Influence of 3D City Layout on Air Quality

Konstantinos Mastorakis
Wessel de Jongh
Giulia Ceccarelli
Imke Lánský
Jinglan Li

Influence of 3D City Layout on Air Quality

by

Konstantinos Mastorakis
Wessel de Jongh
Giulia Ceccarelli
Imke Lánský
Jinglan Li

for the Synthesis Project of the MSc Geomatics program.

Student numbers:	4844238, 4433017, 4924606, 4973372, 4781937	
Project duration:	April 24, 2019 – June 28, 2019	
Supervisors:	C. Garcia Sanchez,	TU Delft
	J. E. Stoter,	TU Delft
Client:	S. Teeuwisse	RIVM
	Gemeente Amsterdam	

[Image front page, *source*: `freepik.com`]

TU Delft

Abstract

In 2021, noise pollution monitoring will be mandatory in the Netherlands, which requires data on traffic that can be re-used for air quality estimation models. One of the important input parameters for the latter is the street type, which is required by the dilution parameterisation used within the air quality model.

The goal of this project is to show whether automatic street classification[1] for air quality estimation is feasible and reliable, considering the geo-spatial data currently available in The Netherlands. The motivation for this project originates from the common data used in noise and air quality monitoring tools by the *Dutch National Institute for Public Health and the Environment*, (RIVM).

Currently, street classification is performed manually by many municipalities. The larger municipalities are legally obliged to monitor air quality levels, which makes use of the street types. Automating the process by using existing datasets can save a lot of time, costs, and resources, while providing standardised results in comparison to manual classification. In addition, our method is extendable to the whole of the Netherlands. Consequently, our method can have a large societal impact, since it allows the provision of air quality estimations for all municipalities; even those that are not yet required to do so. To our knowledge, no similar work has been conducted in this field, which made it even a bigger challenge.

The implementation of the automatic classification algorithm, which is thoroughly explained in this report, shows very promising results. We first tested the approaches in a small area, the Weesperstraat in Amsterdam, where we have success rates from 76.7% to 83.3% for the different classification methods when compared to the NSL classification. After evaluating the performance of each of the methods, the optimal approach has been tested on larger areas where visual inspection shows a priori promising results as well.

In addition to the automatic classification algorithm, air quality measurements with new *Flow* sensors from *Plume Labs* were performed in the city of Amsterdam. The goal was to investigate whether different street types can be identified through the use of small air quality sensors. The limited measurements did not provide distinct patterns for the different street types, and therefore identification based on pollutant concentrations was not possible within the project.

We hope that the results of this project will motivate public bodies and agencies in the Netherlands to invest in automated workflows using currently available and high accuracy geo-spatial data. This can potentially improve their efficiency, while creating a more standardised and scalable framework.

[1] Based on the ‘*Standaard Reken Methode - 1*’ (SRM-1) air quality estimation model.

Acknowledgements

This project would have never been possible without the support of many professionals from different technical backgrounds.

First and foremost, we would like to express our deepest gratitude to our project supervisors *Clara Garcia-Sanchez* and *Jantien Stoter*, for all the time and effort they invested on us. The support they gave us was remarkable and unconditional. Their attitude towards us was impeccable, promoting us to move forward feeling confident about our progress and skills. Special thanks should be given to *Clara* for joining us in the measurement campaign in Amsterdam, a rather exhaustive task. She also invested a lot of time helping us with the processing and interpreting of the air quality measurements.

Credits should also be given to *Sander Teeuwisse*, our client from RIVM, whose contribution in understanding the technical details and requirements of the SRM-1 model was crucial. He also provided us with background information regarding air quality measurement procedures, and gave useful feedback on the algorithm results.

We are also grateful to *Ravi Peters*, who gave us a way to deal with associating buildings with street segments, by introducing the Voronoi approach.

Finally, we would like to thank *Plume Labs*, for providing us with the raw data of the measurements from their sensors, which we used in our measuring campaign.

Concluding, as a student group we found the whole project a very useful and positive experience. With many ups and downs, through lots of decision making points and tasks to tackle, we finally came up with highly promising end results. We hope our work will be of great importance and of high impact for many municipalities and agencies, helping them to automate procedures and to improve their workflow efficiency by orders of magnitude.

Contents

1 Introduction

By the first of January 2021 the "Dutch Environmental Legislation Act" [26] will be enforced, making it mandatory for all Dutch municipalities to report, among others, their noise sources. To provide the noise emission data, the municipalities will have to collect data such as street traffic intensities, street speed limits, etc.

Nowadays, it has become mandatory for some municipalities such as Amsterdam to report on their air quality levels. Consequently, the Ministry of Infrastructure and Water Management prescribes the method for collecting this information, and the bigger municipalities then collect the required information themselves. This includes categorising street types based on a manual classification method. The re-use of already collected data is becoming more and more crucial for public and private agencies; the National Institute for Public Health and the Environment (Rijksinstituut voor Volksgezondheid en Milieu, RIVM) is no exception to this.

The *Standaard Reken Methode - 1* (SRM-1) model designed by RIVM to model air pollution [30], makes use of similar data to the noise pollution model. SRM-1 also requires street types, which are not necessary in the noise model. The street type is derived from the urban canyon layout, based on the street width and building heights.
Based on statements from our client (RIVM) and our understanding of the current situation, there are two goals set regarding air quality monitoring. Apart from creating a tool to model air quality levels, the side goal is to make information on street types, traffic intensity, traffic speed etc. publicly available and transparent to all citizens.

Currently, the legislation on European and National level is not so strict regarding the maximum allowed values of pollutant concentrations and penalising policies [22, 25], unless extreme violation of terms occurs [7, 8]. On the other hand, guidelines of the World Health Organisation (WHO) recommend lower pollutant concentrations to benefit human's health [21]. If these guidelines would be translated into National legislation, a broader part of Netherlands would potentially be eligible for monitoring. In this sense, air quality data could prove to be of crucial importance from a municipality up to a national level.

Considering that the trend in air quality monitoring is moving towards a more rigorous framework, this venture can only be seen positively. It is future-proofing while setting the foundations for official procedures to be built upon, if and when the situation changes.

Problem Statement

In the case of Amsterdam, the municipality collaborates in the '*Nationaal Samenwerkingsprogramma Luchtkwaliteit*' (NSL) [4], which makes the municipality the responsible party to conduct the air pollution monitoring. After verbal discussion with our client, it came to our knowledge that the street type classification is carried out manually by using Google's street view service, and by people going into the field to assess the streets. The required values, i.e. street width and building heights, are rough estimations based on visual information. Digital tools, like Geographic Information Systems (GIS), are also used. All these different methods are performed manually; they can introduce a lot of human errors and are time and resource intensive. This negatively affects the classification accuracy.

This method induces a lot of human error in these values (e.g. visually estimating the height of buildings and their distances to street), affecting the classification accuracy, while it is extremely consuming in both time and resources.

Automating the classification procedure has plenty of benefits. By designing and implementing an algorithm using up-to-date geo-datasets, street types could be identified automatically in a fraction of the initial time. Moreover, any kind of human error would be eliminated. From a geometrical point of view the accuracy would also be significantly higher. Finally, the biggest advantage of automating is the scalability of the algorithm. Possessing all the required data covering Netherlands, it would be a matter of choice to either focus on an individual municipality, or scale up to the whole country.

So, the research question is: *"Would it be feasible to automate the whole process of street classification up to the scale of a municipality like Amsterdam and still have valid results?"*

It should be clarified that the project goal is to automate the street classification process. This means automating the extraction of one and only one input parameter of the air pollution model. We are not estimating the model values themselves.

Air quality measurements were also carried out with handheld air quality measurement sensors, purchased specifically for this project. The purpose of this venture is both to see how well the sensors perform, and to identify patterns of pollutant concentrations for the different street types.

This generated the second research question: *"Would it be possible to identify the street type based on the sensor measurements?"*

The aim of this report is to provide the reader with an insightful understanding of all the aspects that were tackled during the research procedure. This includes problems that could not be resolved or were solved partially. It will also present the results of the research, followed by a final conclusion and elaborate on future work and suggested improvements.

More precisely, in the next chapter the WHO guidelines and the European Legislation on air quality will be presented, alongside with the framework implemented by RIVM for estimating the air quality. Chapter 3 will focus on the geo-datasets that are used as input, and on the pre-processing that was carried out on these datasets. Chapter 4 will cover an in depth explanation about the core task of this project, the street classification procedure including all the technical details needed to grasp how the algorithm operates. Chapter 5 discusses the air quality measurements campaign that we set up, measurements locations and give details on *Plume Lab Sensors*. Chapter 6 will present the results of the algorithm, together with an extensive evaluation of the output that was produced by implementing different alternatives of our algorithm (e.g. different classification method and input parameters). In the final chapter we will present our conclusions and discuss the limitations and shortcomings of the algorithm, and propose methods on how this algorithm can be further improved.
At the end of this report we hope that the reader will get a clear picture of the project, understanding the underlying concepts and the small details that affected our decision making process.

Lastly, the amount of time invested on this project was 10 weeks in total. This includes the brainstorming process, which was required because according to our knowledge, nothing similar has been attempted before.

2
Theoretical framework

Air pollution has been a major political issue in Europe and worldwide since the 1970s [6]. According to the World Health Organisation (WHO), it represents the main environmental risk in Europe, affecting mainly people who live in urban areas [19]. In this chapter, we describe the air pollution sources, the main air pollutants and their effects on humans' health. Furthermore, we introduce the instruments and legislative framework that limit their emissions, enforcing better air quality. Lastly, Dutch policies and the Dutch model used to calculate air quality, SRM-1 ('*Standaard Reken Methode - 1*'), are discussed.

2.1. Air pollution

Air quality depends on pollution emissions and different external factors. Air pollution emissions are generated from different natural and man-made sources; these are classified as primary emissions if substances are directly released in the atmosphere, and secondary emissions if they derive from chemical reaction with other substances [1].

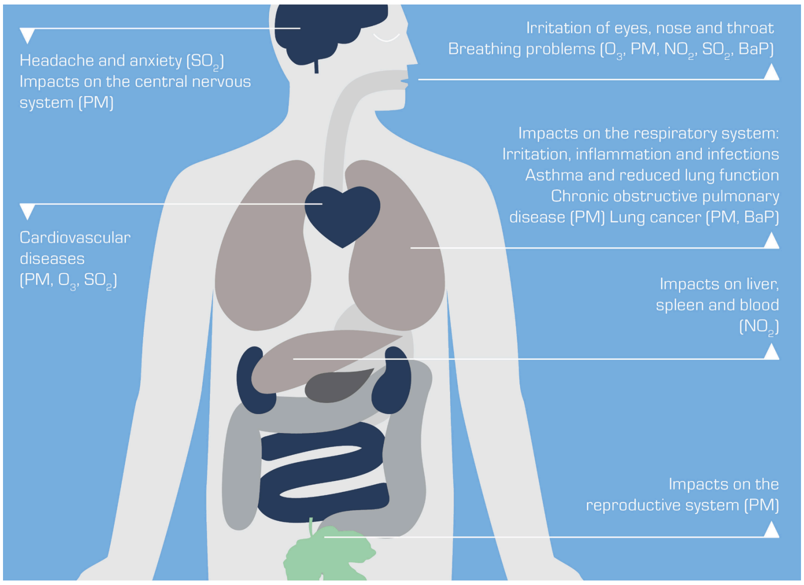

Headache and anxiety (SO_2)
Impacts on the central nervous system (PM)
Irritation of eyes, nose and throat
Breathing problems (O_3, PM, NO_2, SO_2, BaP)
Impacts on the respiratory system: Irritation, inflammation and infections Asthma and reduced lung function Chronic obstructive pulmonary disease (PM) Lung cancer (PM, BaP)
Cardiovascular diseases (PM, O_3, SO_2)
Impacts on liver, spleen and blood (NO_2)
Impacts on the reproductive system (PM)

Figure 2.1: Air pollution - Health effects. *Source*: European Environmental Agency.

Particle Matter (PM) and Nitrogen Dioxide (NO_2) represent the primary air pollutants in urban areas; around 40% of their emissions is caused by road transport [1]. PM consists of solid and liquid substances, such as pollen and Black Carbon. It is classified on its dimension in $PM_{2.5}$ (fine particles with a diameter up to 2.5μm) and PM_{10} (coarse particles with a diameter up to 10μm) [19]. In particular, $PM_{2.5}$ derives mainly from combustion, while PM_{10} derives from mechanical processes. Particle Matter negatively impacts the nervous system, can lead to breathing problems and cardiovascular diseases.

NO_x generates from combustion sources, and affects the liver, spleen and blood [2, 3]. Some of these health effects are shown in Figure 2.1.

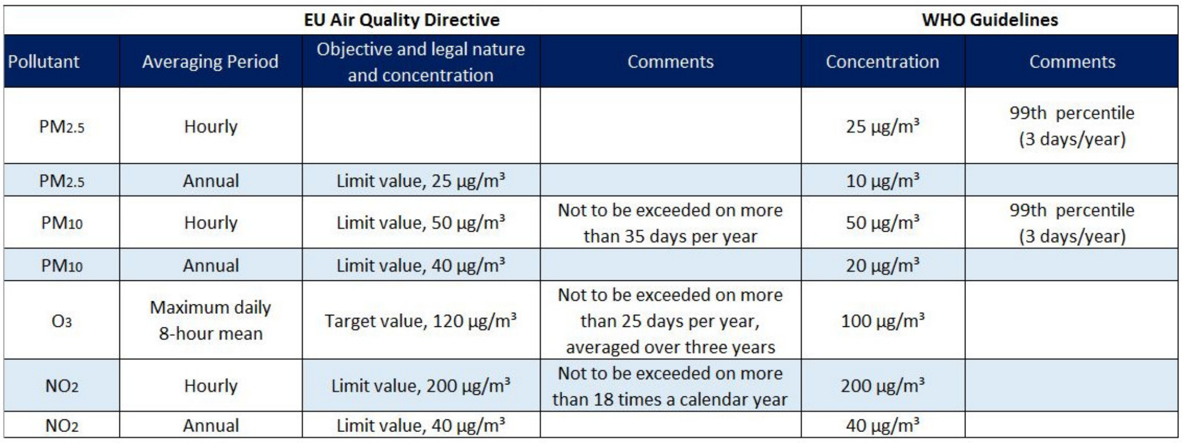
Other factors that influence air quality are the proximity to the source and the altitude at which pollutants are released, topography and geographical location, interactions between pollutants and local weather conditions [19]. In particular, weather conditions affect air pollution in numerous ways. Winds can transport particles from the source to the nearby areas, determining higher concentrations in zones far from the emission source. Contrarily, higher concentrations occur in the proximity of the source on wind still days. Additionally, cold temperatures prevent particles from reaching the higher atmosphere, thus elevated concentrations occur usually in the winter. However, NO_2 reacts with Ozone in the sunlight, causing upraised concentrations in spring and summer [14].

2.2. WHO guidelines and European legislation

The WHO guidelines and the European air quality directives are the main instruments fostering air quality. The WHO is a special agency of the United Nations, whose aim is to promote and coordinate international health [20]. In 2005, WHO published the “Air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide”. The document aims to offer guidance and to support decision making to limit health impacts of air pollution; the implementation of national air quality standards is then the responsibility of each country. The text focuses on four main pollutants (PM, NO, O, SO); for each, a threshold concentration value and a rationale for decisions are provided.

In Europe, the principal instruments fostering air quality are a series of Directives that set pollutants’ threshold values which must be respected by the Member States. The “2008/50/EC Directive on Ambient Air Quality and Cleaner Air for Europe” and “2004/107/EC Directive on heavy metals and polycyclic aromatic hydrocarbons in ambient air” constitute the current legislative framework. Each Member State (MS) should divide its territory into zones and agglomerations; for each, air pollution levels should be measured and modelled empirically. Each MS should then report findings to the European Commission; additionally, each MS should develop a plan to limit pollution emissions where threshold values are exceeded [6].

Figure 2.2 illustrates the limit values set by the European Union and WHO for $PM_{2.5}$, PM_{10}, NO_2 and O_3. Limit values are set for an hourly and annual period; according to WHO, annual values should have precedence toward hourly values, because occasional peaks have a smaller influence on the overall value. However, it is important to respect hourly limits [21]. Furthermore, Figure 2.2 shows that WHO guidelines are more restrictive than European ones.



EU Air Quality Directive				WHO Guidelines	
Pollutant	Averaging Period	Objective and legal nature and concentration	Comments	Concentration	Comments
$PM_{2.5}$	Hourly			25 µg/m³	99th percentile (3 days/year)
$PM_{2.5}$	Annual	Limit value, 25 µg/m³		10 µg/m³	
PM_{10}	Hourly	Limit value, 50 µg/m³	Not to be exceeded on more than 35 days per year	50 µg/m³	99th percentile (3 days/year)
PM_{10}	Annual	Limit value, 40 µg/m³		20 µg/m³	
O_3	Maximum daily 8-hour mean	Target value, 120 µg/m³	Not to be exceeded on more than 25 days per year, averaged over three years	100 µg/m³	
NO_2	Hourly	Limit value, 200 µg/m³	Not to be exceeded on more than 18 times a calendar year	200 µg/m³	
NO_2	Annual	Limit value, 40 µg/m³		40 µg/m³	

Figure 2.2: Air pollutants threshold values of the EU legislation and WHO guidelines. *Source*: European Environmental Agency.

2.3. Dutch policy

In the Netherlands, air quality standards are set in the Environmental Management Act, in accordance with European directives; in particular, limit values for PM and NO are specified in "Title 5.2: air quality requirements". Furthermore, specific elements of the law are defined in Decrees and Ministerial Regulations [11].

Administrative units and governments are responsible for measuring and improving air quality. To do so, they collaborate in the The National Air Quality Cooperation Program (NSL). The program defines actions to meet European standards, such as monitoring and assessing the impact of spatial development, ensuring citizens' health [27]. Each year, local governments publish traffic and environmental data and the progress of measurements and projects, among others. This information is then used to compute air quality.

RIVM is the responsible body for air quality calculations. These are performed using the "Rekentool" (calculation tool), commissioned by the Ministry of Infrastructure and Environment (IenM). Calculations follow three standard methods: SRM-1, SRM-2 and SRM-3. SRM-1 is applicable to compute air quality in built-up areas, SRM-2 is applicable in non-urban areas, while SRM-3 focuses on point and surface air pollution sources such as chimneys. The study presented in this report focuses on street classification in urban areas, thus we further analyse SRM-1 calculation method [30].

2.3.1. Standaard Reken Methode - 1

"*Standard Reken Methode - 1*" is derived from the CAR II model. The CAR model was the first air quality calculation method developed in the Netherlands in the 1980s. CAR II, the improved version of this model, was implemented in 2002. CAR II introduced high resolution background concentration values from GCN (Generic Concentrations in the Netherlands) in the model. In the following years, further improvements, such as the calibration factor and high resolution wind speeds, were implemented [30].

SRM-1 is used nowadays to compute air quality in built-up areas, which are characterised by the proximity and density of buildings. The morphology of the city prevents air from flowing, determining the creation of "street canyons" with an upraised concentration of pollutants. SRM-1 classifies urban streets in four categories, based on the heights of nearby buildings and their distance to the street axis. The latter must be maximum 60 meters; building facades should not be farther than 15 meters and their heights should be homogeneous [30]. Figure 2.3 provides a visual representation of the different classes as specified in the SRM-1 model.

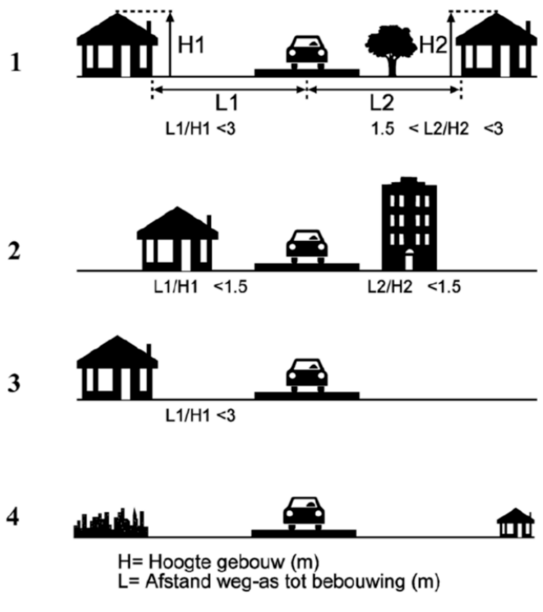


Figure 2.3: The different street types as specified by RIVM for the SRM-1 model. H = height of the building, L = distance from street axis to building facade. *Source*: "Technisce beschrijving van standaard rekenmethode 1 (SRM-1)" [30].

According to the technical standards of RIVM [30], streets are classified as follows:

Class 1 “more or less continuous buildings on both sides of the street at a distance of up to 60 meters from the street axis, where the distance between street axis and facade is less than three times the height of the buildings, but is larger than 1.5 times the height of the buildings”;

Class 2 “more or less continuous buildings on both sides of the street at a distance of a maximum of 60 meters from the street, where the distance between the street axis and facade is less than 1.5 times the height of the buildings”;

Class 3 “more or less continuous buildings on one side at a distance from a maximum of 60 meters from the street axis, where the distance between the street axis and facade is smaller than 3 times the height of the buildings”;

Class 4 “all streets in an urban environment, other than street types 1, 2 and 3”.

RIVM also has some other requirements for the street segments [29], for which the scanned document is attached in Appendix A. It states the following:

1. “the length of the street should be at least 100 meters”;
2. “an object is seen as a building if it has a minimum height of 3 meters”;
3. “a street should possibly get a different street type if the sum of the gaps between buildings along the street is more than 15 meters”.

3
Datasets and preparation

This chapter will introduce the study area, including a smaller focus area used for testing purposes. The software required for the project is also discussed, followed by the pre-processing steps for the datasets.

3.1. Study area

The focus of this research will be mainly on the city of Amsterdam, since the project is in collaboration with the Municipality of Amsterdam. The city lies in the province of North-Holland and is the capital of the Netherlands. It provides an interesting study area, not only because of its size, but also because of the city layout with its many narrow streets and canals. Being a big city has the advantage that a lot of air quality information is collected in the area, which is useful as reference data for the project outputs. Figure 3.1 shows the entire study area for the city of Amsterdam as taken into account in this research. There will also be focus on a smaller area around the Weesperstraat. The Weesperstraat area is interesting in particular, because it contains all four different street types.

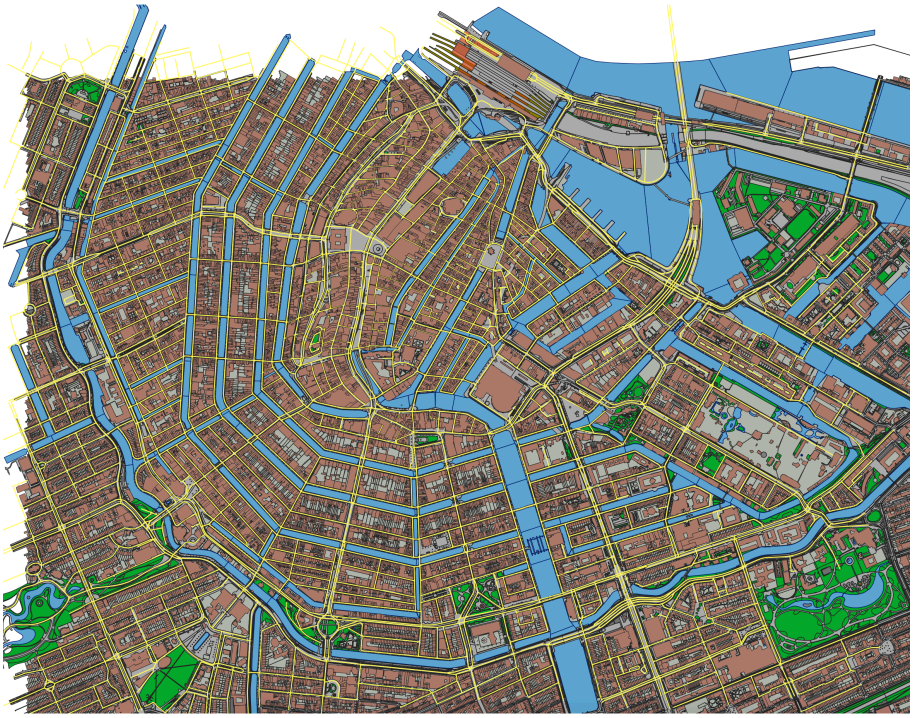


Figure 3.1: The study area for the city of Amsterdam.

3.2. Software specifications

The implementation of the project makes use of multiple tools for analysis, computation and visualisation. These tools include QGIS, Python and 3dfier.

More specifically, *3dfier* is a software tool developed by the 3D Geo-Information group at the TU Delft [9]. It takes 2D GIS datasets and '3dfies' them by extruding the polygons to 3D. It includes data preparation tools, and especially the script to prepare BGT datasets is of interest for this project.

Python is an open source programming language that allows the installation of multiple third party packages [12]. This project requires the installation of `Numpy`, `Shapely`, `Scipy`, `Rtree`, `Pyhull` and `Fiona`. Fiona is used to read the geo-spatial data files, and Shapely is mainly used for different spatial operations on the data.

Lastly, *QGIS* is a free and open source Geographic Information System (GIS) [24]. It can be used to create, edit, visualise and analyse geo-spatial information on different operating systems. Through Open Geospatial Consortium (OGC) web services data can be (down)loaded, which proved useful for background maps and retrieving national datasets.

3.3. Data

The project requires two types of datasets: data used as input for the street classification algorithm, and data used as a reference for the air quality measurements and street classification results.

3.3.1. Algorithm input data

The classification algorithm requires building footprints, building heights, and road network data. The building footprints are available in the '*Basisregistratie Grootschalige Topografie*' (BGT), which can be downloaded through '*Publieke Dienstverlening op de Kaart*' (PDOK) [23]. Four tiles are downloaded, as shown in red in Figure 3.2. These tiles cover most of the city of Amsterdam and all of our air quality measurement locations. The measurement locations will be discussed into more detail in Chapter 5.

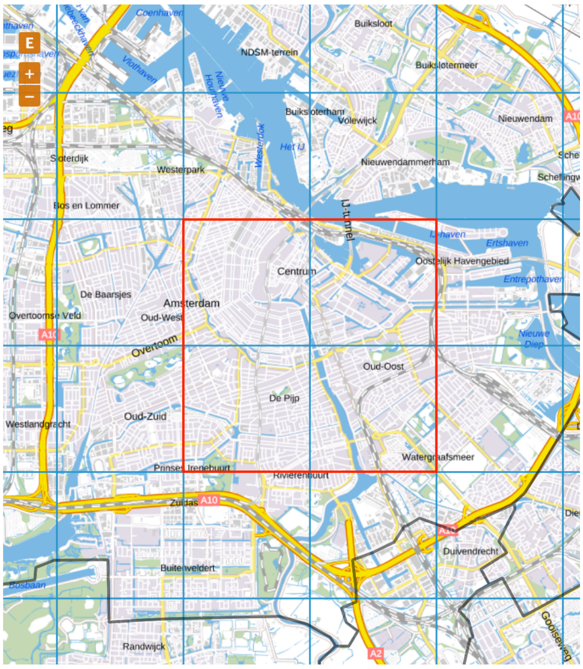


Figure 3.2: The four BGT tiles that were downloaded of the city of Amsterdam from the PDOK website, highlighted in red.

