blox, there are more high OMC. High OMC can be seen as indicator for NLOS observations as well as low C/N_0 but this conclusion cannot be drawn the other way around. In the context of classified road sections, OMC are more likely to be high in streets with an orientation that is perpendicular to the satellite's azimuth angle. Out of all detour lengths, 10% match the Javad 82 OMCs to a threshold of 3 m and 15% for u-blox 1771. An SPP solution with only matching detour length corrections in the observations is expectedly better than an uncorrected solution.

Reasons for the disparities between OMC and detour lengths include inaccuracies in the CityModel, obstacles in the signals pathway that are not modeled and diffraction and multipath effects instead of reflection. But this is dependent on the street that is considered. In some, the model matches well and the detour lengths do fit the OMC but in others, problems occur more frequently.

8 Conclusion

8.1 Summary

During the course of this work, an algorithm has been developed that correctly determines the LOS status of a satellite based on a CityModel and computes the detour lengths through the signal's reflection on a building surface. A kinematic experiment with multiple receivers has been conducted to investigate LOS/NLOS characteristics in the kinematic sense and to inspect what improvements are achieved in the position domain by correcting observations with the computed detour lengths.

High DOP values for LOS only observations show that the availability of a positioning with LOS only satellites is not sufficient. It has been noted that the u-blox receivers record a high number of NLOS signals which is significantly higher than those the Javad 82 receiver received. These NLOS observations have a high OMC value and subsequently, the u-blox receivers generally have a higher OMC level than Javad 82. However, the matching rate of determined detour length and OMC for NLOS signals is higher for u-blox 1771 receiver than for Javad 82, but is an overall low rate with 15% of OMC that fit the detour length to 3 m of difference. Out of the matching detour lengths, most of them were found in one street only. This specific street is characterized with simple buildings rather than with oriels and other additional objects as in other streets. It could be also noticed that at some epochs, the C/N$_0$ value is at nominal level and the OMC is low, but still because the satellite was not in sight, a detour length was computed. These cases lead to overcorrections with negative effects for positioning solutions.

An SPP solution only with matching detour lengths subtracted from the observed code lengths fits the reference trajectory slightly better than with original data, but is worse if all computed detour lengths are applied. Reasons for not-matching detour lengths can be found in inaccuracies of the CityModel and in diffraction effects that are neglected. Additionally, the size of the Fresnel ellipse around the reflection point is of a dimension of several dozens of square meters, which means that the assumption that satellite reflection can be modeled in a straight-line approach is not sufficient for many cases. With this approach, diffraction and multipath effects are neglected as well as the impact of the different surfaces inside the Fresnel ellipse.

8.2 Outlook

Generally, the concept of detour lengths modeled from the Urban Trench is demonstrated be appropriate as many examples show where the OMC values match the computed additional path length. The problem is that other factors also come into place that the CityModel cannot catch. Hence, the application of detour lengths on the code observations should be dependent on

- the C/N$_0$ value at the respective epoch
- the size of the Fresnel ellipse around the reflection point.

As shown, high C/N$_0$ indicate that the signal is unlikely to be reflected. Other effects might have occurred instead. Fresnel ellipse sizes indicate the amount of objects and surfaces that have an impact on the reflection process. If a Fresnel ellipse is too large, it is simply unreasonable to model the reflection process with a single point.

These factors should be taken into account when working with observation corrections through detour lengths. Due to the high computational load of the CityModel, further studies should also focus on a combination of observation parameters that indicate reflections, taking only a minimum of the building model into account.

Bibliography

Ben-Moshe, B., Elkin, E., Levi, H., and Weissman, A. (2011). Improving Accuracy of GNSS Devices in Urban Canyons. In *Proceedings of the 23rd Annual Canadian Conference on Computational Geometry, CCCG 2011*.

Bender, P. (2010). Eine einfache Formel für den Flächeninhalt von Polygonen. *Von Geometrie und Geschichte in der Mathematikdidaktik*, pages 53–70.

Betaille, D., Peyret, F., Ortiz, M., Miquel, S., and Fontenay, L. (2013). A New Modeling Based on Urban Trenches to Improve GNSS Positioning Quality of Service in Cities. *IEEE Intelligent Transportation Systems Magazine*, 5(3):59–70.

Boehm, J., Heinkelmann, R., and Schuh, H. (2007). Short note: A global model of pressure and temperature for geodetic applications. *Journal of Geodesy*, 81:679–683.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. The M.I.T. Press Massachusetts Institute of Technology Cambridge, Massachusetts, 2nd edition.

Csati, Z. (2017). collinear.m. MathWorks,https://de.mathworks.com/matlabcentral/fileexchange/62737-collinear.

Falkena, W. (2012). Xml2struct.m. MathWorks, https://de.mathworks.com/matlabcentral/fileexchange/28518-xml2struct.

Google Maps (2019). Google Maps Representation of Hanover Police Department. https://www.google.com/maps/@52.365639,9.7329722,272m/data=!3m1!1e3.