The building height data is available in the *3D 'Basisregistratie Adressen en Gebouwen'* (BAG) dataset, which is generated by the 3D research group at TU Delft [10]. This dataset contains roof heights at different percentiles (25, 50, 75, 90, 95, and 99). The current, up-to-data version '*Actueel Hoogtebestand Nederland 3*' (AHN 3) is used to obtain the height values. In the case that AHN 3 is not yet available, AHN 2 (the previous AHN version) is used to assign height values. For the city of Amsterdam AHN3 is available, but we will also test our algorithm on other cities and smaller towns in different parts of the country which makes this important to mention.

The BAG building outlines could also be used as ‘footprints’. The BAG represents the buildings as if you were looking at them from above. However, it also contains many underground structures that often cover the street geometries. The same could happen if buildings have overhangs. This can pose problems with the distance calculation from building facades to the street axis. Therefore, it is decided to use the BGT building footprints as input data for the classification algorithm.

Lastly, a road network dataset is needed. *‘Nationaal Wegen Bestand’* (NWB), available through the web services of PDOK, provides this data. The streets in this dataset are represented as lines, which is different from the BGT, where streets are represented as polygons. This is a required feature, since the distance from the building facade to the medial axis of the street is necessary for the classification algorithm.

Figure 3.3 shows the BGT building data and the NWB street data that is input into the algorithm, and Figure 3.4 shows the smaller study area around the Weesperstraat used for testing different parameter combinations.

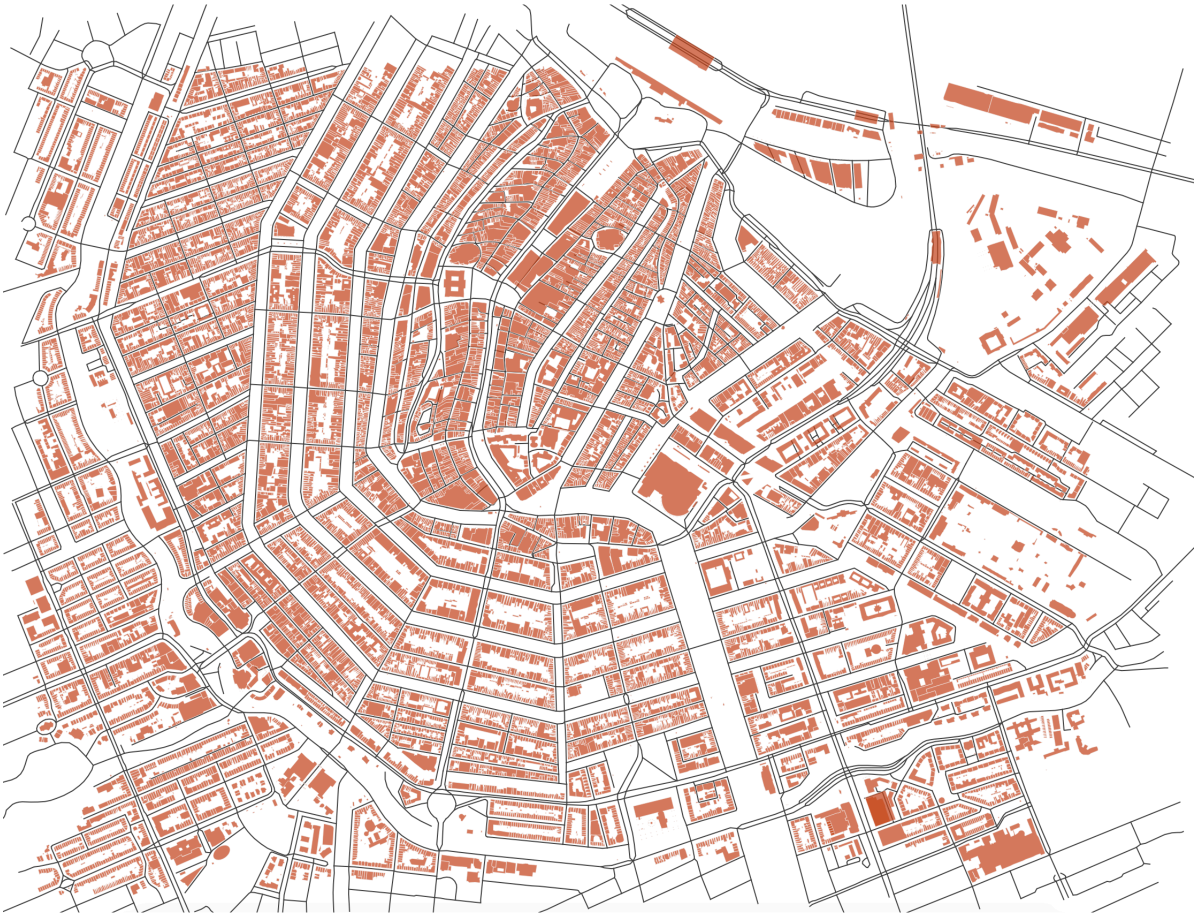


Figure 3.3: The BGT building data and NWB street data as it will be input into the street classification algorithm for the city of Amsterdam.

3.3.2. Reference data

The output of the classification algorithm must be verified. ‘*Nationaal Samenwerkingsprogramma Luchtkwaliteit*’ (NSL) is established to improve the air quality in the Netherlands [28], and the *NSL-Monitoring tool* provides the user with data from RIVM on air quality and the different street types [27]. Through the web interface, users can browse the data of different jurisdictions. When the data is

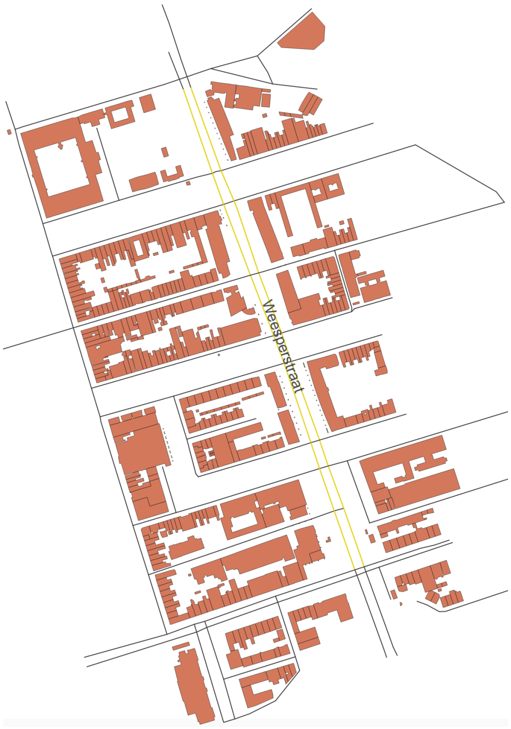


Figure 3.4: The BGT buildings and NWB street data for the smaller study area around Weesperstraat. Weesperstraat is shown in yellow.

downloaded, it contains receptor points, or calculation points, that have a street type assigned to them. Comparing the automated classification tool to these manual classifications gives a good estimation of the performance of the algorithm. However, we will never be able to verify the algorithm with a 100% certainty, because the provided manual classification relies on the interpretation of different persons, making it prone to human errors.

Besides the reference data for the street classification, there is also reference data needed for the air quality measurements. Gemeente Amsterdam has multiple reference stations in the city of Amsterdam, which can be accessed through the website of '*Luchtmeetnet*' [31]. The reference station of our interest is `NL49014` in the Vondelpark. Figure 3.5 shows the Vondelpark reference station and two other nearby reference stations on the map, together with our five air quality measurement locations. The reference data can be used to perform comparisons to our own measurement data. Especially the Vondelpark reference station is of interest to us, since it is used to determine the background pollution in Amsterdam. Being stationed in the park, the traffic emissions are generally reduced. Therefore, the Vondelpark measurements can be used to get an understanding of the contribution of traffic to the air pollution in other zones in the city; the Vondelpark measurements are subtracted from the local measurements.

In our research, Vondelpark is chosen as a measurement location for the following purposes;

1. To obtain a background pollution measurement for the sensors, and;
2. To compare the sensor measurements to the reference measurements to get an idea of how well the sensors perform when there is a lack of additional pollution sources such as traffic.

3.4. Pre-processing

The datasets require some pre-processing before they can be input into the street classification algorithm. The 3D BAG and NWB datasets both cover the whole of the Netherlands. They are clipped to the extent of the corresponding BGT tiles to cover the same area. This is a relatively easy step to perform. The following sections will describe the more complicated pre-processing steps.

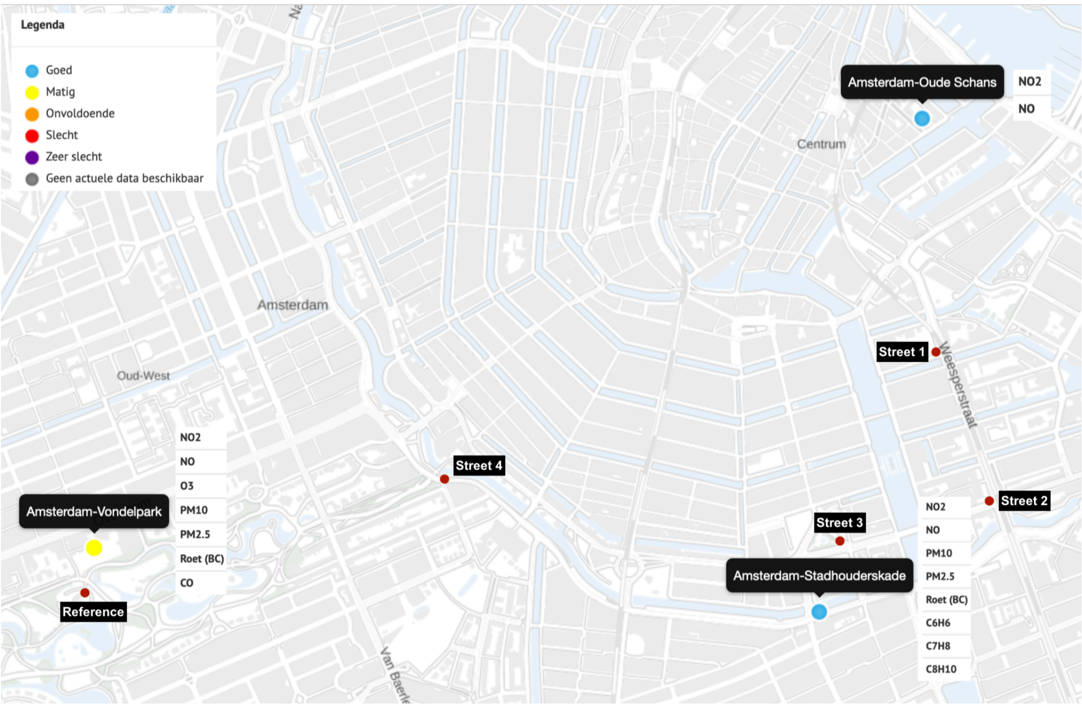


Figure 3.5: The three reference stations used from Luchtmeetnet. For each reference stations it is stated what is measures. The red dots represent our five measurement locations.

3.4.1. Preparing BGT data

The BGT data downloaded from PDOK contains the geometry type `CurvePolygon`, which is not supported by software such as QGIS. Therefore, the type should be converted to the regular `Polygon` geometry type before further processing steps can be applied. The software tool `3dfier` contains the 'BGT_Prepare' script to perform this conversion, and it also filters the history of objects based on the '`eindregistratie`' (end registration date) attribute [9]. If this attribute is not set, it means the object currently exists. It is desired to filter on this attribute, because the algorithm requires the most up-to-date representation of the street layout. The BGT conversion is stored in a `GeoPackage` file, which is used in the next pre-processing step.

3.4.2. Combining 3D BAG and BGT

The BGT building footprints do not contain any height data. Therefore, the BGT must be enriched with height data from the 3D BAG dataset. The BGT contains an attribute called '`identificatieBAGPND`', which is also available in the 3D BAG under the attribute name '`identificatie`'. QGIS is used to perform a join on these two attributes. However, the 3D BAG and BGT do not use the same data types for their attribute fields. The BGT uses an `integer64` field, while the 3D BAG uses a `string` field. This results in slightly different IDs; the 3D BAG IDs start with a zero depending on the ID length. The 3D BAG attribute field is therefore first converted to `integer64`, and then the datasets can be joined. The result is a dataset with the BGT footprints, and the 3D BAG height data for the different roof height percentiles.

3.4.3. Street simplification

It was also attempted to simplify the NWB street network data. The dataset contains many parallel streets, and merging them into a single lane would reduce computation time because less streets have to be classified. The first step in the simplification process is to identify parallel streets. This makes use of two assumptions; streets are parallel:

1. when their endpoints do not touch. In other words, when they have a distance bigger than zero from each other; and
2. when their centroids are within a certain distance from each other.

However, this approach introduces some difficulties; streets do not always have the same length, resulting in their centroids being too far apart. The different street lengths also pose problems with snapping the end points of the street. Figure 3.6 shows a street network, including the centroids and which streets are detected as being parallel. The image makes it even more clear that it is hard to correctly detect and merge street segments.
First of all, streets are not always detected as being parallel, even though their distance from each other is similar to streets that are detected as being parallel. This is because their centroids are not directly opposite of each other. Using a fixed distance threshold to check for closeness is not robust.
Secondly, street segments can contain more than two points, which introduces corners in the street if the points are not collinear. In this case, the centroid of the street falls outside of the street segment. The street is then split into different parts, and for each part the centroid is calculated again. Extra computations are introduced, as more centroids and segments must be checked against the two rules that were described above. In the end, the smaller segments of the streets should still refer to the original street ID. It can now happen that the segments of one street get merged with multiple other streets. This introduces difficulties with which street ID to keep for further processing.

These challenges, that showed up by doing our experiments, made it clear that it is not easy to simplify the street network. Since the simplification does not affect the street classification in any way, we decided not to investigate the problem any further. The feasibility and gains for the project were not enough given the project time limit.

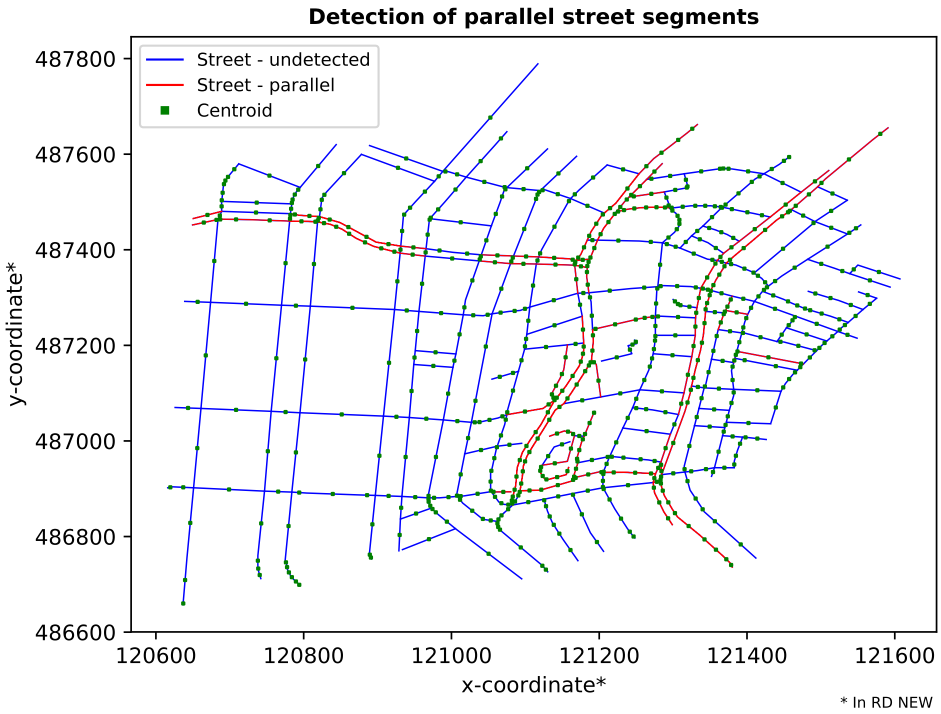


Figure 3.6: The street simplification visualised. It can be seen that the detected red lines are not always parallel or in need of merging. Other times, no parallel streets are detected where you would expect it.

4
Street classification

In this chapter, we explain into detail the core steps of our algorithm. Firstly, the association between streets and buildings is discussed. This includes a more in depth look at how buffers and the Voronoi diagram can be used to identify which buildings face a chosen street. Later, the four different street classification methods that we developed are explained. And lastly, some ways to speed up the algorithm are introduced, which improves the scalability of the algorithm. Figure 4.1 provides a high-level overview of the steps.

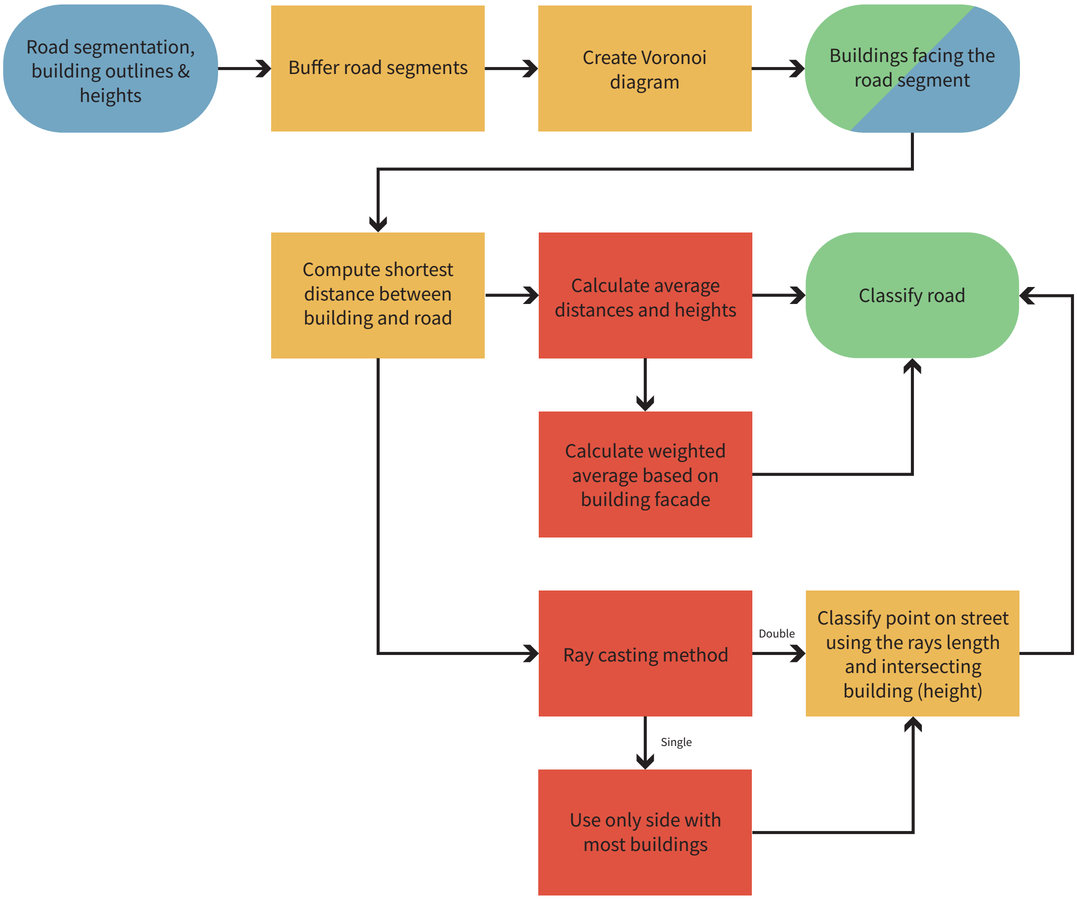

Road segmentation, building outlines & heights
Buffer road segments
Create Voronoi diagram
Buildings facing the road segment
Compute shortest distance between building and road
Calculate average distances and heights
Classify road
Calculate weighted average based on building facade
Ray casting method
Double
Classify point on street using the rays length and intersecting building (height)
Single
Use only side with most buildings

Figure 4.1: A flow-chart of the entire classification process.

4.1. 'Eerstelijns bebouwing'

The RIVM classification method requires knowledge on the buildings along the street segment. Associations between the street segment and the buildings must be made; the so called '*eerstelijns bebouwing*' (first line of buildings along a street) should be found. This detection is performed in several steps, which are discussed in the following sections. All different implementation methods that have been attempted, but did not make it in the final implementation, are attached in Appendix C.

4.1.1. Respecting the 60 meter rule

The technical description for the street type classification by RIVM states that the buildings should be at a maximum distance of 60 meters from the medial axis of the street segment [30]. Buildings that are further away are not considered. In order to satisfy this rule, a buffer implementation is used.

The `Shapely` library contains a buffer function which can be called on a geometry; a `LineString` in the case of a street. When the street has a more complex geometry (e.g. many sharp corners), the buffer can self-intersect. These intersections can lead to 'wrong' buffer shapes, or no buffer is created at all. This problem is solved by splitting the street into parts if the `LineString` is defined by more than two points. For every part of the street a buffer is created, and a separate classification is performed.

The buffers are created by using a 60 meter offset with respect to the street segment. The offset results in another line segment, which gets combined with the original street line segment to create a `Polygon`; the newly created buffer. The buffer is then used to detect buildings within 60 meters of the corresponding street; a building satisfies this rule if it intersects with the buffer. Figure 4.2 displays a street segment, its buffers and the buildings that intersect with the buffers. The next step in the process is to detect which buildings are facing the street.

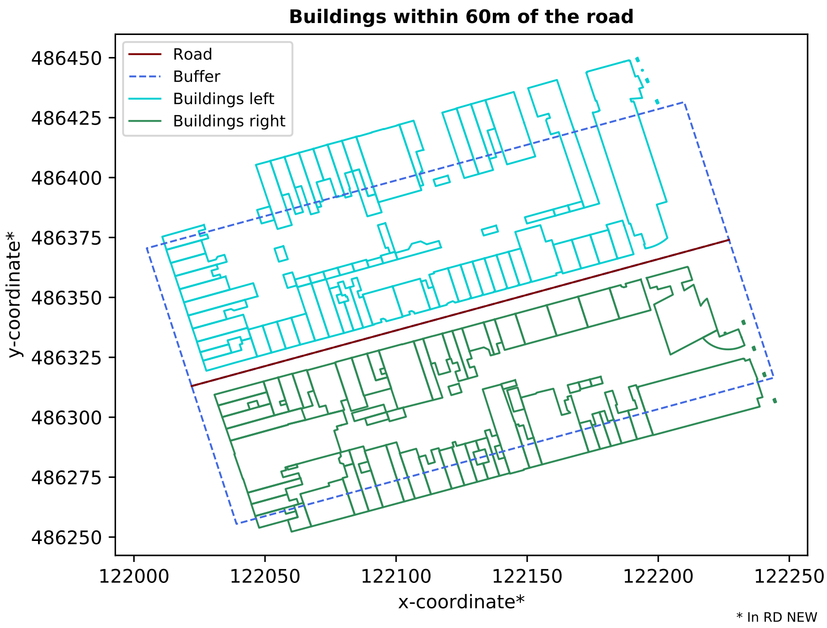


Figure 4.2: Overview of the buildings that fall inside of the 60 meter buffer (blue dashed line) around the street segment (red line). Buildings on the left side of the street are depicted in blue, and buildings on the right side in green.

4.1.2. Associate buildings to streets

Finding the relationship between streets and buildings proved to be difficult. We have studied several alternatives; the solution must be flexible, because urban environments have heterogeneous street layouts. A simple buffer at a fixed distance is not sufficient; the distance from the street to the building facade differs even within the same street segment. Another possible solution would be to discretise the street segment into points, and to cast rays perpendicular to the left and right of the street segment.

The first intersection with the ray provides the building facing the street. This approach has the limitation that the distance between the points on the street matters; buildings can get skipped otherwise.

A more robust solution to this problem is the use of a so-called *Voronoi diagram* (VD) [5]. It consists of convex cells, which are generated by data points; the generators. When n points are input, the VD consists of n cells. The Voronoi vertices are equidistant from three generators, and the Voronoi edges are equidistant from two generators. The adjacency of Voronoi cells, including the corresponding generators, can be used to detect which buildings are facing the street segment. Figure 4.3 shows an example VD with its different properties. Voronoi cells on the edge are infinitely big.

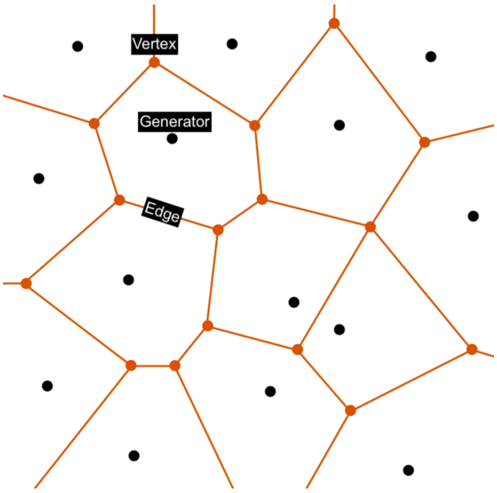


Figure 4.3: An example of a Voronoi diagram with the generators (data points), Voronoi vertices, and Voronoi edges.

For our application, the VD is created from the street and building data. It is required to first convert the geometries into a set of points, which can then be used as input for the `VoronoiTess` function of the `Pyhull` library. Figure 4.4 shows the VD for a street of the Amsterdam dataset. It should also be mentioned that a bounding box of the extent of all geometries in the scene is used to avoid infinite Voronoi cells for the street and building points at the border. This reduces the chance of wrong buildings to street classifications. Figure 4.5 illustrates what happens when no bounding box is included; a lot of buildings on the side get classified as facing the street.

The VD is used to find 'ridges', which are pairs of generators that are adjacent to each other through their Voronoi cells. Separate lists for buildings points and street points are used, and this provides a way to check if the points in the ridge pair belong to both the street and a building. If this is the case, we mark the building as facing the street. The checking process is repeated for all the ridge generator pairs. Once this step is done, we know which buildings are facing the street, and on which side of the street these buildings are located. Figure 4.6 shows which buildings are detected as facing the street (green), and which buildings are not (red).

The Voronoi method also has limitations, for which we have tried to find solutions. Firstly, in some rare cases no VD could be created. This is due to precision errors in `qhull` of the `Pyhull` library. The lower precision results in a non-convex shape, which is not allowed in a VD. For these street segments it cannot be detected which buildings are facing the street. Therefore, it is decided to give them class '`None`'. Chapter 6 will provide actual numbers on how often this problem occurs in the different datasets.

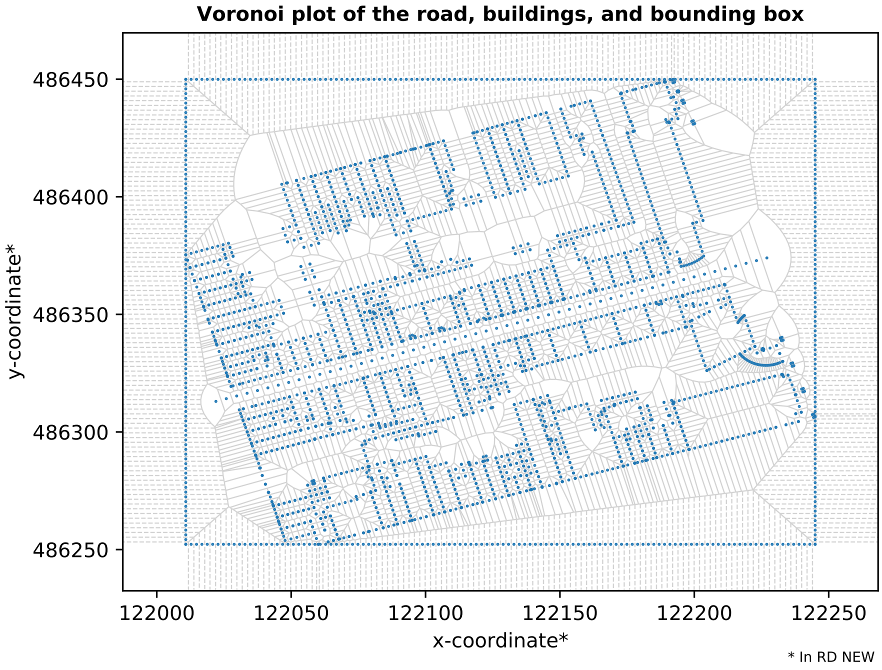


Figure 4.4: A Voronoi diagram created from points making up the street, the buildings and the bounding box around all features. The points represent the generators, and the Voronoi edges are shown in grey.

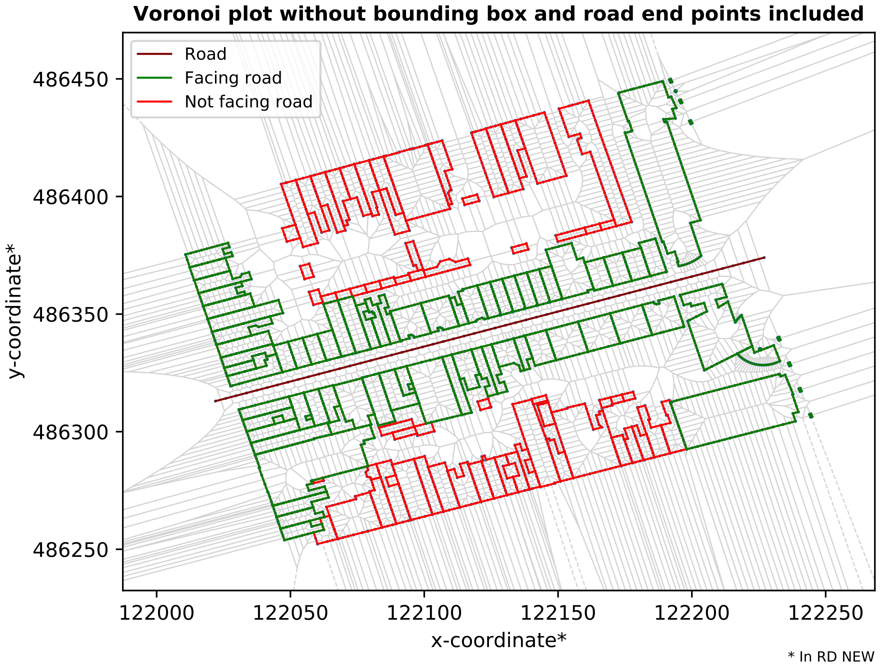


Figure 4.5: A Voronoi diagram created from points making up the street and the buildings. The points represent the generators, and the Voronoi edges are shown in grey. Buildings that do not face the street are detected as facing the street because of the infinite Voronoi cells.

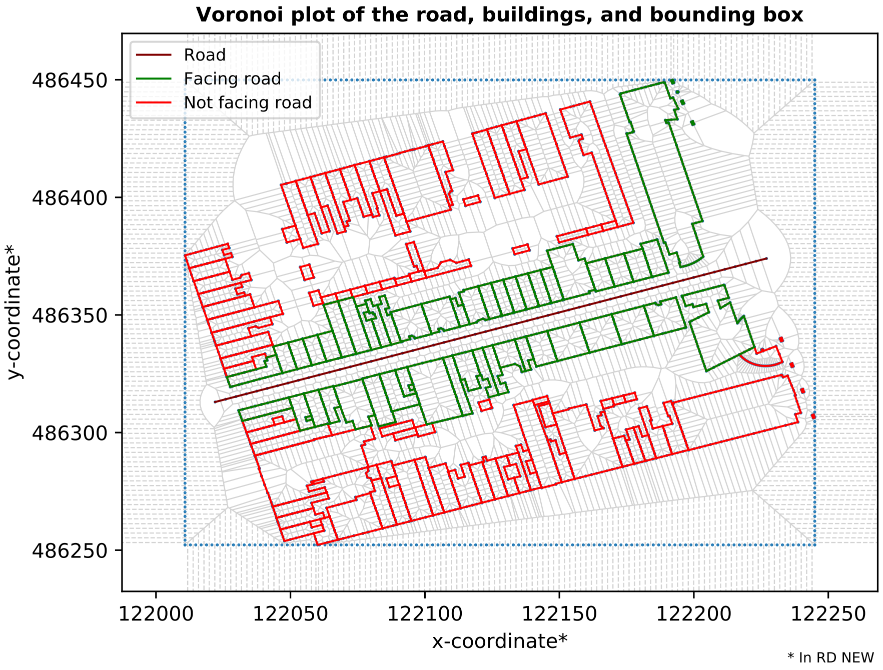


Figure 4.6: A Voronoi diagram where the green buildings have neighbouring cells to the street cells, and the red buildings do not. This allows to detect which buildings face the street. The street end points are excluded.

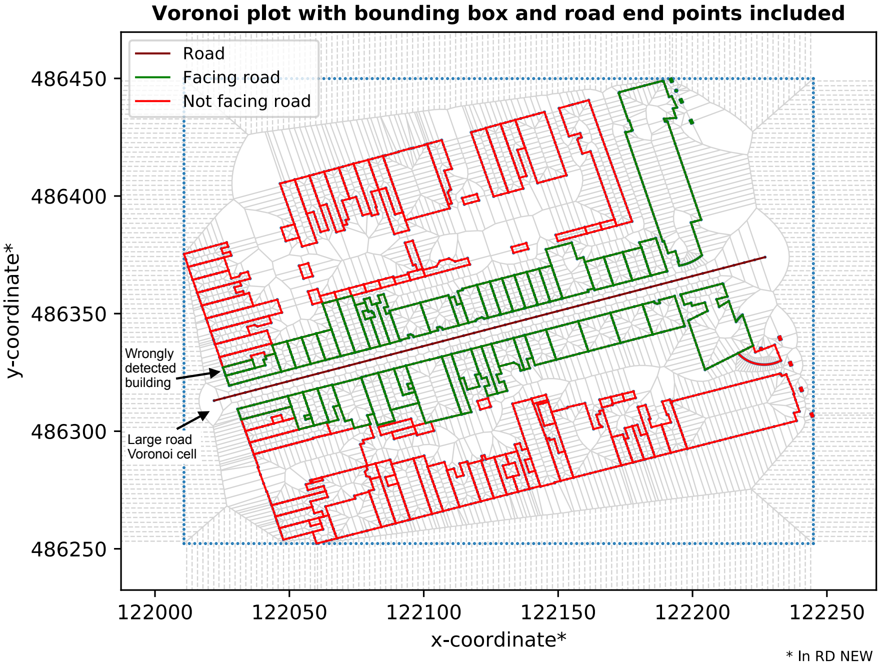


Figure 4.7: A Voronoi diagram where the green buildings have neighbouring cells to the street cells, and the red buildings do not. The street end points are not excluded, and we see that there is a building wrongly associated to the street.

Secondly, the bounding box limits the Voronoi cells generated by the street and building points around the edges, but there are still buildings wrongly associated to the street segment. This happens at the end points of the street segment, especially if the street Voronoi cell is large. It introduces a higher chance of the building Voronoi cells touching the street Voronoi cell. Therefore, it was decided to exclude the two end points of the street in the association checking process; all cells adjacent to these two cells are ignored. Figure 4.7 depicts this problem; the bounding box is present, but the street end points are not excluded in the process of finding buildings that face the street. The arrow points at the building that should not be associated to the street segment.

Lastly, it can happen that a Voronoi cell of a building 'blocks' the Voronoi cell of another building that is facing the street. This can be seen in Figure 4.6, where on the right side the red curved building is facing the street, but it is not detected. The Voronoi cells of the green building next to it are blocking the Voronoi cells of the red building. This is a characteristic of the VD that we have no influence on, but it must be kept in mind that this problem can occur in some cases.

4.2. Street classification

Here, we explain in detail the four classification methods implemented in our algorithm: the average method, weighted average method, single-sided ray casting method and double-sided ray casting method. These procedures are used to classify streets in accordance with RIVM's *'Standard Reken Methode - 1'*, see 2.3.1. Each method takes as input one street and the buildings associated to it. The average and weighted average methods output one classification value for each street; the single-sided ray casting and double-sided ray casting methods output a per-building classification, which is then aggregated into one street classification. The output of all classification methods is written to a `CSV` file containing the street ID, class and feature geometry. This can be used for further analysis or visualisation purposes.

4.2.1. Average method

In the average method, buildings facing the street on both sides of the street are considered; in Figure 4.8 these are represented by the green buildings in side 1 and 2. Firstly the shortest distance of each building to the corresponding street is computed. In a second step, distances and buildings' heights are summed up separately and divided by the total number of buildings, eleven in the case shown in Figure 4.8. Thus, the average distance and average height values are obtained. These are then used to classify the street segment.

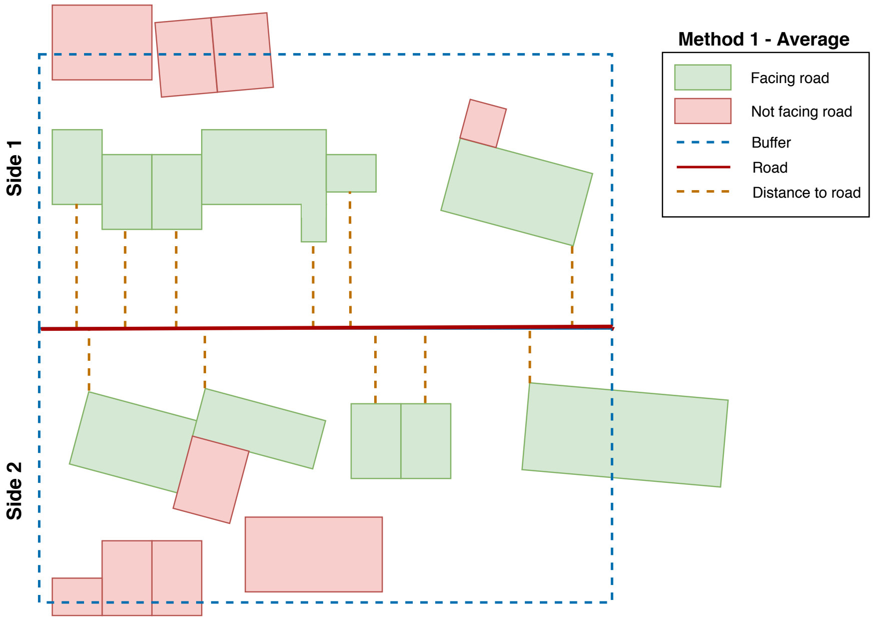


Figure 4.8: The averaging method, where for both sides the average distance and average building height is calculated. These are then used for the classification.

This procedure is straightforward and reliable for streets presenting continuous and parallel buildings; however, it could lead to uncertainties when buildings' heights, distances and facade lengths vary greatly in the same street segment. Buildings with a narrow facade count the same as buildings with a wide facade. If two extremes of buildings are present in the segments, averaging everything out can lead to a classification on the 'middle-ground', which is not necessarily preferred.

4.2.2. Weighted average method

Buildings on both sides of the street are considered in the weighted average method. In Figure 4.9 these are represented by the green buildings in side 1 and 2. While each building has a weight of one in the average method, in this procedure, we assign a weight to each building. The weight is applied to its height and distance to the street. The weight is an integer value, determined during the association of the building to street procedure, as described in 4.2.1. It corresponds to the number of Voronoi cells belonging to the same building that "touch" the street Voronoi cells. Figure 4.10 shows the building Voronoi cells (in green) that are adjacent to the street Voronoi cells (in red).

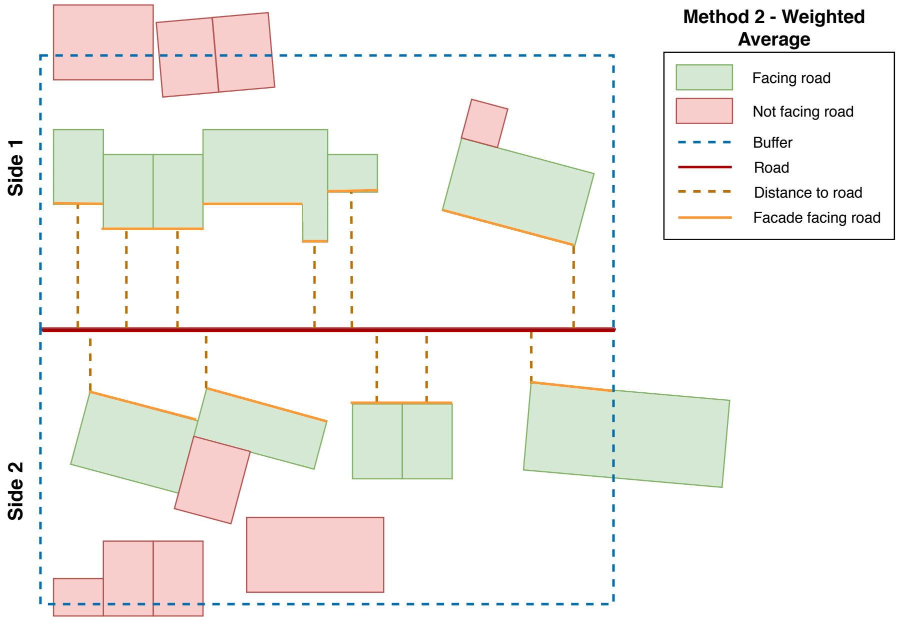


Figure 4.9: The weighted average method, where for both sides the weighted average distance and building height is calculated. These are then used for the classification.

The weights are then normalised; each one is divided by the total number of building cells facing the same street segment. After that, the shortest distance of each building to the corresponding street is computed. Later, the weight is applied to the distances and building heights, which are then summed and divided by the number of buildings. Thus, the weighted averages for the distance and height values are obtained. These are then used to classify the street segment.

This procedure introduces an improvement in the accuracy of the classification if buildings are continuous and of the type `Polygons`. Multi-parts, i.e. when a building is a `MultiPolygon`, should be minimised. Weights based on the Voronoi diagram could cause errors if discontinuities occur. For example, some of the buildings in the BGT dataset present columns facing the street because of the footprint representation of the BGT buildings. When these buildings are converted into Voronoi cell generators, see Figure 4.5, points representing the polygon would be sampled at different distances. In the main structure of the building, points would be created at a fixed distance based on the building's length, for example 1 meter. However, supposing that columns are two meters apart from each other, one point for every column would be created. Consequently, bigger cells would originate where columns are present, thus the building weight might be biased. Larger cells lead to less cells touching the street cells, and the weight of the building facade will be compromised.

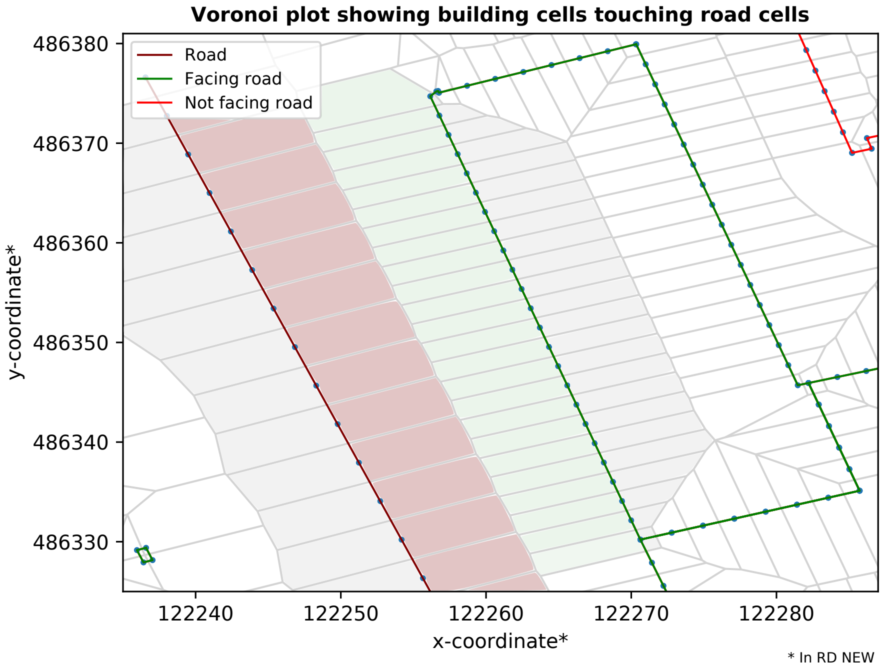


Figure 4.10: A Voronoi diagram where the green buildings have neighbouring cells to the street cells, and the red buildings do not. The green fill shows the part of the building Voronoi cell touching the street Voronoi cell. The red fill shows the part of the street Voronoi cell touching the building Voronoi cell.

Also, a larger building construction might be on top of these smaller structures, which is not taken into account. Figure 4.11 provides a visual representation of this problem.

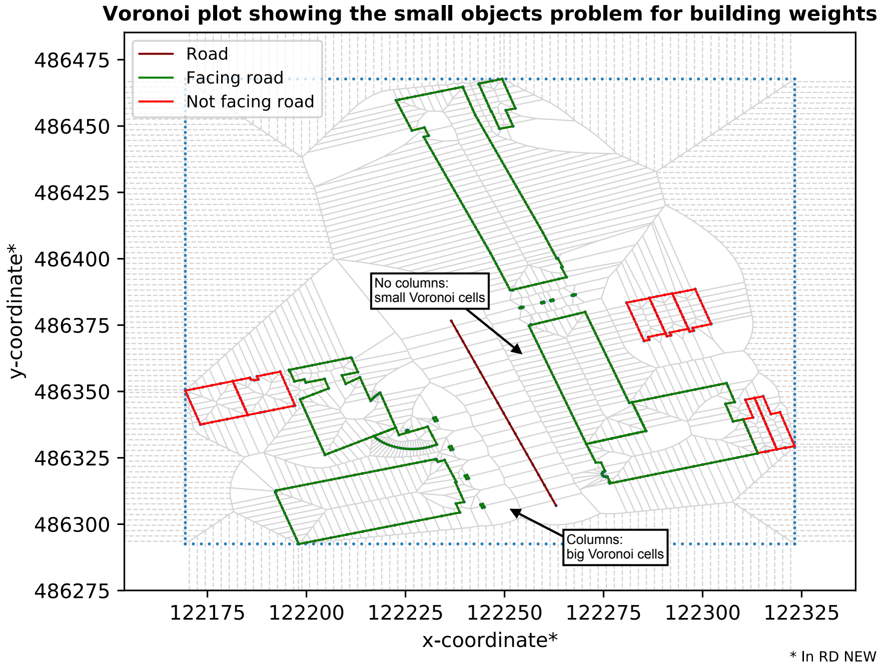


Figure 4.11: A Voronoi diagram where the green buildings have neighbouring cells to the street cells, and the red buildings do not. The columns of the building on the left result in bigger Voronoi cells, affecting the results of the weighting average method.

4.2.3. Single-sided ray casting method

In the single-sided ray casting method, the side of the street that has most buildings is selected; side 1 in Figure 4.12.

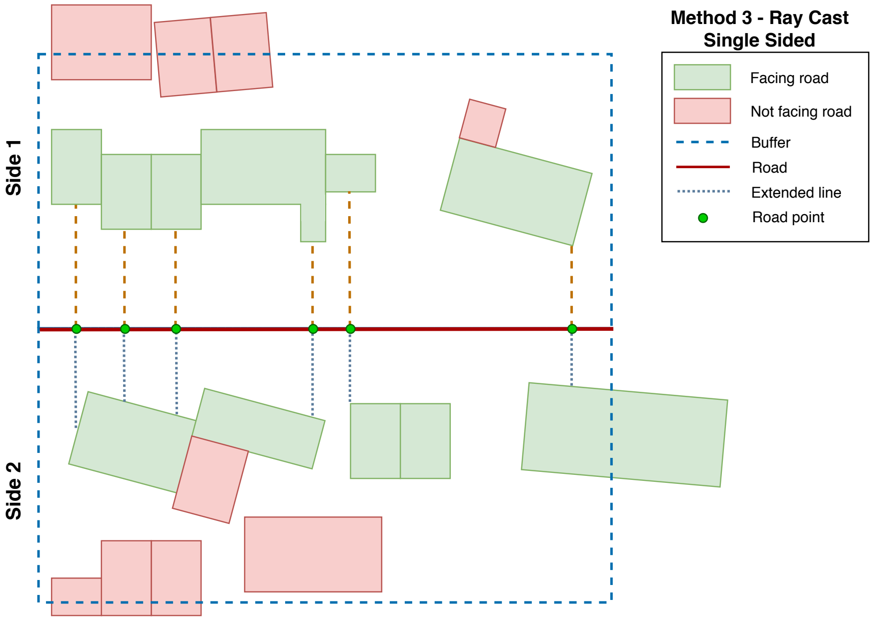


Figure 4.12: The single-sided ray casting method, where the side with the most buildings is used to cast a ray to the other side. The point on the building nearest to the street is used. The street points get classified, and a voting mechanism classifies the entire street.

For each building the nearest point to the street is computed, using the `shapely` `nearest_points` function. The function takes as inputs two geometries and returns two points, one for each geometry, which is the segment with the smallest distance between them. In this way, ideally we “draw” a ray from the building to the street segment, the orange lines in Figure 4.12. The two points are then used to compute a third point belonging to the same ray, on the other side of the street.

To do so, we compute the angle between the two points, using the formula:

$$angle = arctan2(x_{street} - x_{building}, y_{street} - y_{building})$$

Then, we calculate the third point, which is stored as a tuple:

$$extended\ point = (x_{street} + 61 * sin(angle), y_{street} + 61 * cos(angle))$$

`Extended point` corresponds to a point, laying on the same ray as the nearest points computed earlier. `Extended point` is at a maximum distance 61 meters from the point belonging to the street; the value was selected as it is slightly above the value indicated by RIVM (60 meters) as the maximum distance between a building and the street. At this point, we create a `LineString` geometry connecting the street point and extended point. This way, we look for buildings on the other side of the street, checking if the `LineString` intersects with any of the buildings on the side with fewer buildings. If an intersection occurs, we retrieve the corresponding building height and its distance from the street. Then, we classify the point on the street based on the RIVM SRM-1 specifications as discussed in 2.3.1. The same procedure is repeated for each building; the output corresponds to a number of classified points on the street, which are related to the number of buildings on the side of the street with most buildings. Lastly, we use these points to output one classification for the whole street segment, based on a voting system. This system counts the number of street type occurrences based on the classified points. The most dominant class is assigned to the entire street segment. In the case that different classes occur equally often, the class belonging to the first maximum value found in the list is assigned to the street. For example, classes 2 and 3 both occur five times, then the street segment is assigned class 2.

This procedure introduces advantages and disadvantages. As only one side of the street is considered to cast rays, the method allows for a fast implementation. However, some buildings on the other side of the street could be excluded from the classification procedure if they are not directly opposite of another building, or if they have a narrow facade. Furthermore, the side with most buildings does not always correspond with the most occluded side. In fact, the length of the buildings facing the street is not considered in this method. Moreover, the voting system could lead to an inaccurate classification. This may occur in streets where buildings' variety would lead to different punctual classifications. For example, in a street where the same number of points belong to different classes, the voting system would always assign the classification of the last point.

4.2.4. Double-sided ray casting method

The double-sided ray casting method follows the same procedure as the single-sided ray casting method described in 4.2.3. The latter selects the side of the street segment where most buildings are, while the former repeats the same procedure for both sides.

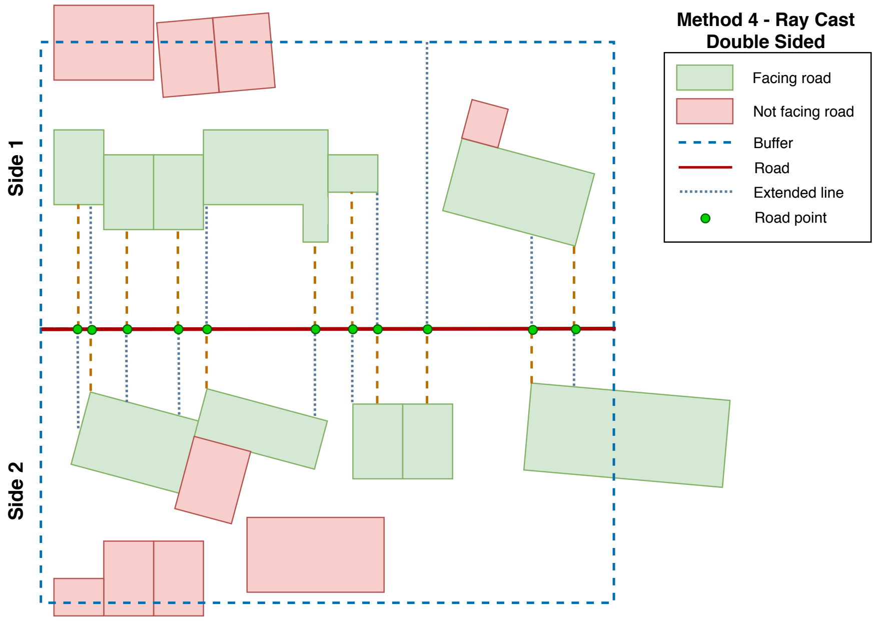


Figure 4.13: The double-sided ray casting method, where from both sides rays are cast to the other side. The point on the building nearest to the street is used. The street points get classified, and a voting mechanism classifies the entire street.

With the double-sided ray casting method, a higher computation effort is necessary; however, a more exhaustive representation of the street is made possible, since more rays are created. Thus, the street type is computed in more locations. This method could lead to building oversampling, for example, a long building could be taken in consideration more than once. This aspect is seen as a positive consequence, as it introduces a basic form of weighing. Figure 4.13 provides a visual representation of the double-sided ray casting method.

4.3. Speeding up the algorithm

In order to optimise the runtime of the algorithm, two different improvements are implemented. We will first discuss the use of an index for faster spatial queries, and then explain a line simplification method to reduce geometry complexity.

4.3.1. Indexing building geometries

The steps described in the previous sections are performed for every street segment. This can get resource intensive when the datasets are large. Especially the step of intersecting the buffer with the buildings takes up a lot of computation time; the intersection with the buffer is performed on every building in the dataset.

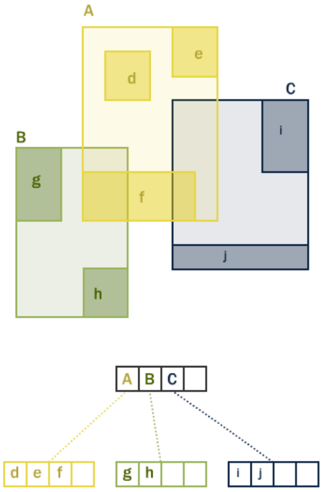


Figure 4.14: A simple example for a 2D R-tree. *Source:* `PostGIS.net`

To improve the intersection step, an *R-tree* is implemented on the building geometries using the `rtree` library in Python. It groups nearby objects based on their minimum bounding rectangle (MBR). A tree structure is created, and when going down in the tree structure each node will contain less objects than the level above it. At the leaf level, the rectangles describe a single object. Figure 4.14 provides a visual representation of what this would look like.