Gröger, G., Kolbe, T. H., Nagel, C., and Häfele, K.-H. (2012). *OGC City Geography Markup Language (CityGML) Enconding Standard*. Open Geospatial Consortium, 2.0.0 edition.

Groves, P. D. (2011). Shadow Matching: A New GNSS Positioning Technique for Urban Canyons. *The Journal Of Navigation*, 64(3):417–430.

Haines, E. (1994). Point in Polygon Strategies. *Graphics Gems IV*, pages 24–46.

Hannah, B. M. (2001). *Modelling and Simulation of GPS Multipath Propagation*. PhD thesis, Queensland University Of Technology.

Hartmann, K. (2019). Towards visibility geometry and quality assessment of high sensitivity GNSS equipment under urban conditions. Leibniz University Hanover.

Hofmann-Wellenhof, B., Lichtenegger, H., and Wasle, E. (2008). *GNSS - Global Navigation Satellite Systems. GPS, GLONASS, Galileo and more*. SpringerWienNewYork.

Hristov, H. D. (2000). *Fresnel Zones in wireless links, zone plate lenses and antennas*. Artech House Inc. Norwood, MA, USA.

Hsu, L.-T. (2017). Analysis and modeling GPS NLOS effect in highly urbanized area. *GPS Solutions*, 22(1):7.

International GNSS Service (IGS) (2019). GNSS Atmospheric Products. https://cddis.nasa.gov/Data_and_Derived_Products/GNSS/atmospheric_products.html.

Javad GNSS Inc. (2019). Description of Javad Delta. https://www.javad.com/jgnss/products/receivers/delta-3.html.

Kumar, R. and Petovello, M. (2014). A novel GNSS positioning technique for improved accuracy in Urban canyon scenarios using 3D city model. *27th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2014*, 3:2139–2148.

Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation (2017). *Produktinformation Digitales 3D-Stadtmodell*. Landeshauptstadt Hannover, Rudolf-Hillebrecht-Platz 1, 30159 Hannover, 3rd edition.

Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation (2018). Hannover 3D - Virtual City Map. https://stadtmodell-test.hannover-stadt.de/HostingMap-extern/#/.

Leick, A., Rapoport, L., and Tatarnikov, D. (2015). *GPS Satellite Surveying*. Wiley, 4th edition.

Mansfeld, W. (2004). *Satellitenortung und Navigation*. Friedr. Vieweg und Sohn Verlag, 2., überarbeitete und ergänzte auflage edition.

MATLAB (2019). MATLAB documentation R2019a. http://de.mathworks.com/help/.

Möller, T. and Trumbore, B. (2005). Fast, Minimum Storage Ray/Triangle Intersection. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA. ACM.

National Geodetic Survey (2019). Antenna Calibration Data Sheet. https://www.ngs.noaa.gov/ANTCAL/LoadFile?file=ngs14.atx.

OpenStreetMap contributors (2019). Map Representation of Hanover Südstadt.

Rao, B. R., Kunysz, W., Fante, R. L., and McDonald, K. (2013). *GPS/GNSS Antennas*. Artech House Publishers, Norwood, USA.

Ron Lake, G. S. I. (2019). Introduction to GML. https://www.w3.org/Mobile/posdep/GMLIntroduction.html.

Rost, C. (2011). *Phasenmehrwegereduzierung basierend auf Signalqualitätsmessungen geodätischer GNSS-Empfänger*. PhD thesis, Deutsche Geodätische Kommission bei der Bayrischen Akademie der Wissenschaften, Reihe C, Nr. 665.

Schön, S. (2015). Lecture Positioning and Navigation in winter semester 2015 at Leibniz University Hannover.

Smits, B. (2002). Efficient Bounding Box Intersection. *Ray Tracing News*, 15(1).

Takasu, T. (2013). *RTKLIB ver. 2.4.2 Manual*.

Terratec AS (2011-2019). TerraPOS software.

Trimble (2017-2018). GNSS Planning Online. https://www.gnssplanning.com/.

Tuszynski, J. (2018). Fast vectorized triangle/ray intersection algorithm. Technical report, MathWorks.

u-blox AG (2019). Description of u-blox EVK-M8. https://www.u-blox.com/de/product/evk-8evk-m8.

ugt holding (2019). Description of Javad GrAnt G3T. http://ugt-holding.com/shop/product/antenna-grant-g3t.

Weinbach, U. and Schön, S. (2011). GNSS receiver clock modeling when using high precision oscillators and its impact on PPP. *ASR*, 47(2):229–238. DOI: 10.1016/j.asr.2010.06.031.

Williams, A., Barrus, S., Morley, R. K., and Shirley, P. (2005). An Efficient and Robust Ray-box Intersection Algorithm. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA. ACM.

Zimmermann, F., Schmitz, B., Klingbeil, L., and Kuhlmann, H. (2018). GPS Multipath Analysis Using Fresnel Zones. *Sensors*, 19(1).