The intersection query is then performed on the buffer and the created R-tree index, in order to provide an initial filtering step. Only the MBRs that intersect with the buffer are returned, which limits the amount of buildings that must be checked individually. The algorithm runs more than three times faster for the dataset of the city of Amsterdam with the use of an R-tree, which allows better scalability of the algorithm. Chapter 6 contains results on how much time the algorithm needs to process datasets of different sizes.

4.3.2. Simplifying line segments

A single street can consist of many coordinates, resulting in possibly small street parts. We want the street parts to be as long as possible, while also preserving the topology of the streets. These longer street segments are necessary for the RIVM air quality calculations model (SRM-1); street segments are preferred to have a minimum length of 100 meters [29]. It must be said that the simplified streets are not guaranteed to be longer than 100 meters.

The line segments are simplified using the `simplify` function from `Shapely`, which makes use of the *Douglas-Peucker* algorithm. Douglas-Peucker tries to find a line similar to the original line, only with fewer points. The points in the new line are a subset of the points of the original line. A tolerance is specified, and the simplified geometry will contain only points within this tolerance. We set the tolerance to five meters. The function respects the topology of the original line segment.

The simplified streets are not only beneficial because of the longer street parts, but also because fewer segments must be processed. This reduces computation time, which will become more noticeable when the datasets are larger. Chapter 6 will provide statistics on the distribution of street lengths for the simplified and non-simplified streets.

5 Air quality measurements

This chapter covers details about the air quality measurements, which are performed in parallel to the design of the automatic road classification algorithm. First, we describe the sensors we use in this project, including some of its features. After that, we explain the purpose of these measurements and how they were performed. This includes the measurement locations, the measurement days and the method used.

5.1. Plume Labs and the Flow sensor

The measurements are performed with the *Plume Labs Flow* sensors, which is shown in Figure 5.1. Plume Labs provides a mobile application which displays the current pollution values in an orderly manner. The application is linked to a sensor, and the phone then connects to the sensor via Bluetooth. The GPS of the phone is used to track where the measurements are taken. Figure 5.2 shows what the app looks like.

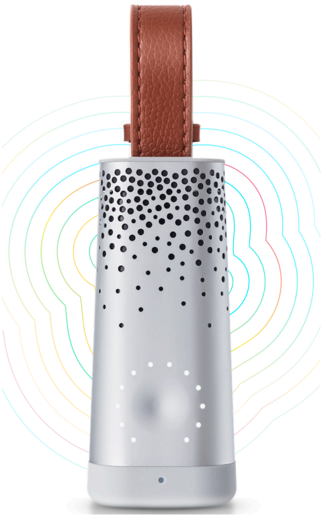
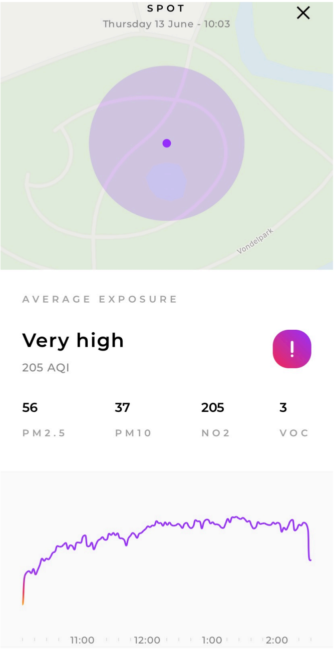


Figure 5.1: The Flow sensor. *Source:* Plume Labs [15].



SPOT

Thursday 13 June - 10:03

AVERAGE EXPOSURE

Very high

205 AQI

56	37	205	3
PM2.5	PM10	NO2	VOC

Figure 5.2: The mobile application showing a daily overview.

From the application we can see the different pollutants that are measured by the sensor: $PM_{2.5}$, PM_{10}, NO_2 and Volatile Organic Compound (VOC). The first three pollutants are already explained in Chapter 2. VOC are molecules which consist mainly of carbon and they move around as gasses in the air

[16]. Since they are very volatile, they can be found far away from where the original emission took place. Sources emitting VOC include traffic and the residential sector among others.

In order to retrieve the measurement data from the sensors, an e-mail should be send to Plume Labs. They then send the data back in `CSV` format. The file contains raw data for NO_2 and VOC in parts per billion (ppb), and for $PM_{2.5}$ and PM_{10} in $\mu g/m^3$. *Plume Labs* also includes data for their own established index [18], but we will not make use of it in this research.

5.2. Measurement campaign

The measurements are performed at five different locations, which are shown in Figure 5.3. Table 5.1 provides the geographical coordinates of the locations. These coordinates are used to ensure that we measured at the same location every time. Four of the locations were chosen to reflect the four different street types of the RIVM method, while the fifth location is really close to a reference station of the Gemeente Amsterdam. Since this location is in the park, it is supposed to give a better sense of the background concentration.

The aim of performing the measurements is to get an idea of how well the sensors perform. In addition, to the best of our knowledge, these sensors have not been previously compared with reference stations in real urban environments. Their performance has been tested inside the lab [17], but we have not found any information regarding their performance outside the lab. Therefore, within this project, we focus on a qualitative assessment with the goal of identifying street types through the measured values (PM_{10}, $PM_{2.5}$ and NO_2). To ensure the reliability of the measurement data, we measure the same location with different sensor each time.

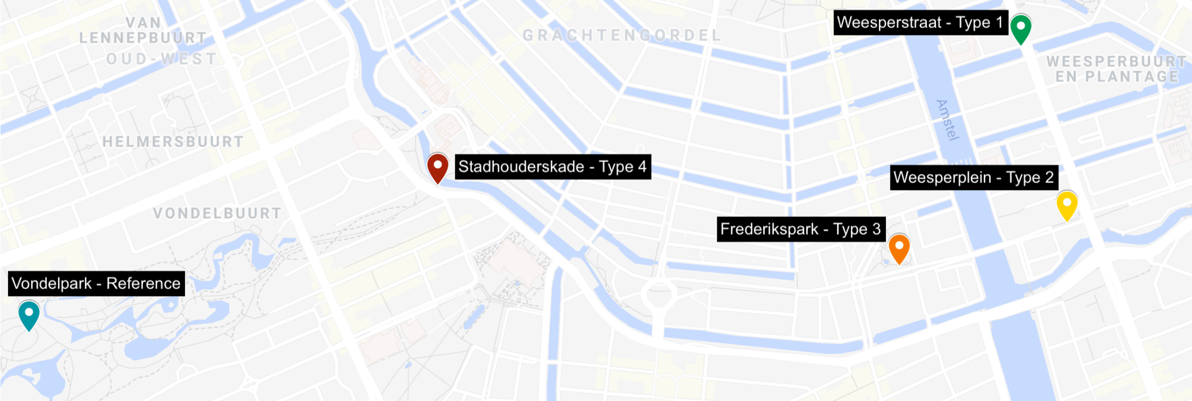


Figure 5.3: The five different measurement locations for the measuring campaign.

Location	Street Type	Coordinates	Comments
Weesperstraat	1	52°21’55.2”N 4°54’19.2”E	High & low buildings
Weesperplein	2	52°21’39.7”N 4°54’25.7”E	Similar height buildings
Frederiksplein	3	52°21’35.7”N 4°54’01.6”E	Park & buildings
Stadhouderskade	4	52°21’43.1”N 4°52’54.9”E	Canal & buildings far away
Vondelpark	-	52°21’30.0”N 4°51’56.5”E	Open space in park

Table 5.1: The five different measurement locations with their corresponding street type and coordinates.

5.2.1. Measurement dates and times

The measurements were performed in the same time frame on three different days: the 17th and 20th of May (Friday and Monday) and the 13th of June (Thursday) 2019. Arriving at the location at 10:00, we let the sensors stabilise for 30 minutes before the actual measurement. From 10:30 till 14:30, the sensor would measure the air quality at an interval of one minute. This interval is constant as long as the button on the sensor was pressed once every 15 minutes. The measurements are limited to days with no rain, since the small sensors are sensitive to humidity.

5.2.2. Measurement setup

The sensors were mounted on a tripod. The height of the tripods was set to 1.5 meters across all measurements, which is the pedestrian-level height used in the RIVM SRM-1 model. One GoPro Hero camera was available to monitor street activity, and was also attached to the tripod. The camera recorded a time-lapse picture every 60 seconds on the 17th of May, and every 10 seconds on the 13th of June. The GoPro took pictures for the entire measurement duration. Street types 1 and 4 were recorded with the GoPro. In addition, every 15 minutes notes were taken on temperature, cloud conditions, humidity, wind speed and traffic intensity. All this information is available in Appendix D.

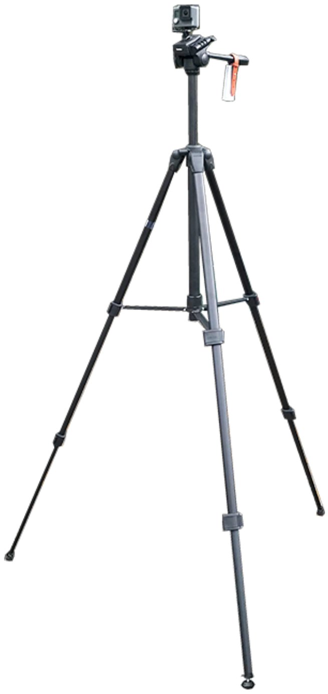


Figure 5.4: Setup of the PlumeLab Flow Sensor and the GoPro Hero camera mounted on the tripod.

6 Results and findings

This chapter presents the results of both the automatic street classification algorithm and the air quality measurements. It also presents observations and initial conclusions about the results.

6.1. Street classification

6.1.1. The different test datasets

The testing procedure for the street classification algorithm makes use of several datasets. First, the four different classification methods are run on the smaller dataset of the Weesperstraat area for two different height percentiles from the 3D BAG; the 95th and 99th percentile. We chose the area as the location of sensor measurements, which were described in Chapter 5, because it contains the four different street types; consequently we selected it as a reference for the classification algorithm. So, in total the algorithm is tested for eight scenarios; each classification method was combined with 95th and 99th height percentile. From these results, the ‘optimal’ solution is selected; i.e. the combination of a height percentile and a classification method (see Chapter 6.1.2 and Table 6.2).

The parameters are then used to perform tests on three different datasets for the city of Amsterdam; a small, medium and large dataset. The datasets ‘grow’ from the same region, which means that the medium and large dataset contain the area of the small dataset, and the large dataset contains the area of the medium dataset. The region they cover is the city centre of Amsterdam, and the larger the dataset, the bigger the area of the city centre that is covered. We selected this area, because of its dense urban environment. The three different datasets, including details such as the number of buildings and street segments, are described in Table 6.1.

Dataset	Filesize (MB)	#Buildings	#Street Segments
Amsterdam Small	6.17	6582	1176
Amsterdam Medium	10.85	12538	2337
Amsterdam Large	14.20	15890	3209

Table 6.1: The different datasets of the city of Amsterdam with their filesize, the number of buildings and the number of street segments.

6.1.2. The Weesperstraat area

The results of the average and weighted average methods for both height percentiles are shown in Figure 6.1, while Figure 6.3 shows the results for the single-sided ray cast and double-sided ray cast methods. The figures also show the receptor points as downloaded from the NSL-Monitoring tool. The classes of the receptor points are coloured in the same way as the classes of the street classification. The two top images contain the same classification method, but for different height percentiles. The same holds for the two bottom images. So, the upper left and lower left, and the upper right and lower right both display the same height percentile, but for a different classification method. As can be seen

from a visual inspection of the figures, the results are significantly similar, showing that the behaviour of different algorithms is similar as well.
Table 6.2 presents how many streets are classified correctly when comparing them to the receptor point classes. We assume that the manually classified streets as available from the NSL tool contain correct street types and can therefore be used as ground truth. However, in some cases we have doubts about the correctness of the manually classified streets; further research is required to confirm their quality. Furthermore, it must be noted that there is no direct link between the streets and receptors, and the numbers in the table are based on visual analysis only. We will now discuss the smaller details and differences between the four methods.

First, we analyse the outcomes of the average method and the weighted average method, shown in Figure 6.1. Here we compare differences between the two height percentiles; the horizontal axis in the figure. For both methods, differences can be seen only on the upper part of the study area. This zone presents a peculiar urban layout: a wide building on the east side of the street, and an open area on the west side. Figure 6.2 shows the 3D city layout of this area. With the 95th height percentile the methods assigned both streets type 4 (‘base type’); while with the 99th height percentile, both methods classified one part of the street as type 1 (wide street canyon) and the other as type 4. The 99th height percentile implies a bigger chance that the street will be classified as a street canyon. This is because same buildings most probably will have bigger heights. Table 6.2 shows that the different height percentiles barely affect the classification results; for both methods, only the aforementioned street is classified differently.

Method	Percentile	#Correct	#Incorrect	Visual Accuracy (%)
Average	95	23	7	76.7
Average	99	24	6	80
Weighted Average	95	24	6	80
Weighted Average	99	25	5	83.3
Single-Sided Ray Cast	95	23	7	76.7
Single-Sided Ray Cast	99	23	7	76.7
Double-Sided Ray Cast	95	24	6	80
Double-Sided Ray Cast	99	24	6	80

Table 6.2: The four different classification methods in combination with the two height percentiles. For each method the number of correctly and incorrectly classified streets are shown for the Weesperstraat area. These results are based on a visual inspection.

Now we compare dissimilarities in the results between the average and weighted average method; the vertical axis in the figure. Considering both methods with the corresponding height percentile, we can notice that the differences are consistent. Only a few street classifications are conflicting; in two cases, the average method classified one street as type 1, while the weighted average method classified it as type 2.

Furthermore, we analyse the differences with the NSL receptor points; these are in general consistent with our classification. The minimal differences appreciated between the two methods above, make the weighted average method more consistent with the classification in the NSL-Monitoring tool than the average method. Thus, we take this method in consideration for further comparison.

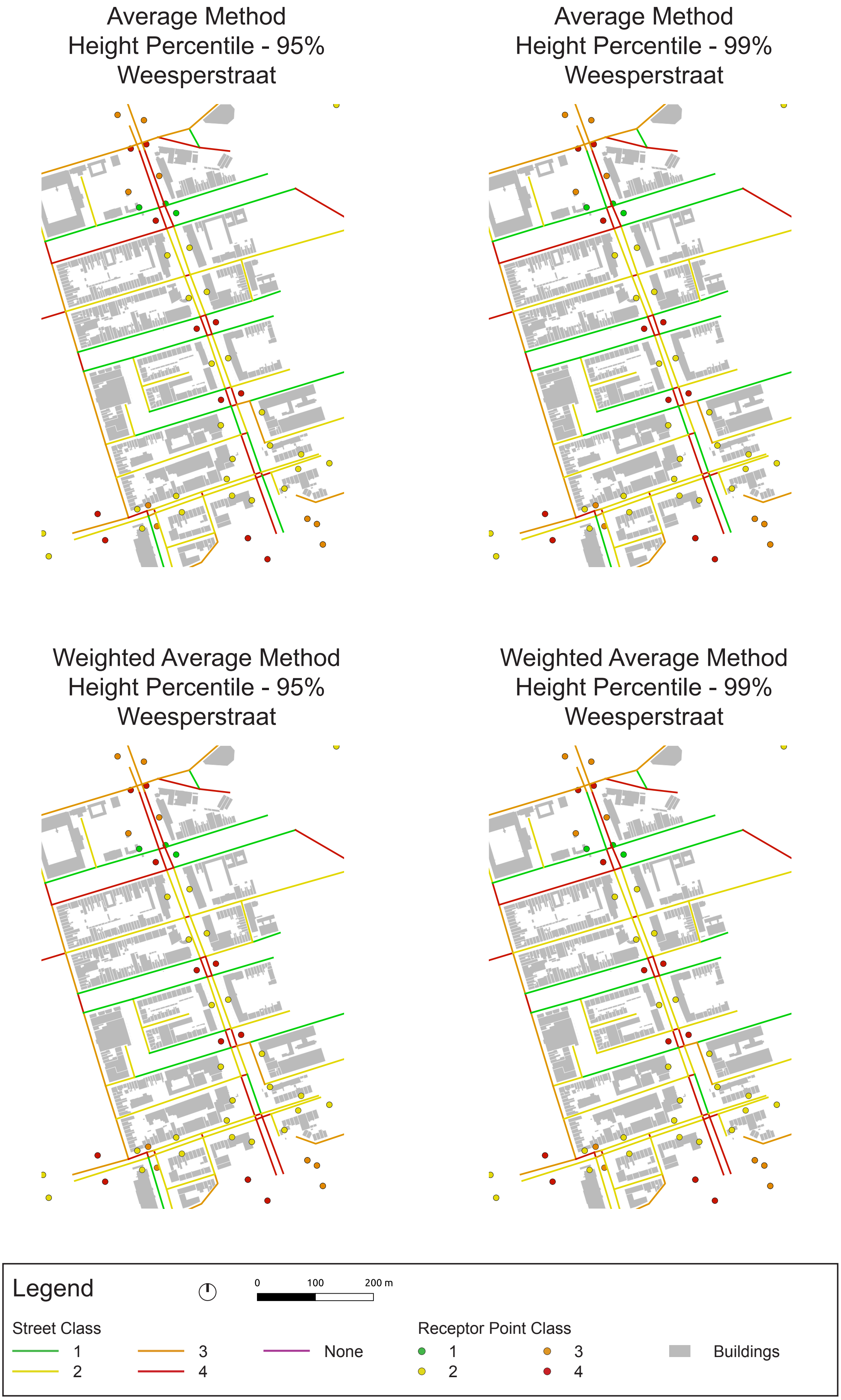


Figure 6.1: The average and weighted average method for both the 95th and 99th height percentile of the roof.

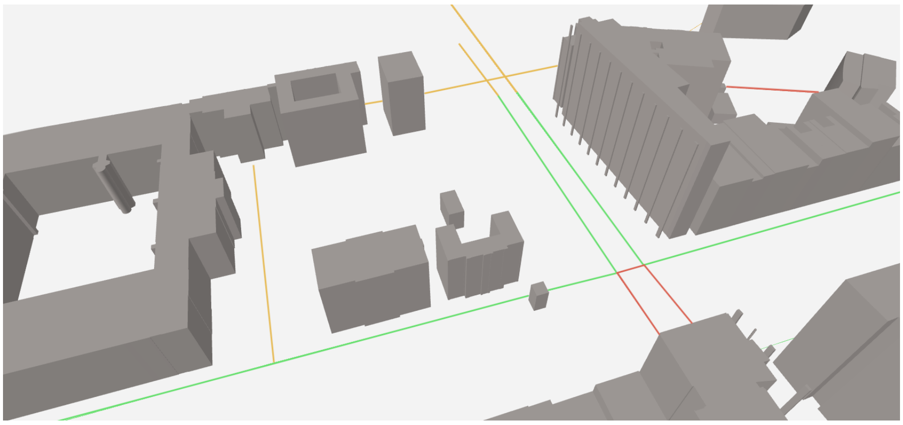


Figure 6.2: The north (upper) area of the Weesperstraat dataset. This displays the distinct urban layout with a big building on the east side, and a quite open area on the west side.

Second, we discuss the outcomes of the single-sided ray casting method and double-sided ray casting method shown in Figure 6.3. This reflection should be consistent with that of the two previously discussed methods; the average and weighted average method. When comparing the two different height percentiles for both methods, we see that they behave exactly the same. In contrast, there is one notable difference between the two methods in two streets of the upper area of the Weesperstraat. In the single-sided ray casting method both street segments have type 4 ('base type'), while having type 1 (wide street canyon) using the double-sided ray casting method. This could be due to the fact that the double-sided ray casting method takes more points into consideration for the final classification of the street segment.

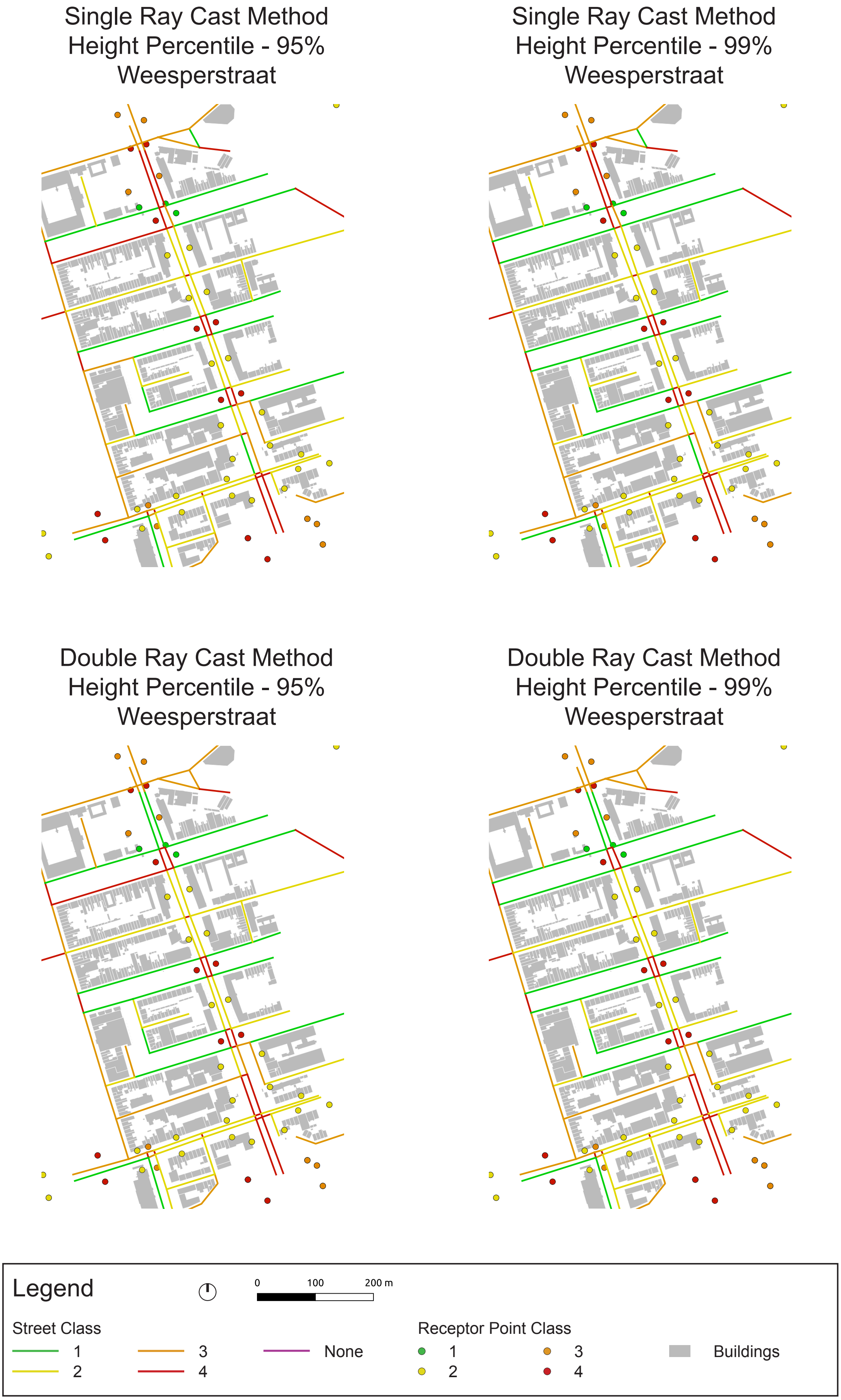
Once more, the comparison with the NSL receptor points, as shown in Table 6.2, highlights the similarity between the two methods and height percentiles. Given the urban layout and the visual comparison, the double-sided ray casting method is preferred over the single-sided ray casting method since it classifies one more street correctly. This method will be used for further analysis and comparison.

Optimal classification parameters

In this section we explain how we identified the optimal classification method and height percentile match based on the previously presented results. Comparing the weighted average method and the double-sided ray casting method, we believe that double-sided ray casting approach gives a more reliable solution for the street classification. The method takes every building on the street segment into consideration; moreover, it introduces a basic weighing in the classification. Large buildings are more likely to be considered multiple times, influencing the results. In contrast, the weighted average method takes into account the buildings' importance, but the methodology can be considered unreliable in some cases. As mentioned in 4.2.2, discontinuities in building geometries can lead to underestimation of their weights.

Furthermore, the visual comparison of the results shows that the double-sided ray casting is more conservative than the weighted average, since it implements the worst case scenario; the shortest distance from each building to the street is used. This increases the chance of a narrow street canyon classification (class 2). For air quality calculations, it is better to overestimate pollutant concentrations than to underestimate them. Table 6.2 shows that the weighted average method did classify 25 streets correctly, and the double-sided ray casting method 24. However, for the reasons stated above, the double-sided ray casting method is chosen as the most optimal solution.

The height percentile that is chosen as the optimal parameter is the 95th percentile. According to RIVM, the highest building point should be considered for air quality computations. However, the 99th percentile may bias the classification; for example, chimney outliers could be counted as the highest points. Thus, the combination of the double-sided ray casting method with the 95th height percentile is used for further processing and analysis. It must be said that this classification method deteriorates the



Single Ray Cast Method
Height Percentile - 95%
Weesperstraat

Single Ray Cast Method
Height Percentile - 99%
Weesperstraat

Double Ray Cast Method
Height Percentile - 95%
Weesperstraat

Double Ray Cast Method
Height Percentile - 99%
Weesperstraat

Figure 6.3: The single-sided and double-sided ray casting method for both the 95th and 99th height percentile of the roof.

data to some extent, since first points are classified individually and then converted, thus generalised, to a street-based classification.

These optimal parameters are applied to the city of Amsterdam and the other test areas because of the arguments stated above. The next sections will present and discuss these results.

6.1.3. City of Amsterdam

This section discusses the results for the dataset of the city of Amsterdam. The results are generated using the selected double-sided ray casting algorithm with the 95th height percentile. Figure 6.4 shows the street classification results.

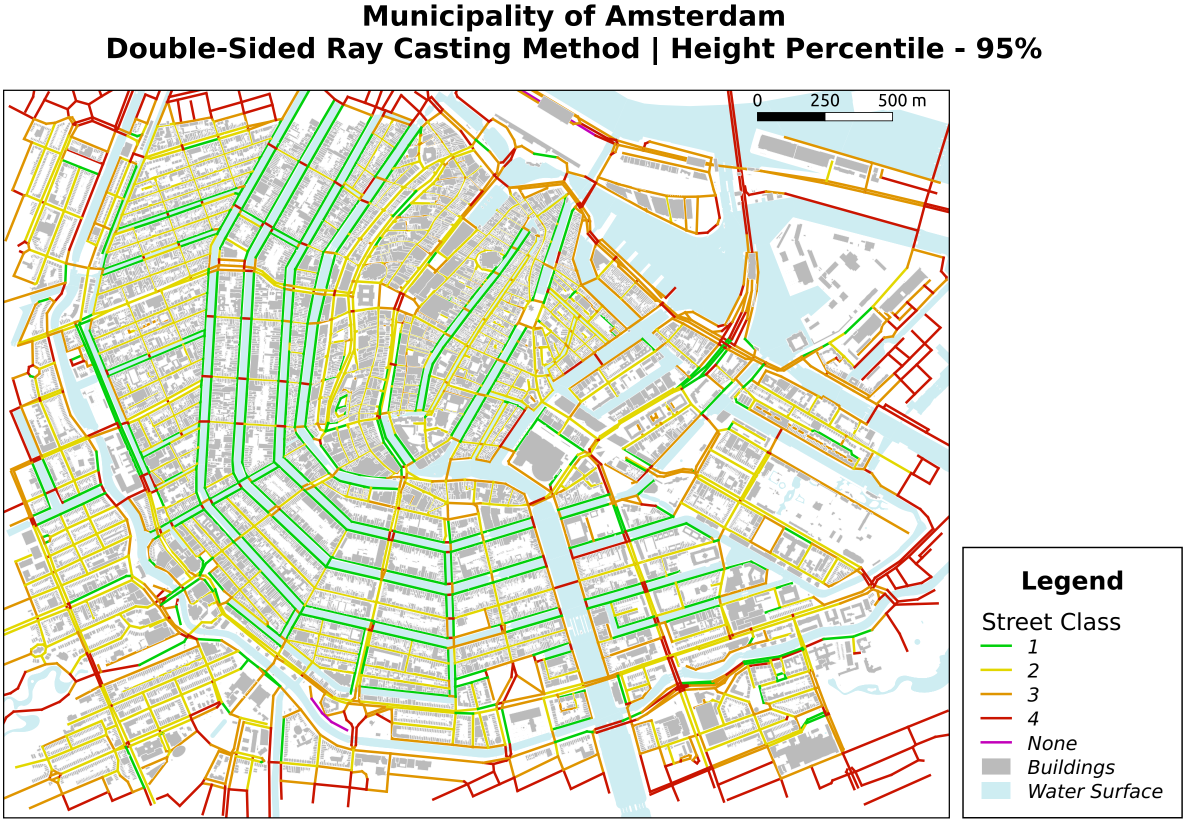

Municipality of Amsterdam
Double-Sided Ray Casting Method | Height Percentile - 95%
0
250
500 m
Legend
Street Class
1
2
3
4
None
Buildings
Water Surface

Figure 6.4: The result of the double-sided ray casting method with the 95th height percentile for the whole city of Amsterdam.

We see that along the more narrow canals, street type 1 is a common classification. This street type corresponds to the broad street canyon. The streets along the wider canals have either street type 3 or 4; the one-sided buildings and 'base type' respectively. Streets that are not adjacent to canals are often classified as type 2, the narrow street canyon. Especially in a city like Amsterdam, with its many narrow streets, this seems a sensible classification.

Table 6.3 shows the distribution of classes for the complete dataset of Amsterdam. Type 2 is the most common street type for the city centre of Amsterdam, followed by street type 3, 4 and 1. Table 6.4 shows the receptor point class distribution for the same area of Amsterdam. The numbers differ because there is no one to one relationship of streets and receptors. However, the receptor data also shows that street type 2 occurs the most in this area, followed by types 3, 4 and 1. This is the same as in the automatic street classification algorithm.

It can be noted that the percentages of class 1 and 4 significantly increases from Table 6.4 to Table 6.3. First and foremost, this could happen because the streets on the edges of the datasets do not necessarily have buildings surrounding them. This happens when the datasets are cropped; the buildings

Class	#Street Segments	%
1	488	15.21
2	1149	35.81
3	919	28.64
4	649	20.22
None	4	0.12

Table 6.3: The distribution of classes for the whole city of Amsterdam. Class None is assigned when no Voronoi diagram could be created.

Class	#Receptors	%
1	14	1.13
2	698	56.43
3	444	35.89
4	81	6.55
None	-	0

Table 6.4: The distribution of receptor points classes for the whole city of Amsterdam. Class None does not exist in the RIVM classification.

and streets are selected on ID for the same area, but the streets can have a long length resulting in these ‘empty’ streets. Another reason could be that the classification algorithm runs for every street in Amsterdam, including those facing the canals, bridges and non-urban roads; the former are mainly assigned to class 1, while the latter two are assigned to class 4. Instead, receptor points are only placed on the busier streets; the spatial distribution for the classes in both datasets is not taken into account in the tables.

The automatic classification algorithm contains an extra type, the `None` class. This type is assigned when no Voronoi diagram could be created, and thus no buildings facing the street can be detected. Only four street segments are assigned type `None`, which is only 0.12% of the street segments in this entire dataset. These streets could also be classified as the ‘base type’; class 4.

Dataset	Runtime (s)*	Runtime Optimised (s)*	Speed-up
Amsterdam Small	107.38	51.93	2.1x
Amsterdam Medium	308.76	102.68	3.0x
Amsterdam Large	465.67	135.22	3.4x

*The average of 6 measurements. Performed on *HP ZBook Studio G5 Mobile Workstation* laptop running on Linux Ubuntu 18.04.2 LTS, with the following specifications: Intel i7-8750H processor, Nvidia Quadro P1000 GPU, 16GB DDR4 2667MHZ RAM, 512GB SSD PCIe NVMe storage.

Table 6.5: The different datasets of Amsterdam with how long it takes to run them. The 95th percentile is used in combination with the double-sided ray casting method. Both the ‘original’ and spatially indexed runtimes are shown.

The automatic classification algorithm is also assessed in terms of runtime. Different dataset sizes are used, which are described in Table 6.1. The results of the timing experiments are shown in Table 6.5. It shows the runtime of both the non-optimised algorithm and the optimised algorithm. The latter makes uses of a spatial index, the R-tree. The entire process of reading the data, creating the buffers, finding buildings facing the street, classifying the streets, and writing the output to the `CSV` file is almost 3.5 times faster using the spatial index. The speed-up of the algorithm becomes more significant when the dataset size increases. This is logical, because the larger datasets contain more streets and buildings. For every street segment, all buildings in the dataset are checked twice to find the buildings within 60 meters of the street. This becomes computationally expensive very quickly, and thus the speed-up is most noticeable when the number of features in the dataset increases.

When we look at the runtime of the small and medium datasets, and compare it to the runtime of the large dataset, it can be said that the runtime doubles when the filesize doubles. This is for the specific datasets of Amsterdam, and it should be further investigated if this statement holds up for more cases.

Lastly, a comparison of the street length distribution in the original street segments and in the simplified street segments was carried out. The original streets consist of many more small parts than the simplified streets. Even though the the simplification changes the street geometry slightly, the larger street segments are preferred for the RIVM air pollution calculation model. Using the simplified street network with the automatic classification algorithm is recommended. Computation time will also be reduced by using this network. Table 6.6 shows the number of street segments in the original and non-simplified dataset for the city of Amsterdam.

Length (m)	#Non-Simplified Streets	#Simplified Streets
<20	2575	486
20-40	1204	691
40-60	683	625
60-80	439	481
80-100	249	261
>100	483	665

Table 6.6: The street length distribution for the whole city of Amsterdam for the original and simplified street segments. The length is measured for every street part that gets classified separately.

6.1.4. Other cities

The purpose of the automatic classification algorithm for RIVM is to make it easier to derive street types for the whole of the Netherlands. Therefore, the double-sided ray tracing method, with the 95th percentile, was used to classify the streets in other cities and towns in the Netherlands as well. Rotterdam, Delft and Groningen were chosen as other test cities, while Posterholt was chosen as a test area for a smaller village. The datasets and the results of the automatic classification algorithm are attached in Appendix B.

The smaller town of Posterholt does not have any NSL data available; only the information about the different street types provided by our algorithm exists. The cities of Rotterdam, Delft and Groningen do have information available in NSL. An initial visual inspection of the results, in comparison to the classes of the NSL receptor points, proved to be promising. However, further research is needed to evaluate the results into more depth, and to refine the algorithm for other areas in the country as well.

6.2. Air quality measurements

This section presents the results of the air quality measurements that were performed with the Plume Labs Flow sensors. Figure 5.3 shows where the five measurement locations are with respect to each other. The calibration of the sensors is discussed first, followed by an analysis of the measurement data from Amsterdam.

6.2.1. Calibration measurements

First of all, the sensors should be compared against each other. Differences in measuring behaviour among the five sensors should be identified in order to perform proper analysis of the measurement results. Figure 6.5 shows the sensor calibration for NO_2, PM_{10} and $PM_{2.5}$. The first thing that stands out is that sensor 4 did not measure during the calibration period.
When looking at the NO_2 concentrations, sensor 3 predicts the highest concentrations, while also being the most insensitive to changes in NO_2 concentrations compared to the other three sensors. Sensor 1 and 5 seem to be correlated, and sensor 2 shows similar tendencies to sensor 1 and 5, but it has a certain bias compared to the other two measures.

For PM_{10}, sensor 1 predicts the highest peak concentrations. However, the concentrations for sensor 1 also drop to zero very quickly compared to the other sensors, which is unexpected behaviour. The lowest peak concentrations of PM_{10} are predicted by sensor 5. Sensors 3 and 5 have some correlation between them regarding the peaks, where sensor 3 always measures higher concentrations than sensor 5.

Lastly, we will analyse the measured concentrations of $PM_{2.5}$. Again, sensor 1 predicts the highest peak concentrations, and similar to its trend in PM_{10}, it again drops to zero very quickly when compared to the other sensors. Sensor 5 measures the lowest peak concentrations for $PM_{2.5}$. As with the PM_{10} concentrations, sensors 3 and 5 appear to have some correlation between them regarding the peak concentrations. Sensor 3 always measures higher concentrations than sensor 5.

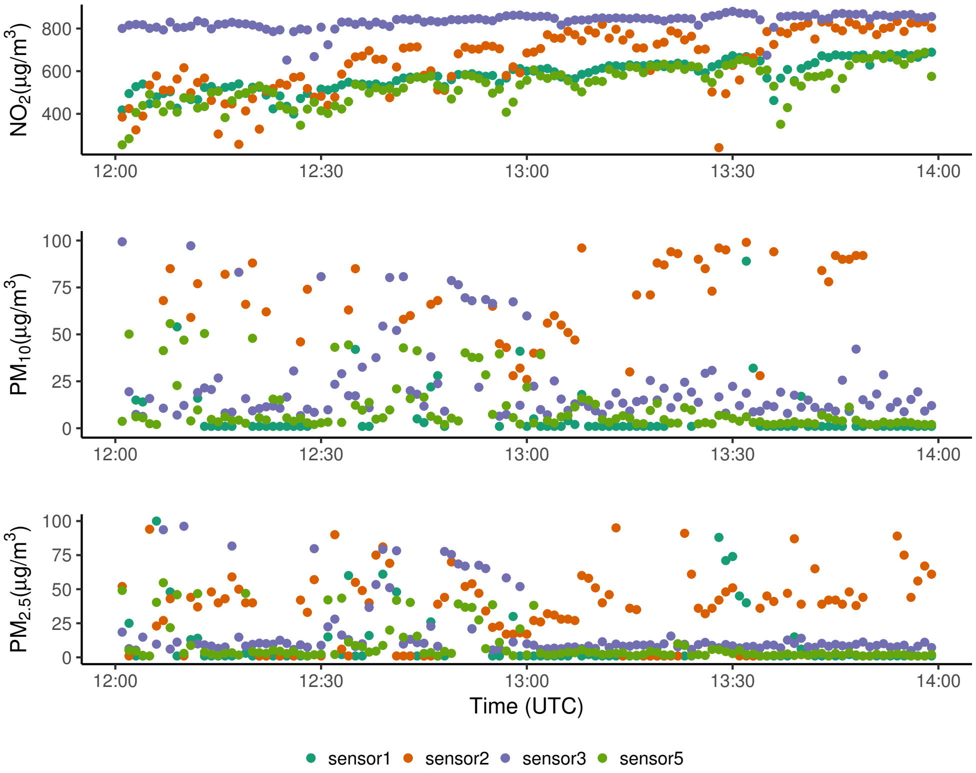


Figure 6.5: Comparison of four of the sensors during the calibration on the 14th of June 2019. This graph shows concentrations of NO_2, PM_{10} and $PM_{2.5}$.

6.2.2. Amsterdam measurements

In this section we describe the pollutant concentration patterns we found for the four different street types. The focus lies on concentrations of NO_2, $PM_{2.5}$ and PM_{10}. First, we will provide a brief overview of the conditions during the measurement sessions in the four different streets.

Street specifications

According to the types as defined by the SRM-1 model, street type 4 is assigned to street segments that cannot be classified in any of the other classes. From the visual representation shown in Figure 2.3, these streets are open and not obstructed by buildings. Therefore, one could assume that this street type has lower pollutant concentrations, thanks to higher dilution; this is because it is not surrounded by buildings, which is different from the other street types. However, in the case of the street we chose, it turned out to be the most busy street in terms of traffic and transit when compared to the other streets. This definitely affects the concentrations of different air pollutants, yielding worse air quality than what would be expected from looking at the street type alone. The streets of type 1 and 2 were also busy with moderate and heavy traffic, while the type 3 street had a low traffic intensity. In addition, the locations of street types 1 and 3 had a significantly bigger number of trees along the street compared to the locations of street types 2 and 4. This also affects the concentration and dilution of pollutants.

This makes clear that the street type is just an indication for the air quality levels for every street. In fact, the model takes into account many more parameters, such as traffic intensity, traffic speed etc. to estimate the air quality. Keeping this notion in mind will help to better understand the graphs presented in the next sections.

Street measurement values

The graphs presented in Figures 6.6 and 6.7 present the recorded concentrations of $PM_{2.5}$ particles by our sensors for all four different street types. The annual $PM_{2.5}$ value, obtained through the NSL-Monitoring tool, is also plotted for each graph. This line, in combination with the three measurements, shows well how much the $PM_{2.5}$ concentrations fluctuate during the day.

Under no circumstances should someone extract reliable conclusions by direct comparison between the annual value and our sensors' values. The time span of each of our measurement sessions was four hours. This implies that these measurements are greatly affected by local conditions during that specific four-hour time slot. To obtain values that can be directly compared to RIVM's annual values, the time span of the measurement campaign should be at least one year. The mean of those values would be directly comparable to the annual value calculated by RIVM.

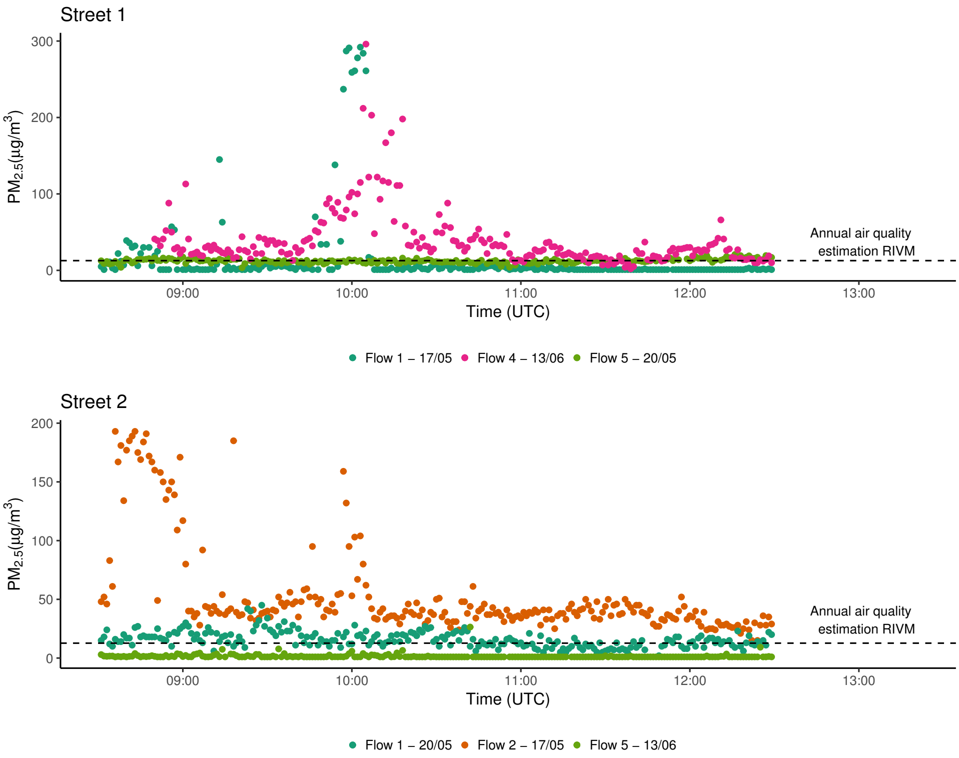


Figure 6.6: The measured $PM_{2.5}$ concentrations for street types 1 and 2.

Figure 6.6 shows street types 1 and 2 and the corresponding $PM_{2.5}$ concentrations. There is a quite noticeable spread between the sensor predictions. Sensor 4 always predicts higher values compared to the other two sensors, but since the sensor could not be calibrated, it is unclear if sensor 4 has a bias. Regarding sensors 1 and 5, we see that they consistently predict lower values than the other sensors.

Something else noteworthy are the peaks around 10:00 (UTC), which is 12:00 (CEST). This could fit with the hypothesis that more people go out during their lunch break, and thus more movements are happening at that time. However, this hypothesis needs to be supported with further measurements, since the amount of data recorded is limited.

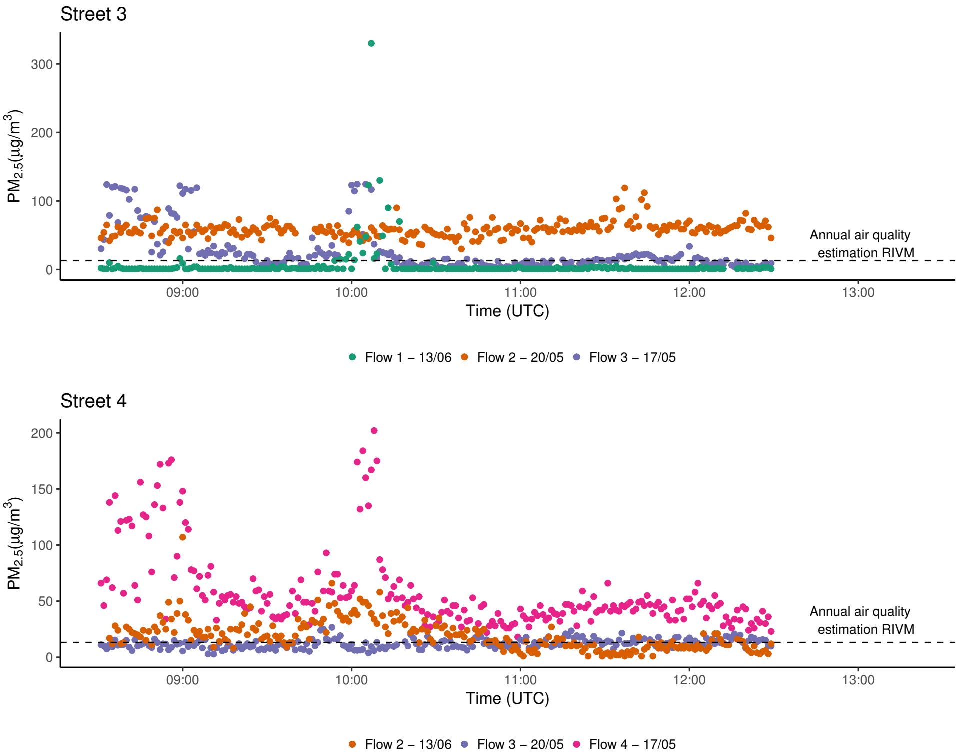


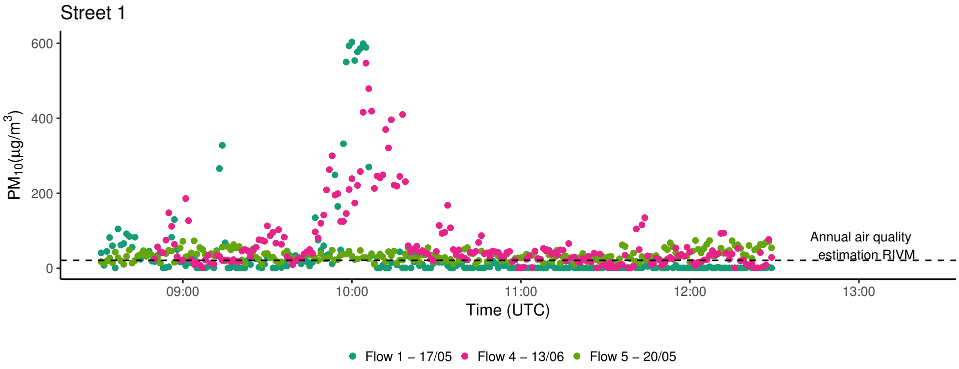
Figure 6.7: The measured $PM_{2.5}$ concentrations for street types 3 and 4.

The concentrations of $PM_{2.5}$ for street types 3 and 4 are shown in Figure 6.7. The graphs show that sensor 2 predicts larger values than sensor 1, which was also the case in Figure 6.6. Instead, sensor 3 predicts lower values for $PM_{2.5}$. Again, the peaks around 10:00 (UTC), 12:00 (CEST), are present.

Date	Street Type	Sensor	NO_2 ($\mu g/m^3$)	$PM_{2.5}$ ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)
17/05/2019	1	1	229.94	16.54	39.35
17/05/2019	2	2	101.33	102.31	123.10
17/05/2019	3	3	246.33	26.91	32.26
17/05/2019	4	4	638.72	58.29	139.97
17/05/2019	Reference	5	60.92	19.92	30.49
20/05/2019	2	1	35.45	15.87	49.18
20/05/2019	3	2	33.67	59.11	193.46
20/05/2019	4	3	29.70	12.19	27.51
20/05/2019	Reference	4	103.19	73.50	191.07
20/05/2019	1	5	298.51	12.44	34.63
13/06/2019	3	1	283.96	6.15	10.70
13/06/2019	4	2	124.47	19.73	35.02
13/06/2019	Reference	3	656.39	34.88	37.10
13/06/2019	1	4	40.76	39.91	71.39
13/06/2019	2	5	146.81	1.58	4.28

Table 6.7: The mean values for the five sensors during the three measurement days.

The concentrations of PM_{10}, for the four different street types, are shown in Figures 6.8 and 6.9. These concentrations, although fairly similar to those from $PM_{2.5}$, are consistently higher than these values. As before, we still observe that sensors 2 and 4 always provide higher overall pollution levels. This is further supported by computing the means across the four-hour measurements for all days (see Table 6.7), where sensor 4 consistently recorded higher pollution levels no matter of its location. This behaviour should be further analysed through longer calibration of the five sensors.



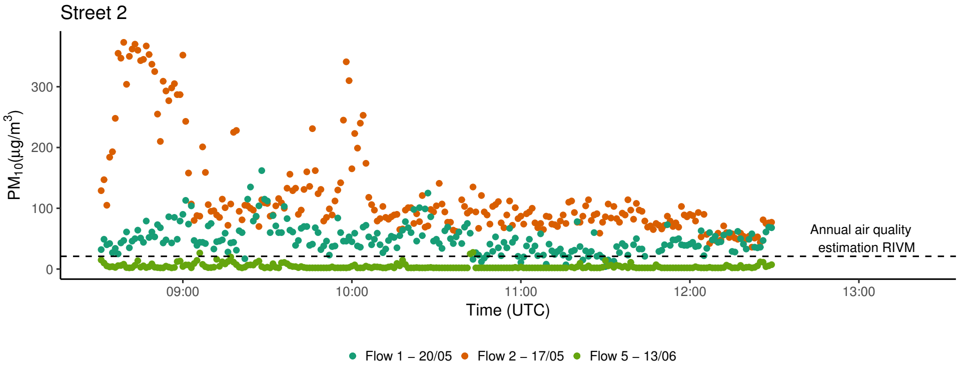
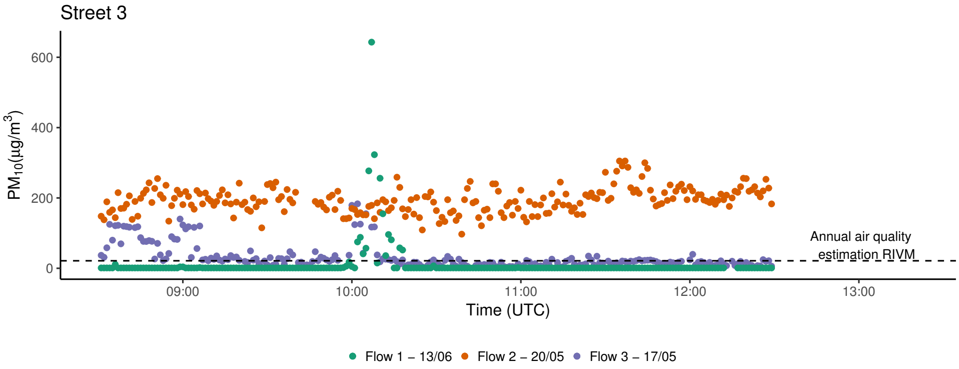


Figure 6.8: The measured PM_{10} concentrations for street types 1 and 2.



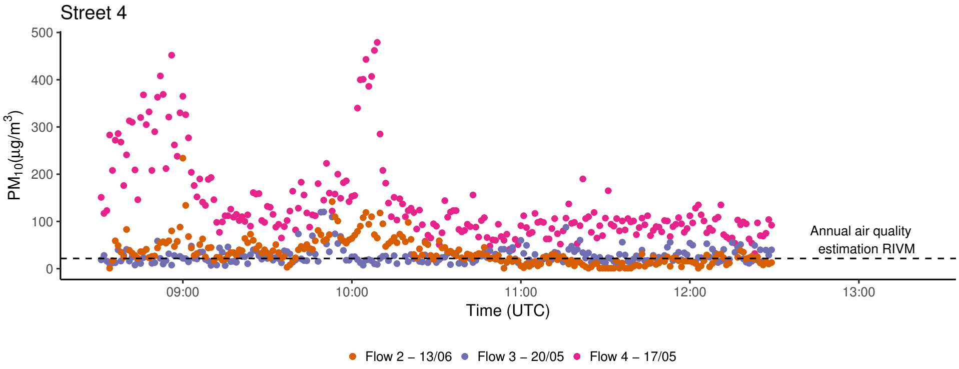


Figure 6.9: The measured PM_{10} concentrations for street types 3 and 4.

Lastly, the concentrations of NO_2 are compared for the four different street types. The results are shown in Figures 6.10 and 6.11. We see that the concentration spread in the NO_2 measurements is larger than that for particle matter. This can be specifically observed for sensors 1, 2 and 5. Measuring NO_2 is still a challenging task for small measurements, due to the uncertainties surrounding the chemical reactions that lead to the NO_2 production thanks to solar radiation [3].

When looking at Figure 6.10 into more depth, we see that sensors 1 and 5 show similar trends for concentration values across time for street type 1. When looking at street type 2, we see that sensor 1 reports lower concentrations when compared to sensors 2 and 5.

For street type 3, shown in Figure 6.11, we see similar predictions for sensors 1 and 3, while sensor 2 predicts much lower values. Instead, for street 4 sensor 2 predicts values larger than sensor 3, which could be related to different local conditions or sensor drifting across time. Additional measurements could help shed some light regarding these hypothesis.

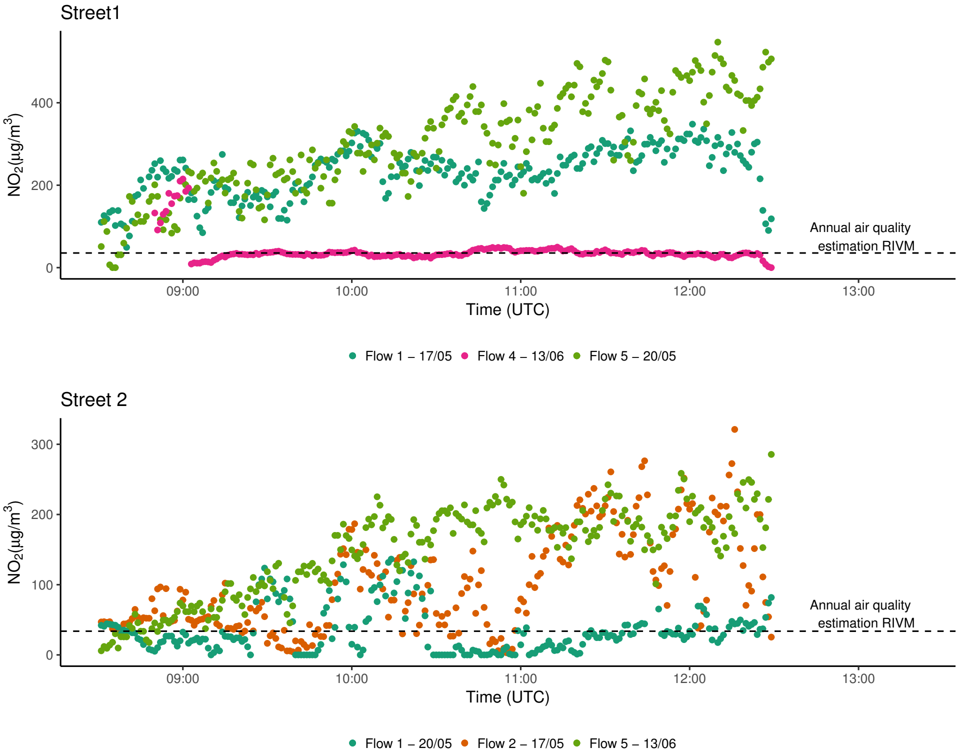


Figure 6.10: The measured NO_2 concentrations for street types 1 and 2.

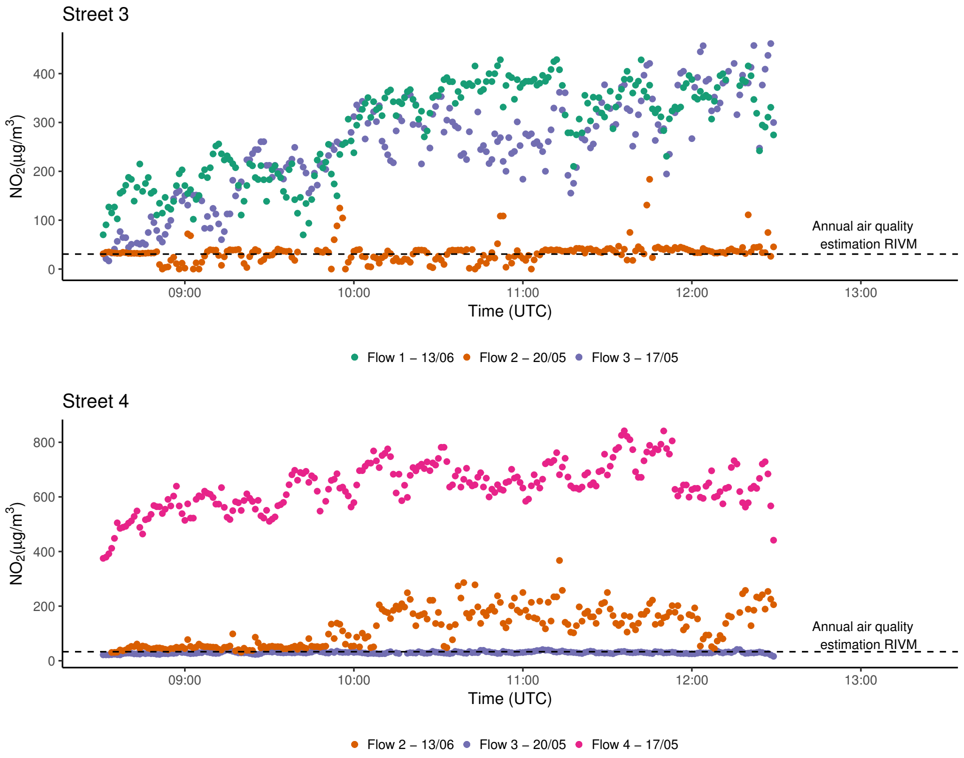


Figure 6.11: The measured NO_2 concentrations for street types 3 and 4.

6.2.3. Comparison to reference measurements

Each measurement day, one sensor (different sensor each time) is located at the same location in Vondelpark as close as possible to the reference station of the municipality of Amsterdam. This is done to obtain a context for the accuracy of the *Plume Labs Flow* sensors. In addition, the background pollution without contribution from traffic is measured.

The reference data provides hourly averages of different pollutant concentrations. Figure 6.12 shows the NO_2 concentrations for the three measurement days. We observe changes in concentrations more frequently than for PM in the sensors. Again, sensor 5 seems to be the one closest to the reference station data, while sensors 3 and 4 overestimate the values.

Figure 6.13 shows the $PM_{2.5}$ concentrations for the sensors and reference station. When the reference data was retrieved from the API (Application Programming Interface), we found that the hourly averages for $PM_{2.5}$ were not available for the 13th of June 2019. Looking at the graphs, when excluding the peaks shown by sensor 5, we see that the concentrations measured by this sensor match the hourly level reported by the reference station. However, sensor 4 seems to over-predict the reference measurements consistently. This could be because of the probable bias of sensor 4, which has already been pointed out before.

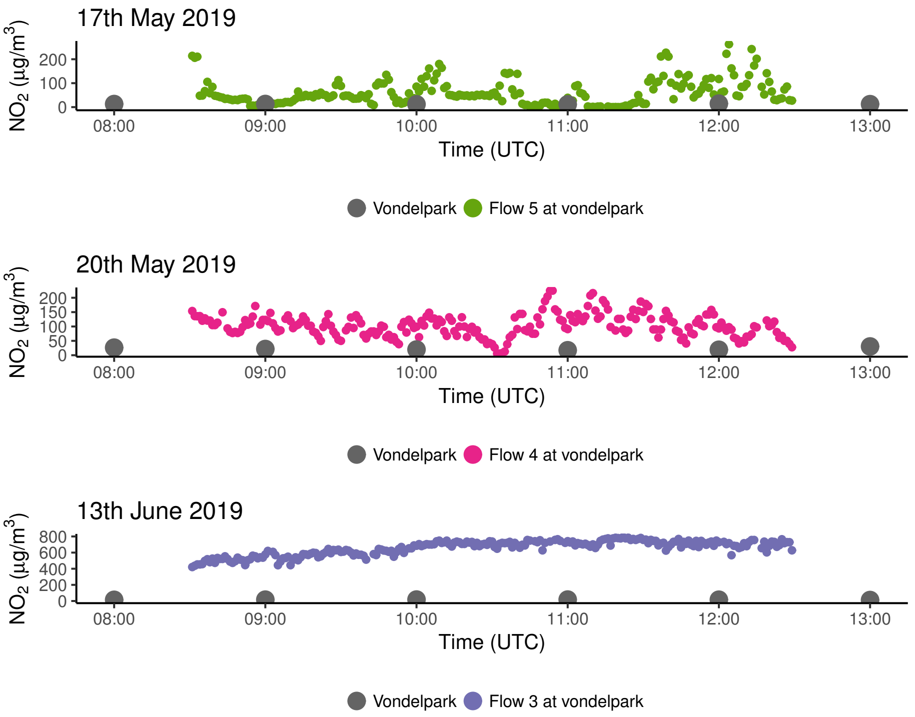


Figure 6.12: NO_2 concentrations for sensors 3, 4 and 5, including the hourly average of the reference station measurements.

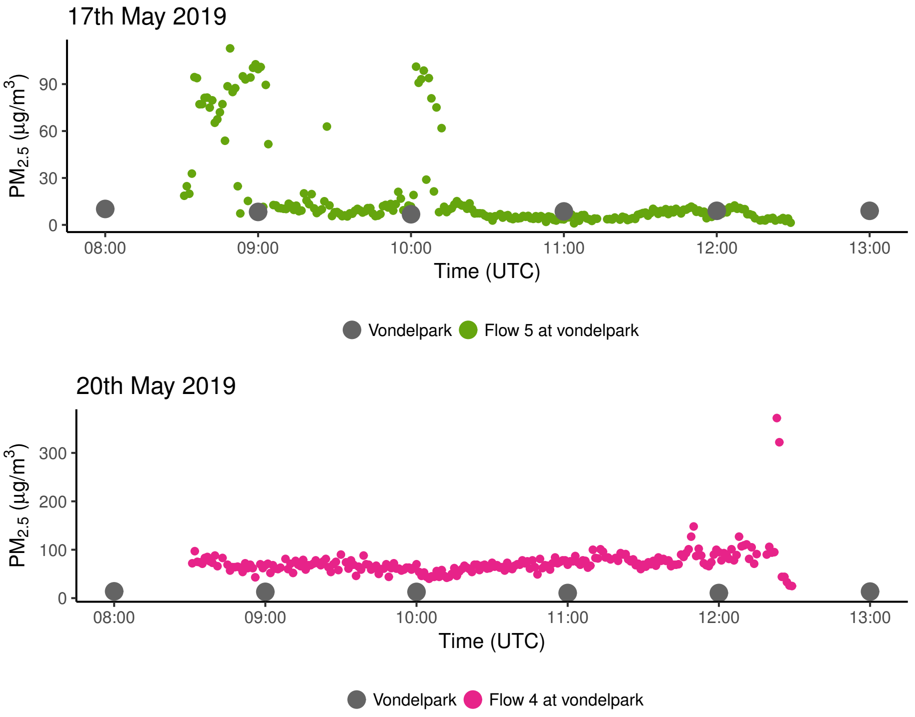


Figure 6.13: $PM_{2.5}$ concentrations for sensors 4 and 5, including the hourly average of the reference station measurements. $PM_{2.5}$ was not measured by the reference station during the last measurement day (13/06/2019), and is therefore excluded in the plot.

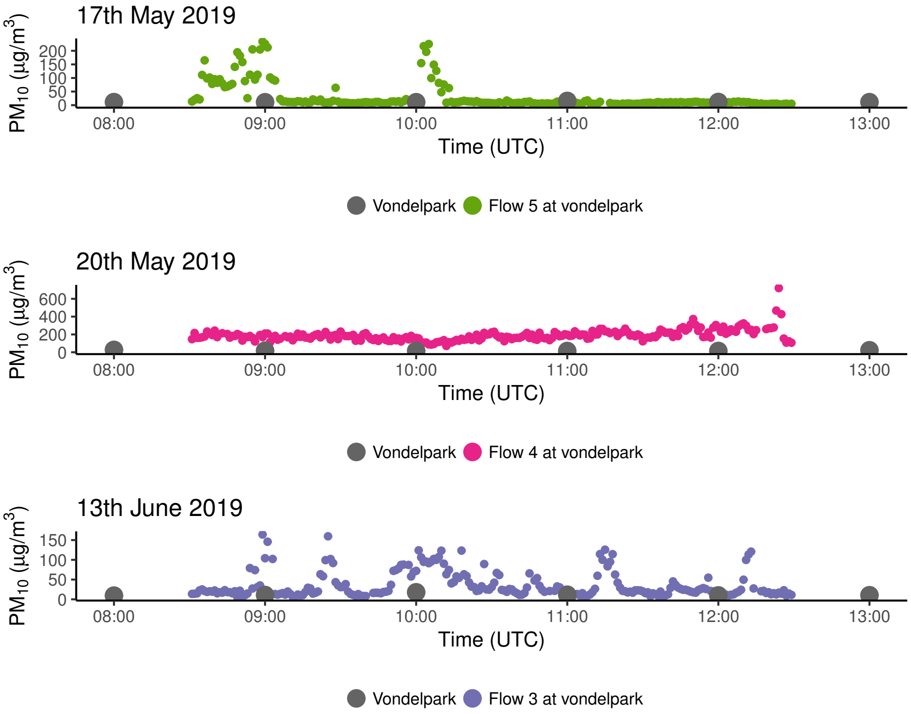


Figure 6.14: PM_{10} concentrations for sensors 3, 4 and 5, including the hourly average of the reference station measurements.

Lastly, the PM_{10} concentrations are shown in Figure 6.14. The patterns look similar to those for the measurements of $PM_{2.5}$. Again, sensor 5 is close to the reference data when the peaks are excluded. Both sensors 3 and 4 overestimate the concentrations compared to the hourly reference values.

The thorough analysis in this section gave insight on the air pollution fluctuation in the five locations; however, it doesn't show a strong correlation between street types and air pollution levels. Also, the consistent spread from the sensor measurements is an important conclusion, highlighting the need for systematic daily sensor calibration, before and after the experiments.

7

Conclusions and future work

In this chapter the research questions of this research will be answered and the conclusions will be discussed. Furthermore, we describe future work that could be carried out to improve the automatic street classification algorithm, and we elaborate on how the air quality measurements can be conducted in a more reliable manner.

7.1. Conclusions

The study presented in this report originated from the necessity to find a feasible and automatic solution to street classification in urban areas to calculate the air pollution produced by traffic. The project originated as an initiative of our client at RIVM, who saw the opportunity to gain new insights on the problem from a geo-spatial perspective. The project spanned over ten weeks, investigating two main research questions:

1. Would it be feasible to automate the whole process of street classification up to the scale of a municipality like Amsterdam and still have valid results?

We developed an automatic street classification algorithm that takes as input street and building (height) data (see Chapter 3) and classifies street segments according to RIVM's '*Standaard Reken Methode - 1* (SRM-1) model. The main focus area was on the city of Amsterdam, and the results of the automatic street classification for the double-sided ray casting method looks promising (see Chapter 6). It shows the feasibility to automatically classify streets based on the four street types as required by SRM-1. First, the similarities of the outcomes with the manual classification carried out by '*Nationaal Samenwerkingsprogramma Luchtkwaliteit*' (NSL) show that the algorithm is reliable and that the topic can be further researched. Moreover, the scalability of the algorithm is reflected in its runtime, which makes it feasible to run the algorithm for the whole country. The algorithm, as in its current state, has some limitations. The suggestions for further improvements are discussed in section 7.2.1.

2. Would it be possible to determine a street type, based on the Plume Labs sensors measurements?

In the limited time frame of this project, we did not find any correlation between air pollution levels and the street classification. We performed air quality measurements on five measurements locations in the city of Amsterdam, see Chapter 5. However, we only measured on three days and the short amount of time strongly affects the results. Therefore, only a qualitative comparison can be made; for the first day the measurements and reference values are quite similar, but for the second and third day larger differences are observed. Also, the yearly averages near the measurement locations are not directly comparable to our measurements. It does, however, show that during the day the pollutant concentrations fluctuate greatly.

Another problem with the results could be related to the sensor calibrations. Due to constrains in time, we could not assess each sensor independently against the reference stations. This resulted in little information regarding their measuring accuracy. Sensor calibration was only performed at the end of the measurements. However, performing calibration every measurement day, before and after the data collection, would have provided further insight in each sensor's peculiarities. This is something to keep in mind for future measurement campaigns with small air quality sensors.

Additionally, it must be considered that measurements were performed only in a limited number of streets, each one representing a street type and one to serve as reference with the Vondelpark reference station. These locations are not necessarily representative for other streets with the same classification. Lastly, the *Plume Labs* sensors are sensitive to many factors, such as rain and humidity. They also drift with time. These factors might have influenced the overall performance of the measurement campaign.

From the measurement campaign we gained useful experience that can be used in future similar ventures. Suggestions on how to improve the measurements are discussed into more detail in section 7.2.3.

7.2. Future work

This section contains suggestions for possible improvements of the algorithm, and recommendations on how the air quality measurement campaign can be made more successful. For the first part, a distinction is made between aspects related to the implementation of the algorithm and the SRM-1 model. It should be noted that input from an air quality expert is required for most specific and targeted improvements, to implement the optimal solution from an air quality point of view.

7.2.1. Algorithm related improvements

Buildings and street proportion

The algorithm detects all the collateral buildings of each street segment in order to classify the street. Thus, if few buildings are present on both sides of the street, the algorithm will classify the whole street segment based on very few buildings. This would lead to incorrect classification in cases where the number of buildings is relatively small and/or does not "cover" the whole length of the street segment.

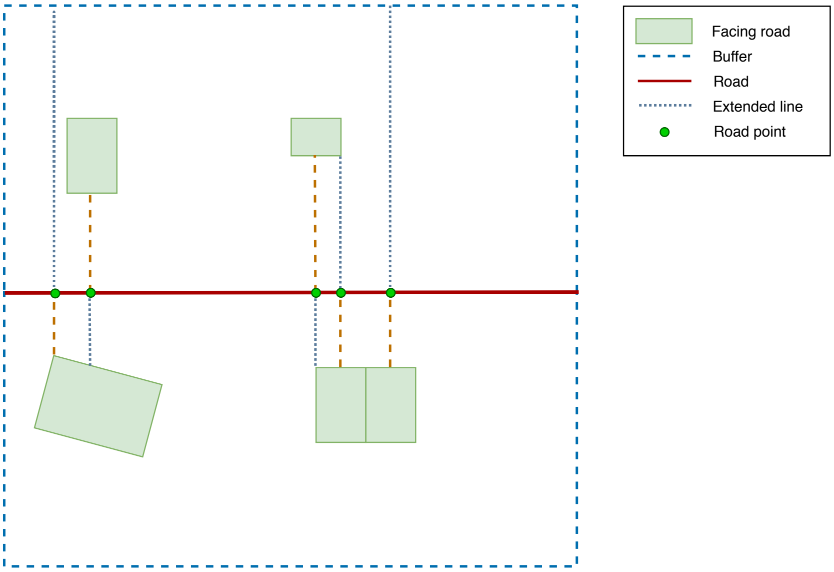


Figure 7.1: Example of a street segment with few collateral buildings.

Figure 7.1 provides a graphical example of this issue. The street segment will be classified based on the five buildings, but since these buildings do not actually form an urban canyon the classification will be incorrect.

In dense urban environments such a case does not occur very often, and the algorithm is expected to perform well. However, when the urban environment is not dense, for example in suburbs or rural areas, the algorithm could lead to incorrect classifications. For such applications the algorithm should be improved to take in consideration the proportion of the street length covered by buildings. In this way, if two small buildings are directly opposite of each other, but the street segment is a lot longer than the building facades facing the street, the street can get a different classification taking this knowledge into account. Flaws of the current Voronoi method are discussed in the next section, together with proposed solutions. These could also help solving the limitation described in this section.

Length of facades as weights

Currently, weights are assigned to buildings in the weighted average classification method. Each weight corresponds to the number or Voronoi cells of a building, divided by the total number of cells “touching” the street segment. This method presents some limitations; first it does not allow to compute the percentage of the street that is ‘surrounded by’ buildings, and secondly, it does not consider that Voronoi cells could have slightly different dimensions between buildings.

In the ray casting methods, to aggregate the classification from the street points to the street segment, a voting system is implemented. It counts how often each class occurs along the street segment, and the most frequent class is assigned to the street. Initially, we thought that the most accurate method to aggregate would be to calculate the length of the facade of each building facing the street, and to use this as the weight with the respect to the total street segment length. Soon, we realised that despite the fact that it looks simple from a human perspective to find a way to calculate facade lengths, it is not that trivial. Calculating the facade length based on the polygon geometry representation of every building is a big challenge, because one would need to identify only the vertices facing the street, out of an arbitrarily arranged list of vertices for every polygon.

Another approach is to calculate the distance between the projected building points on the street segment, as they could be considered to be the projection of the facade on the street. This method though, can only be used for the single-sided ray casting method. In many cases, it happens that the closest point of the building to the street is almost at the corner of the building. But there is no guarantee that for the next building the corner will be the closest point to the street. So, if for any reason the “equivalent” corner is not the closest point to the street, then the length between two consecutive points has no physical meaning.

For possible future improvements we consider this challenge one of the most important ones to be dealt with. The capability to calculate facade lengths for all collateral buildings could also be used to solve the previous aspect. For example, an index could be created to quantify the percentage of the street segment length “surrounded by” buildings. That index could be used to make the algorithm much smarter and accurate on the classification procedure.

Splitting streets based on classification distribution

The ray casting methods project points on the street segment. It can happen that the classification of these points is uniform in parts of the street, i.e. consecutive points have the same class. In these cases, it is desirable to split the street segment between the transition points, i.e. points were the classification changes. In this way, the accuracy of the output of the classification algorithm would improve because less generalisation along the street segment is introduced. The implementation of this method requires the projected points to be sorted along the street segment. Decisions should also be made about cases where the distribution is almost uniform in one part of the street, e.g. one point has a different class from the other points in that street sub-part.

Testing of different input datasets

The automatic street classification algorithm was developed using the BGT building geometries, which are integrated with the 3D BAG building heights. The reason is that BGT dataset contains building footprints, while BAG includes both underground buildings and roof construction outlines. However, the BGT geometries could lead to issues, related to the presence of columns for example, which are excluded in the BAG dataset. Further research could include testing of new input datasets. For example, a thorough investigation of BAG attribute data could enable to discard underground geometries while eliminating low constructions and columns.

Validation of the algorithm

Currently we perform visual analysis to obtain comparisons between results of the automatic classification algorithm and the NSL receptor points. In the future, an automatic validation method could be implemented. For example, receptor points could be associated with the street they belong to; this could be done using attribute values available in the street dataset of the NSL. Then, for each segment correspondences or discrepancies could be checked. It must be noted that the NSL street dataset only contains street geometries where there are receptor points present. All other streets are eliminated in this dataset.

7.2.2. SRM-1 model related improvements

The 100 meter rule

According to RIVM's SRM-1 model, the street segments need to be at least 100 meters long for the classification to be suitable for air pollution calculations. The road network dataset we worked with has the streets segmented based on junctions, which made the most sense for the classification. There are many segments that are less than 100 meters, and which according to the model should be merged to form longer segments. This specification was not implemented in the algorithm due to a number of reasons, which are listed below.

First, it is not straightforward to merge street segments in a road network, especially around junctions. This is because consecutive street segments are not labelled consistently (e.g. through ID), and the angles between the segments should be considered. On a junction for example, one would have to make sure the algorithm would recognise segments in the extension of the original street, and avoid merging street segments perpendicular to the original segment.

In addition, the street segment is the primitive entity used for classification. Adjacent segments, especially around junctions, may have different classifications if merging is applied; thus, this could lead to lower overall accuracy. The same classification would be applied to larger street segments, inducing inaccuracy. Finally, by implementing this rule it would directly contradict with the 15 meter rule explained in the next section.

Our suggestion is that the minimum length of the street segment, required by the SRM-1 model, should be reconsidered as it should be adapted to the city's layout. Given our Geomatics expertise, this suggestion comes from a spatial-analysis point of view, not taking into consideration air propagation model implications.

The 15 meter rule

The SRM-1 model also specifies that if along a 100 meter segment the sum of the distances among buildings on the same street is more than 15 meters, this should lead to different street types. This implies splitting the street segment, so that the distance among buildings of every split part is less than 15 meters. This statement contradicts with the aspect explained in the 100 meter rule. If one was supposed to merge segments to form a segment that is larger than 100 meters, it would be more likely that the 15 meters rule would not be fulfilled. This would result in splitting the segment again, which would contradict the 100 meter rule.

Apart from the contradiction, implementing this capability was also not feasible in the time frame of this project, again because of increased complexity. It is already quite a challenge to "sort" the buildings in a consecutive way with respect to a street segment and to calculate the distance between them.

Selection of appropriate streets

The results for the city of Amsterdam look promising, but it must be kept in mind that the SRM-1 model is only applicable to urban environments. The input datasets should adhere to this requirement. Other types of streets, such as highways, might have other street classification requirements. The datasets used in this research might contain non-suitable streets, because no distinction is made between areas that might require the use of a different classification model. Automatically detecting which model should be applied did fall outside of the scope of this research, but could be an interesting topic to investigate in the future. For the '*Nationaal Wegen Bestand*' (NWB) dataset no appropriate attribute was found for automatic detection, but there might be other datasets available which can be used to obtain the necessary information. For example, a dataset with the location of built areas known as the "*bebouwde kom*".

7.2.3. Air quality measurement improvements

Sensor calibration and validation

As stated before, the sensors were only calibrated once after all measurements sessions. This should have been done before and after the measurements each day, in order to characterise each sensor's behaviour across time, and to detect different meteorological conditions on-site. In addition, the sensors were compared to each other from 14-16h in the afternoon, on the 14th of June 2019. This is not a good time of day, especially for NO_2 concentrations since they are affected by solar radiation [3]. The calibration time frame was fairly short, because there were limitations in time and the data that had to be retrieved from *Plume Labs* via e-mail. Especially the latter could sometimes took long.

In the optimal situation, the sensors are placed next to a reference station for an extended period of time to validate their measuring behaviour. In the scope of this project, this was not feasible. Our main focus was to measure the pollutant concentrations and to see how the sensors compare to each other.

Quantitative assessment of sensors

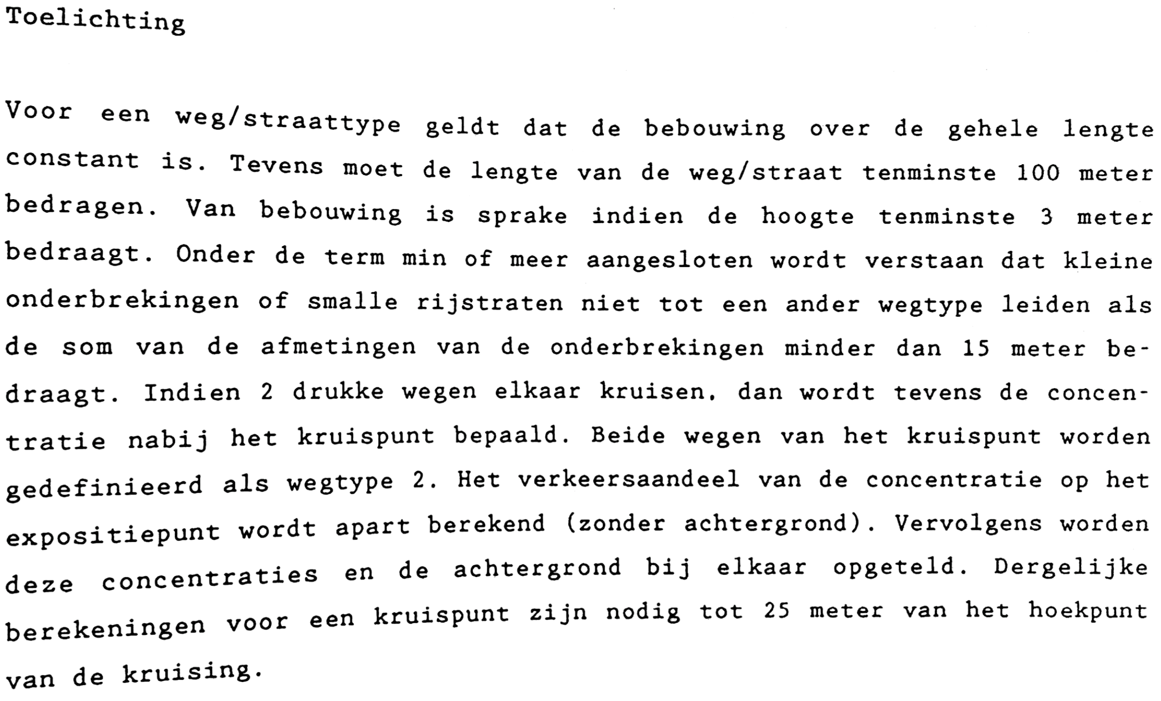
The current measurement values do not allow for a quantitative analysis of the results; only a qualitative analysis can be performed. The measuring period is too short, and not enough different days and types of weather have been covered to ensure repeatability. More measurements is the key for this aspect. It would be interesting to quantify the performance of the different sensors, since to our knowledge their performance has only been measured in lab conditions.

Computation of yearly values

The relationship between air quality and the different street types can be further investigated. When there is a discrepancy between the automatic street classification and the NSL receptor classification, new yearly air pollution values can be calculated with the SRM-1 model. The inputs for the model will be the current and newly established street type as computed by the automatic classification algorithm. This can either positively or negatively impact the pollutant concentrations. Especially the cases where the air pollution deteriorates are of interest for further investigation.

A

RIVM scanned document



Toelichting

Voor een weg/straattype geldt dat de bebouwing over de gehele lengte constant is. Tevens moet de lengte van de weg/straat tenminste 100 meter bedragen. Van bebouwing is sprake indien de hoogte tenminste 3 meter bedraagt. Onder de term min of meer aangesloten wordt verstaan dat kleine onderbrekingen of smalle rijstraten niet tot een ander wegtype leiden als de som van de afmetingen van de onderbrekingen minder dan 15 meter bedraagt. Indien 2 drukke wegen elkaar kruisen, dan wordt tevens de concentratie nabij het kruispunt bepaald. Beide wegen van het kruispunt worden gedefinieerd als wegtype 2. Het verkeersaandeel van de concentratie op het expositiepunt wordt apart berekend (zonder achtergrond). Vervolgens worden deze concentraties en de achtergrond bij elkaar opgeteld. Dergelijke berekeningen voor een kruispunt zijn nodig tot 25 meter van het hoekpunt van de kruising.

Figure A.1: Physical scan of an document by RIVM from the 1980s about the road classification.

B

Datasets and results for other cities and towns

In this appendix four other datasets are presented; the cities of Rotterdam, Delft and Groningen, and the smaller town Posterholt. The automatic classification algorithm was also ran for these datasets. The results are included here as well.

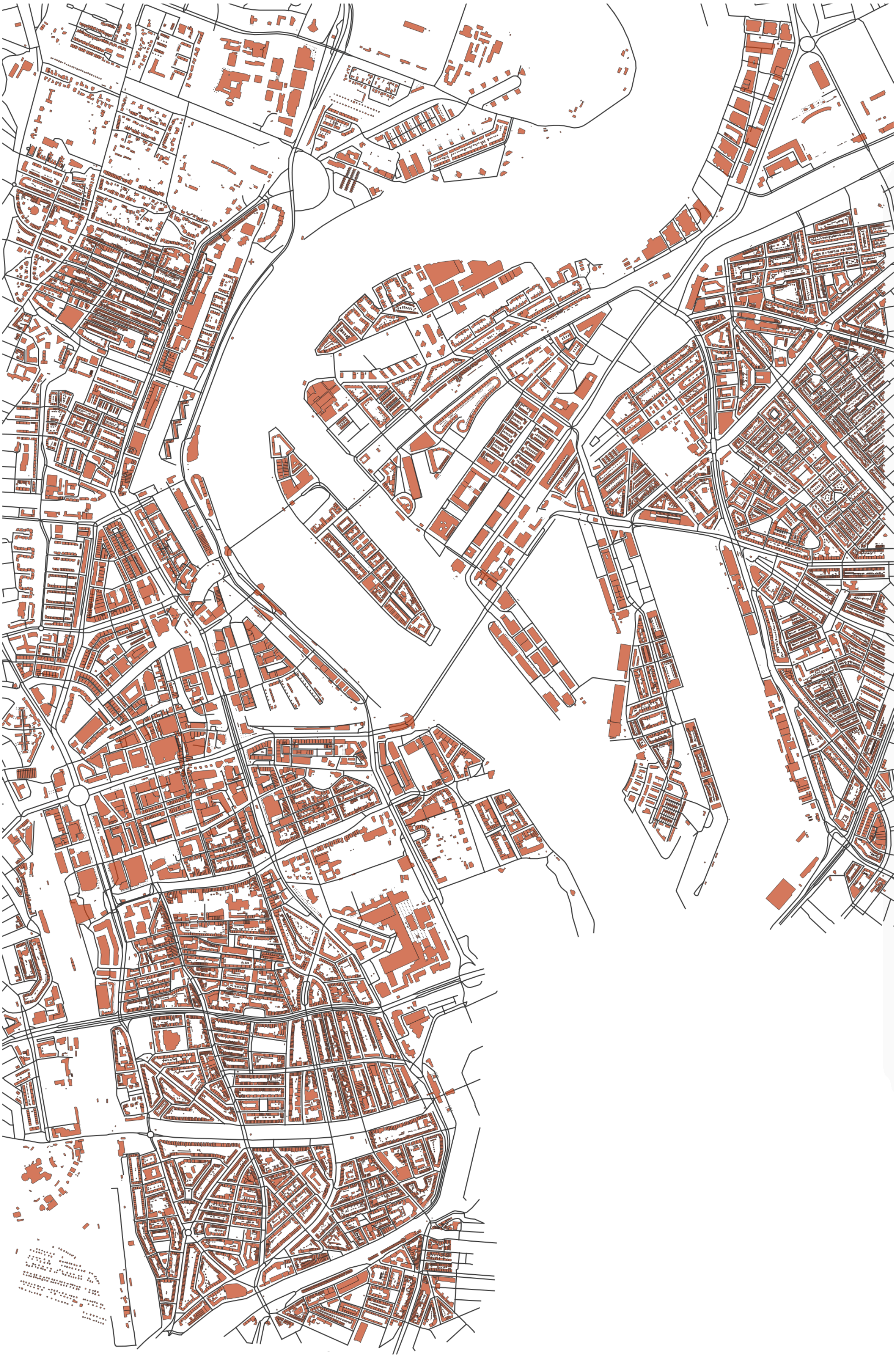


Figure B.1: The BGT building data and NWB road data as it will be input into the road classification algorithm for the city of Rotterdam, Zuid-Holland. The map is turned 90 degrees counter clockwise.

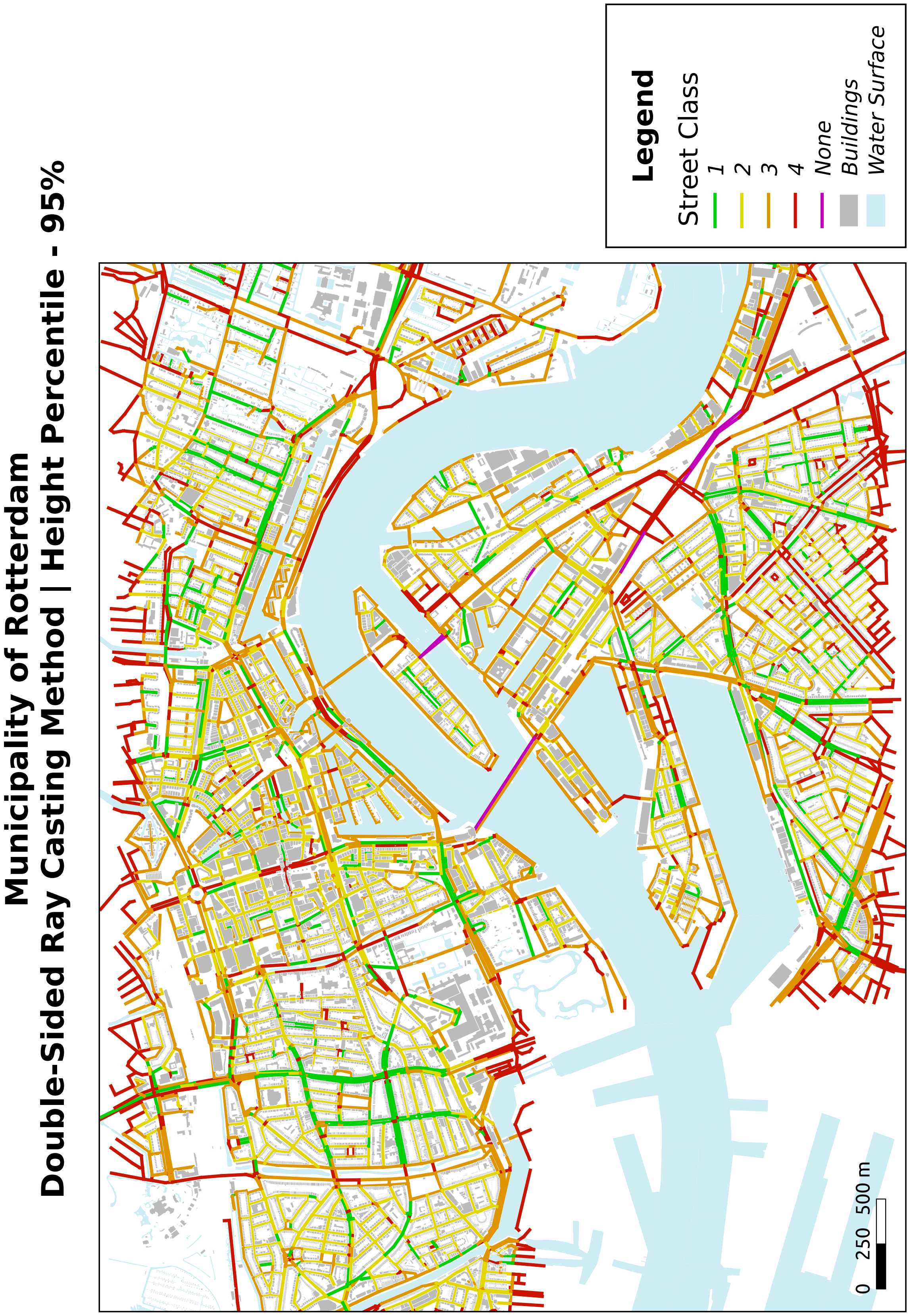

Municipality of Rotterdam
Double-Sided Ray Casting Method | Height Percentile - 95%
Legend
Street Class
1
2
3
4
None
Buildings
Water Surface
0 250 500 m

Figure B.2: The result of the road classification algorithm for the city of Rotterdam, Zuid Holland.

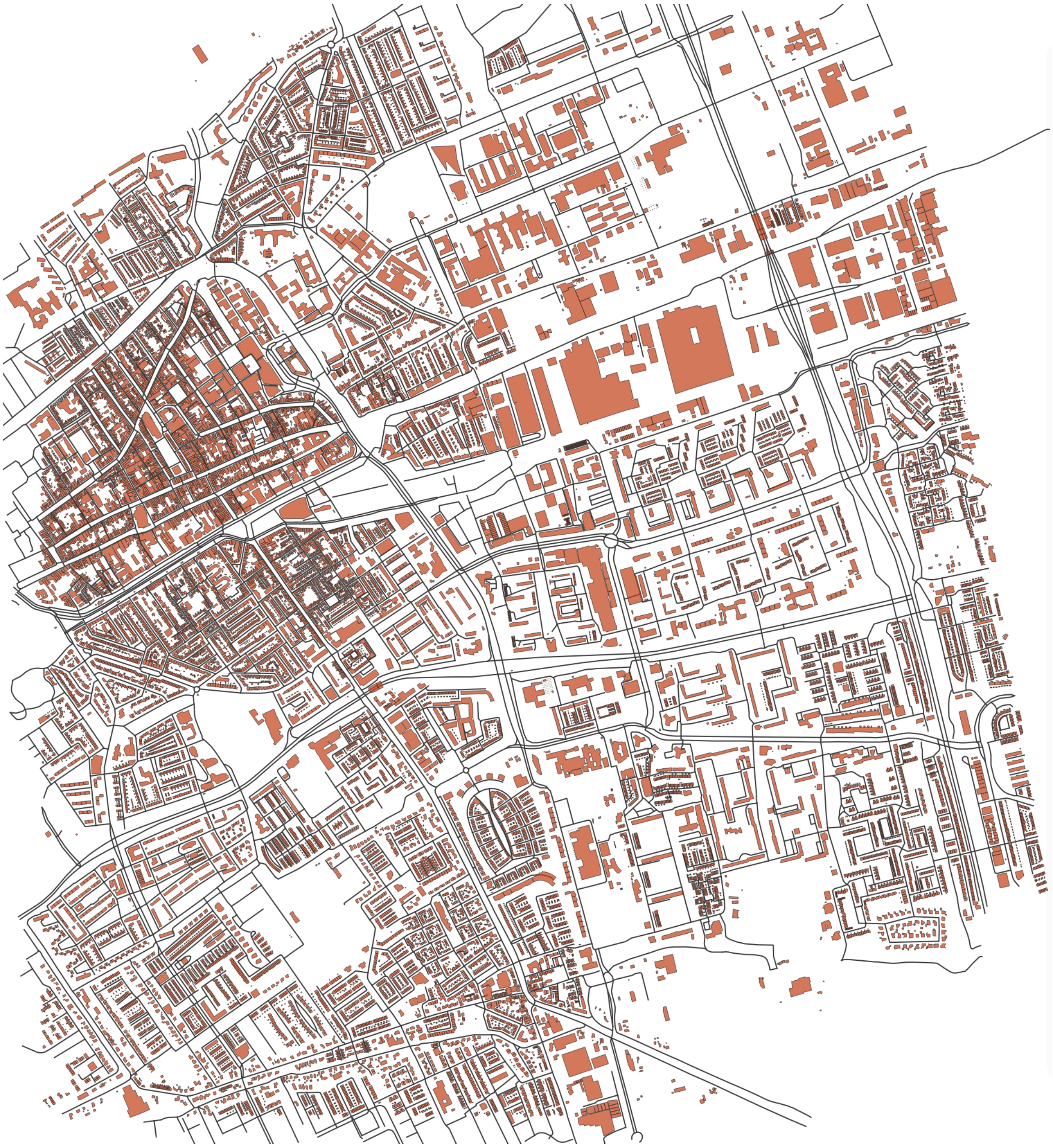


Figure B.3: The BGT building data and NWB road data as it will be input into the road classification algorithm for the city of Delft, Zuid-Holland. The map is turned 90 degrees counter clockwise.

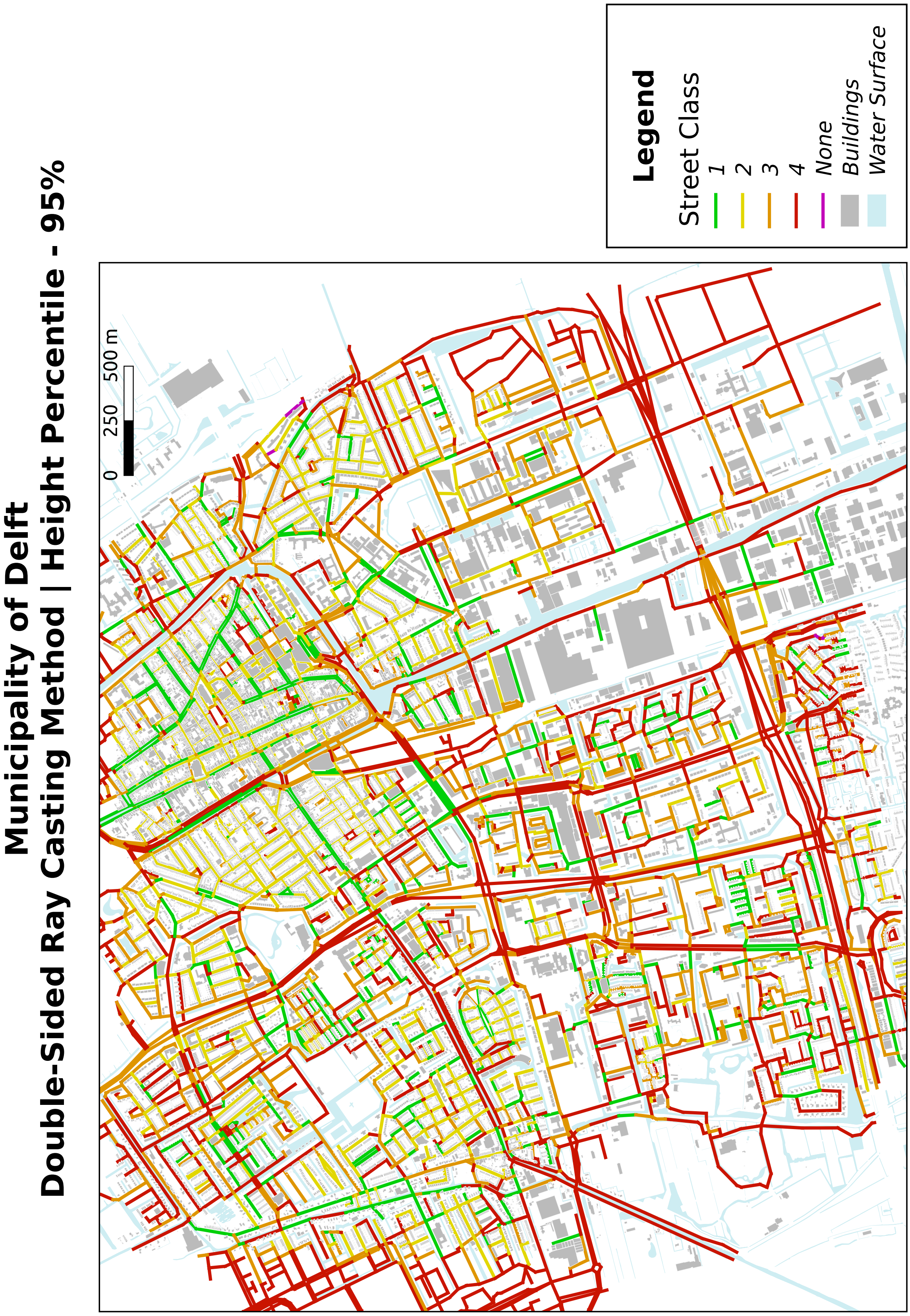


Figure B.4: The result of the road classification algorithm for the city of Delft, Zuid Holland.

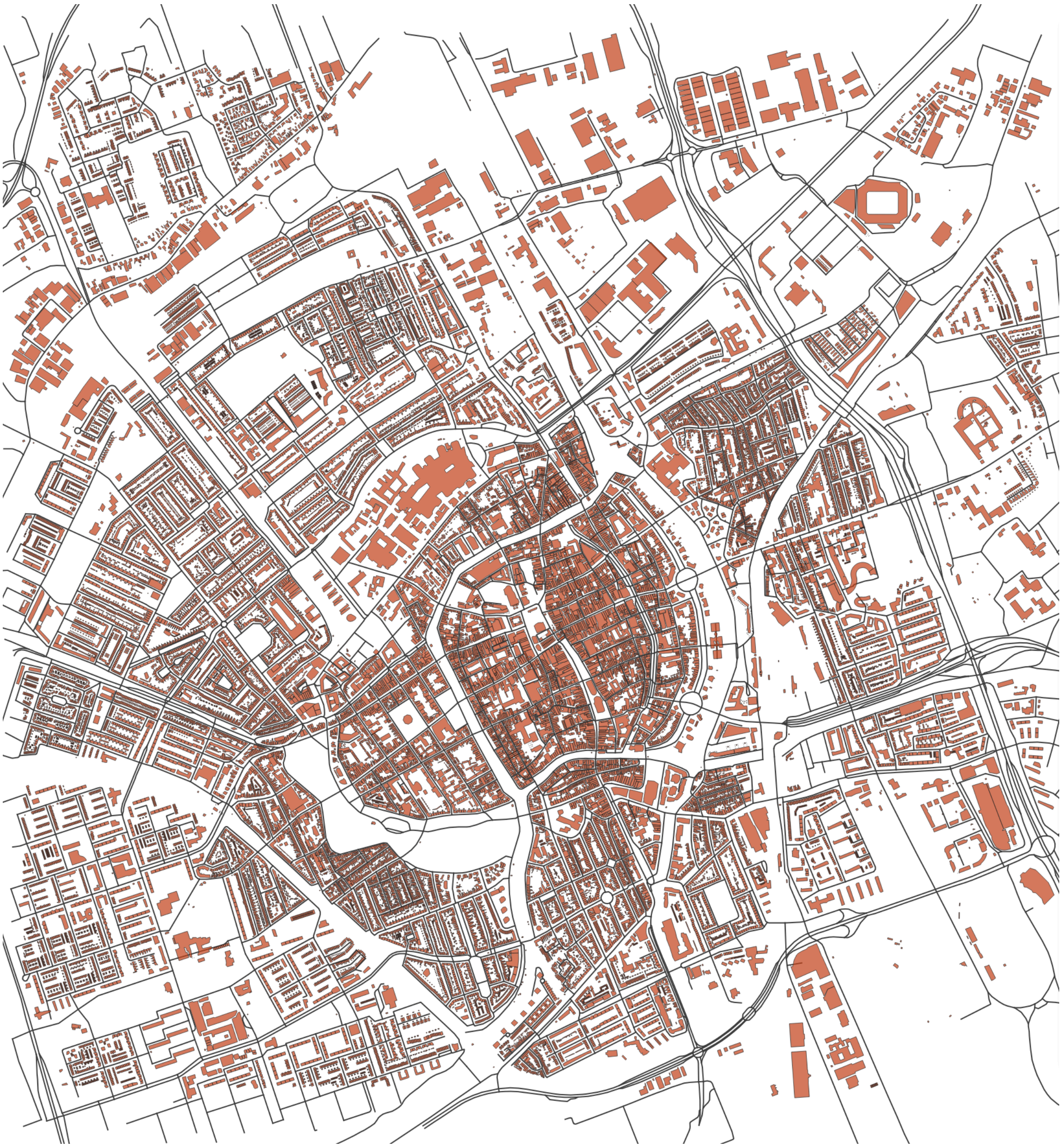


Figure B.5: The BGT building data and NWB road data as it will be input into the road classification algorithm for the city of Groningen, Groningen. The map is turned 90 degrees counter clockwise.

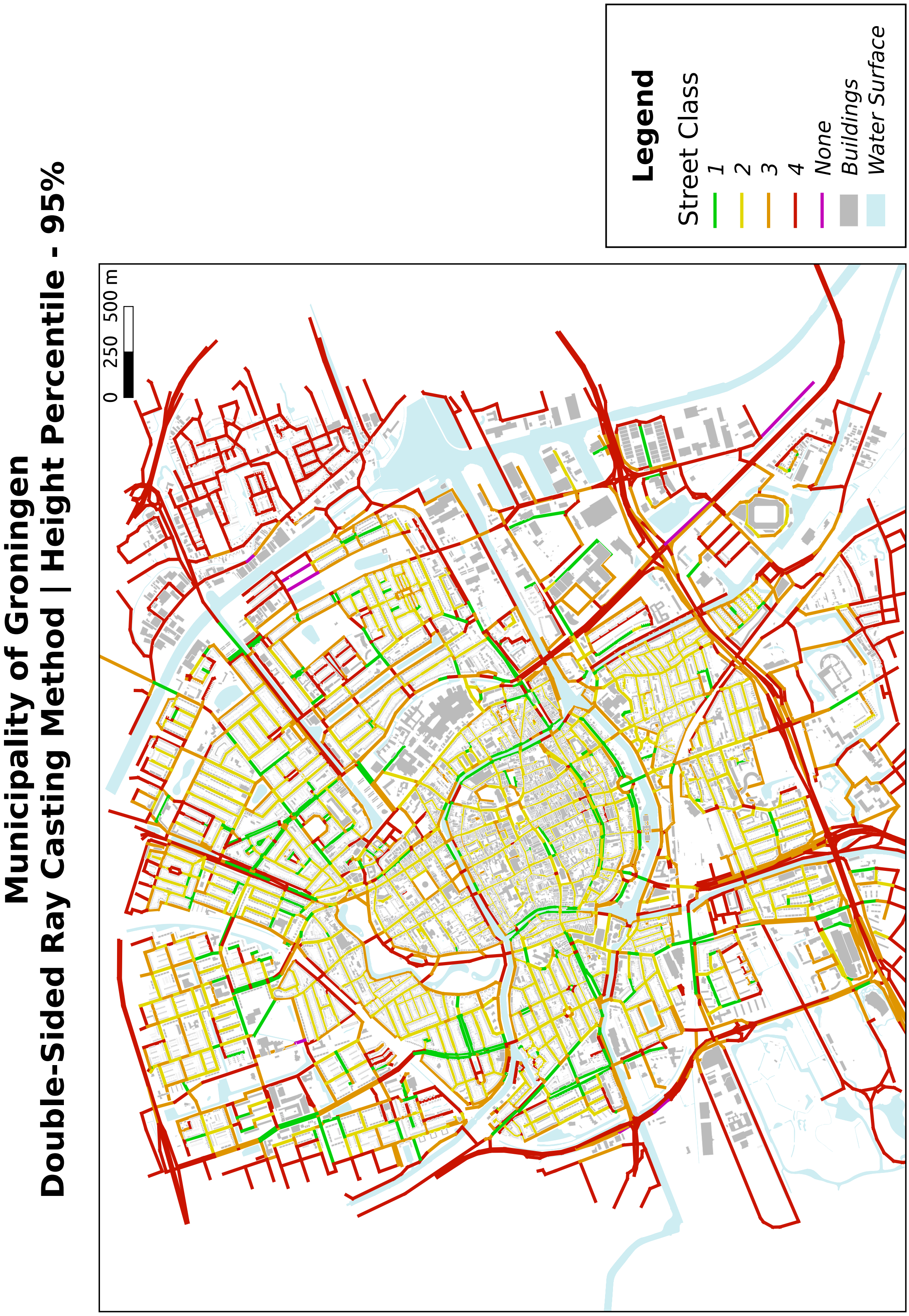

Municipality of Groningen
Double-Sided Ray Casting Method | Height Percentile - 95%
0 250 500 m
Legend
Street Class
1
2
3
4
None
Buildings
Water Surface

Figure B.6: The result of the road classification algorithm for the city of Groningen, Groningen.

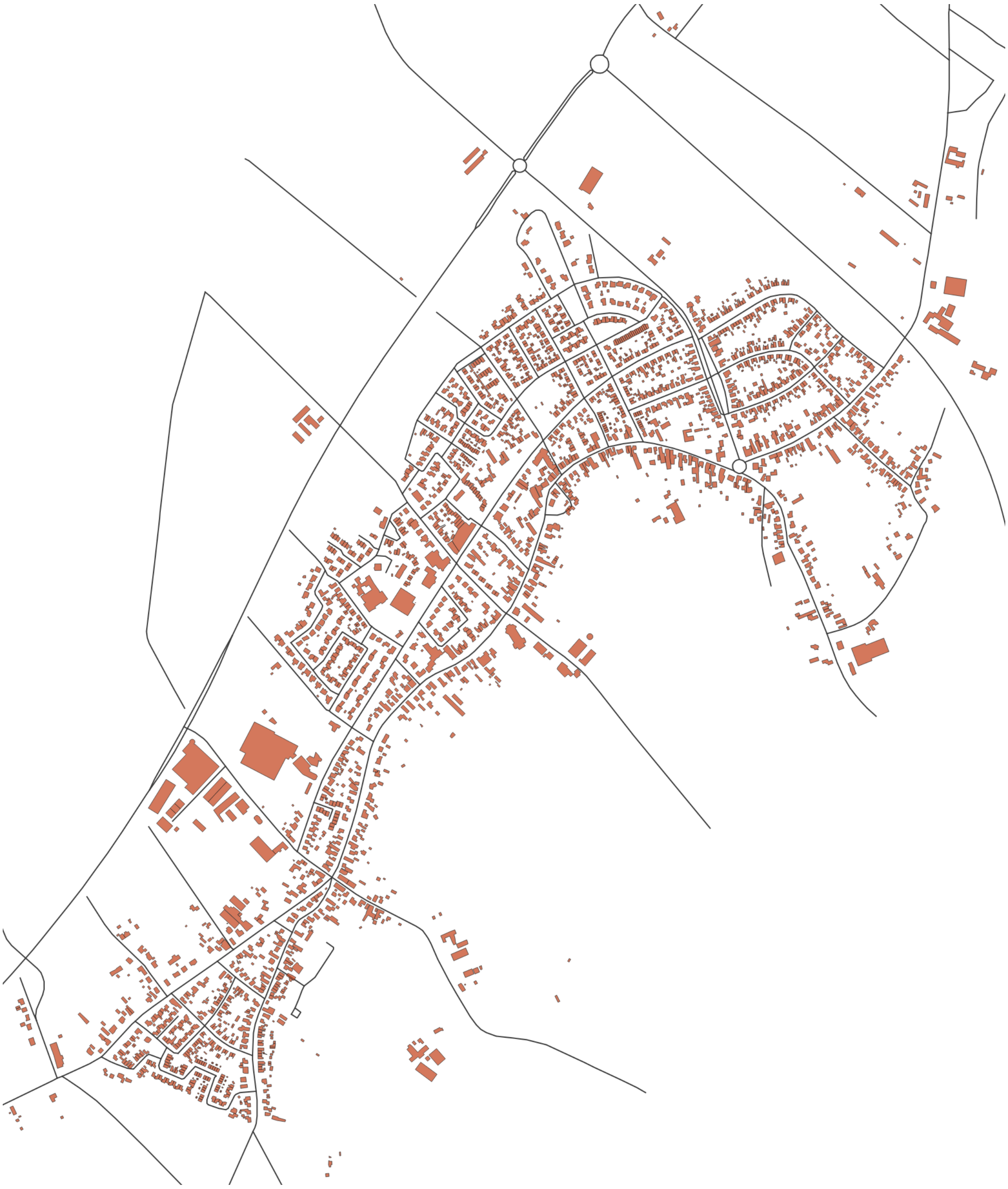


Figure B.7: The BGT building data and NWB road data as it will be input into the road classification algorithm for the town of Posterholt, Limburg. The map is turned 90 degrees counter clockwise.

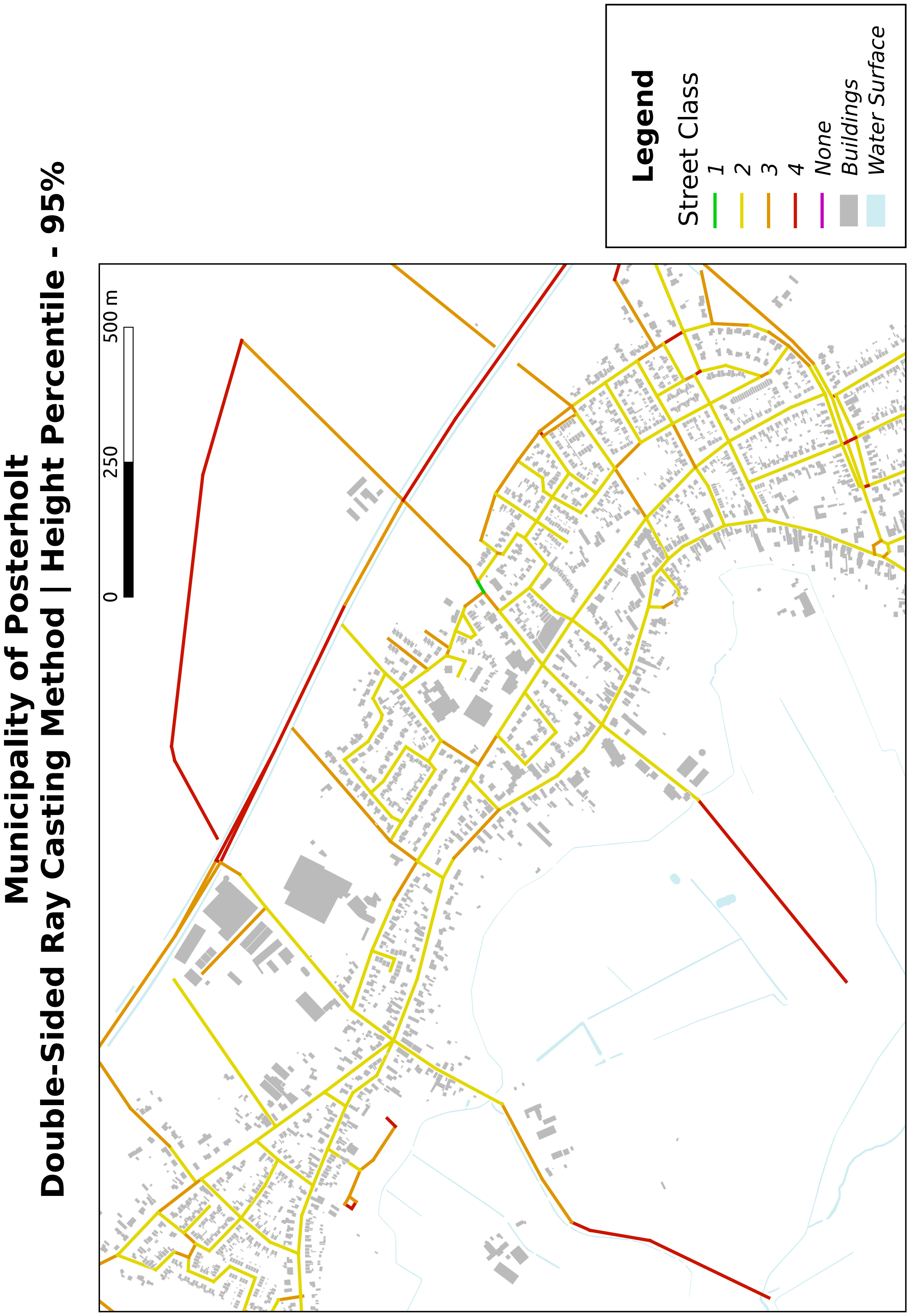

Municipality of Posterholt
Double-Sided Ray Casting Method | Height Percentile - 95%
0
250
500 m
Legend
Street Class
1
2
3
4
None
Buildings
Water Surface

Figure B.8: The result of the road classification algorithm for the city of Posterholt, Limburg.

C
Attempted implementations

This appendix will describe all the alternative methods that we implemented, but did not yield good classification results.

C.1. Buffer methods to associate buildings to roads

To associate buildings along the road segment, we implemented a buffer method, which limits the buildings to those within 60 meters from the road axis. This step is required to proceed with the identification of the first line of buildings facing the road, respecting RIVM's requirements [30]. The method we use is described in section 4.1; it requires roads segments to be split if they consist of more than two coordinates. In fact, road segments that present complex geometries introduce problems in the implementation of the buffer. Initially, we attempted to tackle them, leaving the geometry of the road unchanged. However, these implementations would require large computation efforts, while introducing uncertainties. The following sections describe three buffer approaches that did not provide a satisfactory outcome.

C.1.1. Shapely buffer function

The `Shapely buffer` function takes as input one geometry and outputs a polygon at a given distance to the input. The user can choose between many parameters, but cannot choose the buffer side [13]. In the current study, the distinction of buildings on the left and right side of the road is necessary. Thus firstly, we decided to use `Shapely parallel_offset` function. The function takes as input one `LineString` and returns a parallel geometry at a given distance. Later we created left and right buffer polygons using the road coordinates and the output `LineString` coordinates. However, for complex geometries the `Shapely parallel_offset` function would often output an `empty LineString`; for this reason, this method was discarded.

Secondly, we decided to create the buffer polygon for every road on both sides and to split it using the `Shapely split` function. The function takes as input two geometries and splits the first geometry by the second geometry. Then, it returns a collection of geometries representing the sub-parts [13]. However, the `Shapely split` function led to different problems. In a high number of cases, the buffer polygon would not be divided, as the road geometry was not recognised as a splitting geometry. This problem is caused by low floating point precision in `Shapely geometries`. We attempted to solve this problem by extending the road segment and by adding the road coordinates to the buffer polygon. Nevertheless, the issue could not be solved for every occasion. Thus, this implementation was abandoned.

C.1.2. Convex hull method

The convex hull method aims at tackling complex roads geometry issues, without splitting the road segment. The convex hull corresponds to the smallest convex polygon including all the points in a given set. If one road consists of more than two coordinates, sub-buffer polygons are created for each straight line segment and in both directions. Later, the convex hull of these polygons is created using the `Shapely convex_hull` function. The function takes as input any geometry, reads its coordinates and outputs the convex hull polygon.

This polygon should then be used as a buffer to detect buildings. However, this implementation lead to many particular cases. Complex road segments present different shapes; for example, they can be convex or concave, or present arches facing opposite sides. Thus, the convex hull polygon would need to be cut in correspondence to the road to avoid the inclusion of buildings from the wrong side. The process is simple for road segments made of three coordinates, but gets increasingly difficult and computational expensive for longer road segments. Furthermore, a few segments introduced additional issues, for example, a parallel offset could cross the road. For these reasons, we decided to not further use this implementation.

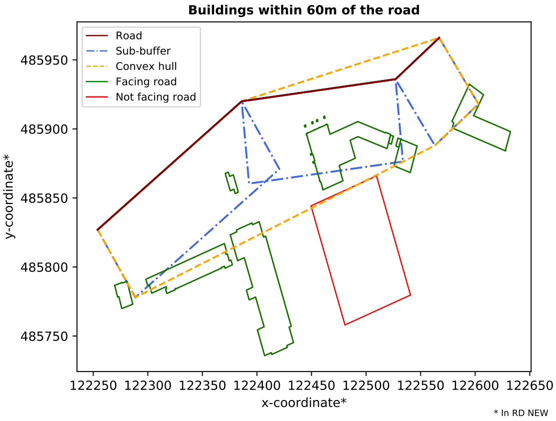


Figure C.1: The image shows the convex hull on the left of a complex road segment

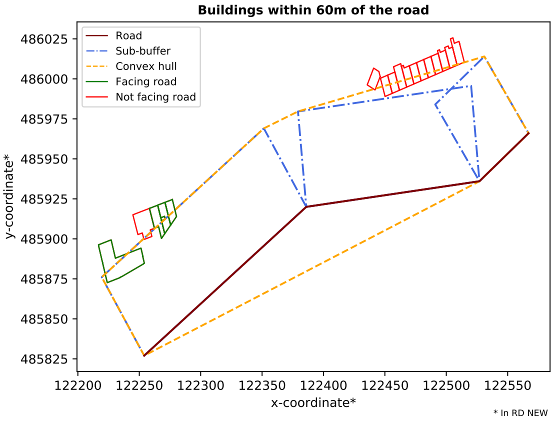


Figure C.2: The image shows the convex hull on the right side of a complex road segment

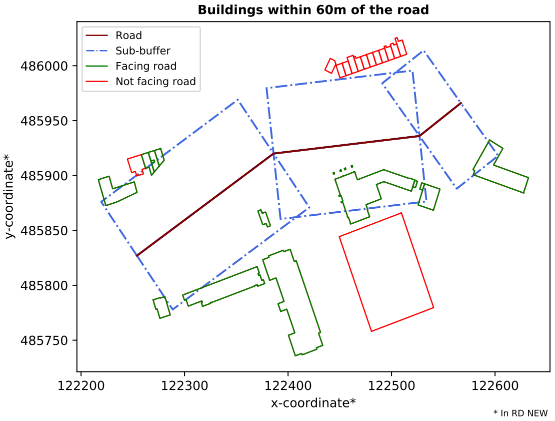


Figure C.3: The image shows the sub buffers created to compute the convex hull of a complex road segment

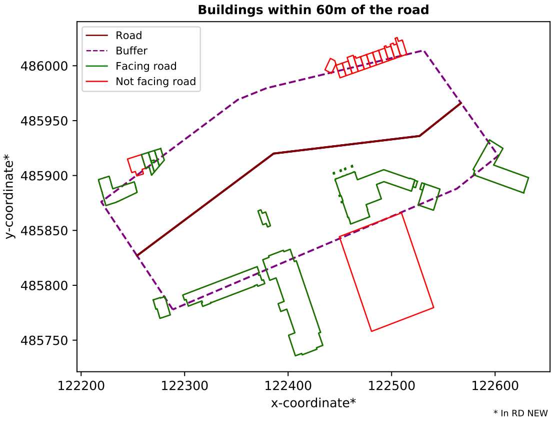


Figure C.4: The image shows the final buffers obtained from the convex hull, after cutting the wrong parts

C.1.3. Alpha shape method

The alpha shape method aims at solving the errors introduced with the convex hull buffer method. The alpha shape creates a concave hull around a finite set of input points, using an alpha parameter (α). “An edge of the alpha-shape is drawn between two members of the finite point set whenever there exists a generalised disk of radius $1/\alpha$ containing the entire point set and which has the property that the two points lie on its boundary.” [32]

We used the `alphashape` library for Python in our algorithm. However, we encountered many issues.

The alpha parameter has a strong influence on the outcome of the algorithm and it is not feasible to determine one value that works for the whole urban layout. The `alphashape` function allows the user make the alpha parameter flexible, as a consequence though the algorithm speed is seriously compromised. Thus, we discontinued investigating this method.

D

Measurement notes

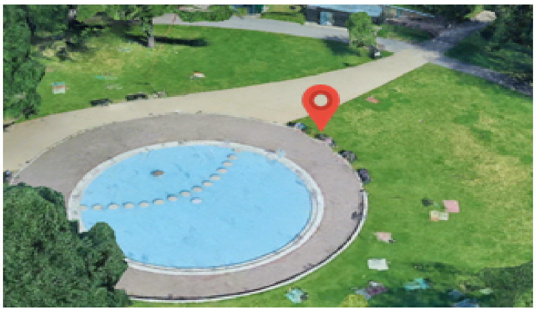


Figure D.1: The measuring location in Vondelpark.

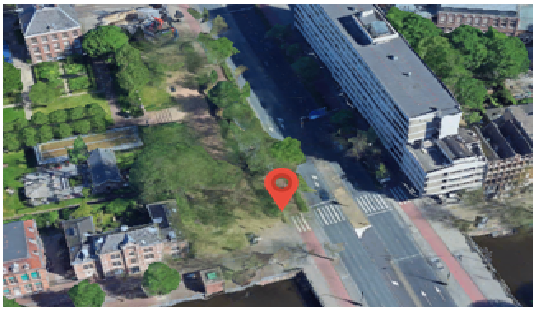


Figure D.2: The measuring locataion at Weesperstraat.

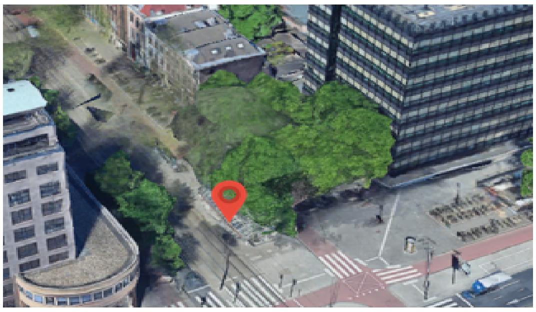


Figure D.3: The measuring location at Weesperplein.

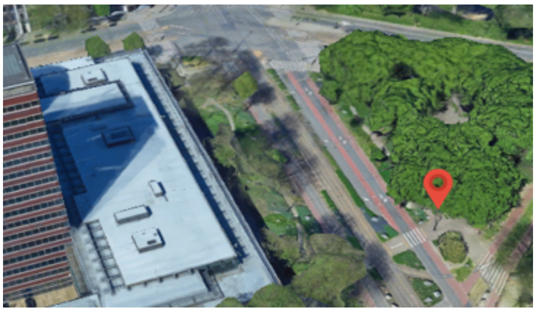


Figure D.4: The measuring location at Frederiksplein.

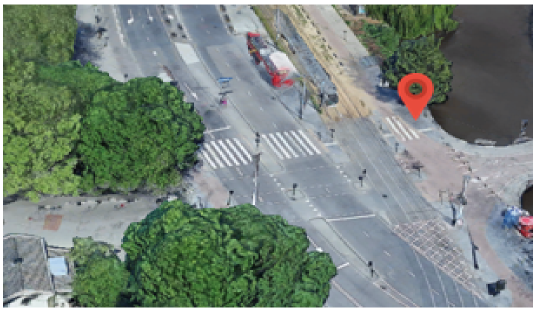
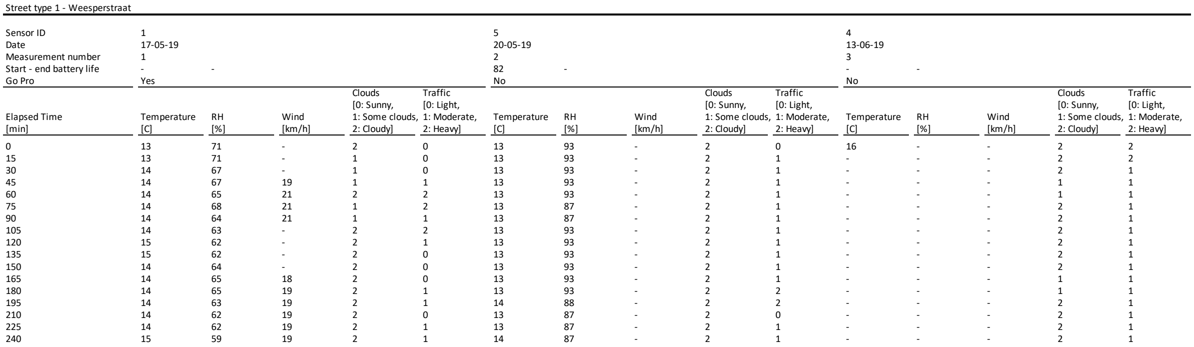


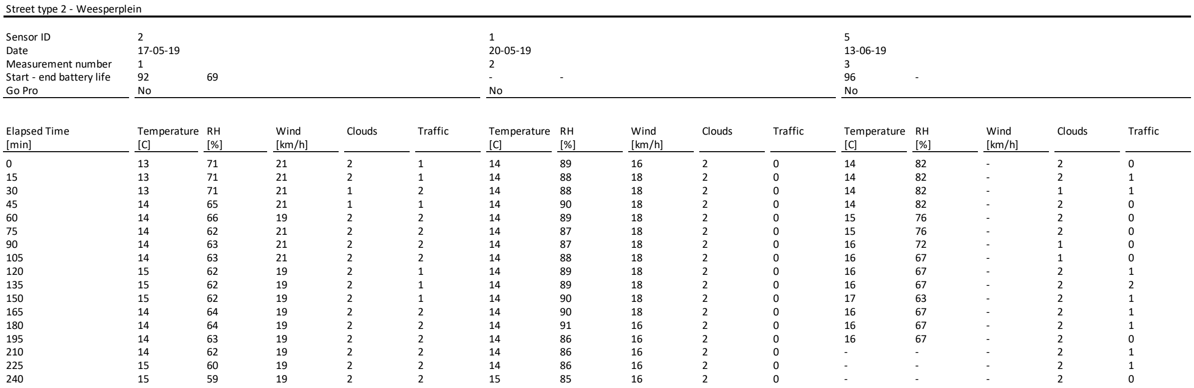
Figure D.5: The measuring location at Stadhouderskade.



Street type 1 - Weesperstraat

Sensor ID	1					5					4				
Date	17-05-19					20-05-19					13-06-19				
Measurement number	1					2					3				
Start - end battery life	-	-				82	-				-	-			
Go Pro	Yes					No					No				
Elapsed Time [min]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some clouds, 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some clouds, 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some clouds, 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]
0	13	71	-	2	0	13	93	-	2	0	16	-	-	2	2
15	13	71	-	1	0	13	93	-	2	1	-	-	-	2	2
30	14	67	-	1	0	13	93	-	2	1	-	-	-	2	1
45	14	67	19	1	1	13	93	-	2	1	-	-	-	1	1
60	14	65	21	2	2	13	93	-	2	1	-	-	-	1	1
75	14	68	21	1	2	13	87	-	2	1	-	-	-	2	1
90	14	64	21	1	1	13	87	-	2	1	-	-	-	2	1
105	14	63	-	2	2	13	93	-	2	1	-	-	-	2	1
120	15	62	-	2	1	13	93	-	2	1	-	-	-	2	1
135	15	62	-	2	0	13	93	-	2	1	-	-	-	2	1
150	14	64	-	2	0	13	93	-	2	1	-	-	-	2	1
165	14	65	18	2	0	13	93	-	2	1	-	-	-	1	1
180	14	65	19	2	1	13	93	-	2	2	-	-	-	1	1
195	14	63	19	2	1	14	88	-	2	2	-	-	-	2	1
210	14	62	19	2	0	13	87	-	2	0	-	-	-	2	1
225	14	62	19	2	1	13	87	-	2	1	-	-	-	2	1
240	15	59	19	2	1	14	87	-	2	1	-	-	-	2	1

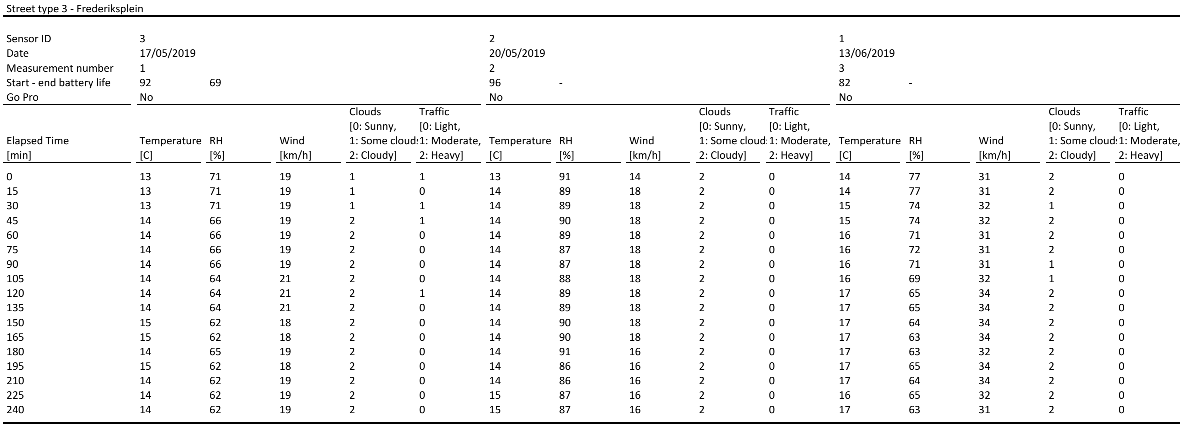
Figure D.6: The notes made during the three measurement days for street type 1 - Weesperstraat.



Street type 2 - Weesperplein

Sensor ID	2					1					5				
Date	17-05-19					20-05-19					13-06-19				
Measurement number	1					2					3				
Start - end battery life	92	69				-	-				96	-			
Go Pro	No					No					No				
Elapsed Time [min]	Temperature [C]	RH [%]	Wind [km/h]	Clouds	Traffic	Temperature [C]	RH [%]	Wind [km/h]	Clouds	Traffic	Temperature [C]	RH [%]	Wind [km/h]	Clouds	Traffic
0	13	71	21	2	1	14	89	16	2	0	14	82	-	2	0
15	13	71	21	2	1	14	88	18	2	0	14	82	-	2	1
30	13	71	21	1	2	14	88	18	2	0	14	82	-	1	1
45	14	65	21	1	1	14	90	18	2	0	14	82	-	2	0
60	14	66	19	2	2	14	89	18	2	0	15	76	-	2	0
75	14	62	21	2	2	14	87	18	2	0	15	76	-	2	0
90	14	63	21	2	2	14	87	18	2	0	16	72	-	1	0
105	14	63	21	2	2	14	88	18	2	0	16	67	-	1	0
120	15	62	19	2	1	14	89	18	2	0	16	67	-	2	1
135	15	62	19	2	1	14	89	18	2	0	16	67	-	2	2
150	15	62	19	2	1	14	90	18	2	0	17	63	-	2	1
165	14	64	19	2	2	14	90	18	2	0	16	67	-	2	1
180	14	64	19	2	2	14	91	16	2	0	16	67	-	2	1
195	14	63	19	2	2	14	86	16	2	0	16	67	-	2	0
210	14	62	19	2	2	14	86	16	2	0	-	-	-	2	1
225	15	60	19	2	2	14	86	16	2	0	-	-	-	2	1
240	15	59	19	2	2	15	85	16	2	0	-	-	-	2	0

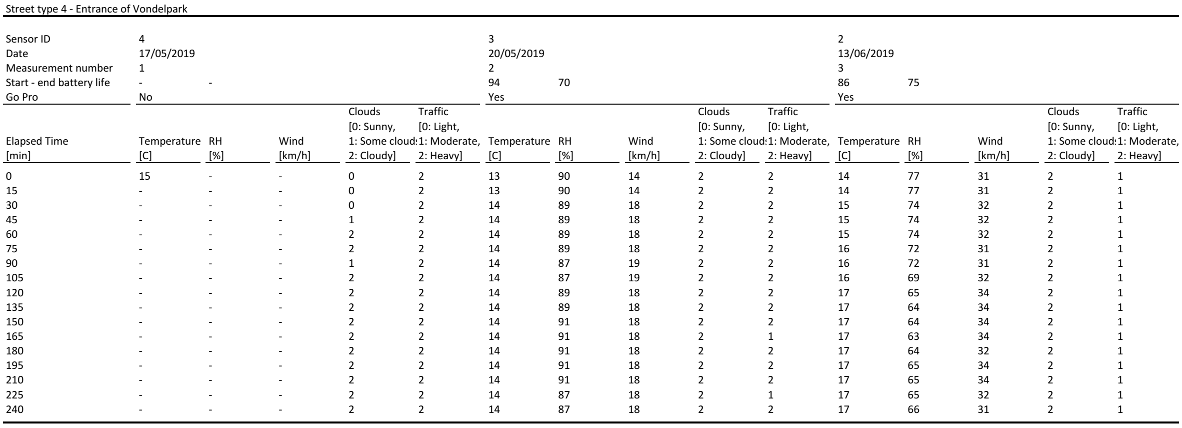
Figure D.7: The notes made during the three measurement days for street type 2 - Weesperplein.



Street type 3 - Frederiksplein

Sensor ID	3					2					1				
Date	17/05/2019					20/05/2019					13/06/2019				
Measurement number	1					2					3				
Start - end battery life	92	69				96	-				82	-			
Go Pro	No					No					No				
Elapsed Time [min]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some cloud 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some cloud 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some cloud 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]
0	13	71	19	1	1	13	91	14	2	0	14	77	31	2	0
15	13	71	19	1	0	14	89	18	2	0	14	77	31	2	0
30	13	71	19	1	1	14	89	18	2	0	15	74	32	1	0
45	14	66	19	2	1	14	90	18	2	0	15	74	32	2	0
60	14	66	19	2	0	14	89	18	2	0	16	71	31	2	0
75	14	66	19	2	0	14	87	18	2	0	16	72	31	2	0
90	14	66	19	2	0	14	87	18	2	0	16	71	31	1	0
105	14	64	21	2	0	14	88	18	2	0	16	69	32	1	0
120	14	64	21	2	1	14	89	18	2	0	17	65	34	2	0
135	14	64	21	2	0	14	89	18	2	0	17	65	34	2	0
150	15	62	18	2	0	14	90	18	2	0	17	64	34	2	0
165	15	62	18	2	0	14	90	18	2	0	17	63	34	2	0
180	14	65	19	2	0	14	91	16	2	0	17	63	32	2	0
195	15	62	18	2	0	14	86	16	2	0	17	65	34	2	0
210	14	62	19	2	0	14	86	16	2	0	17	64	34	2	0
225	14	62	19	2	0	15	87	16	2	0	16	65	32	2	0
240	14	62	19	2	0	15	87	16	2	0	17	63	31	2	0

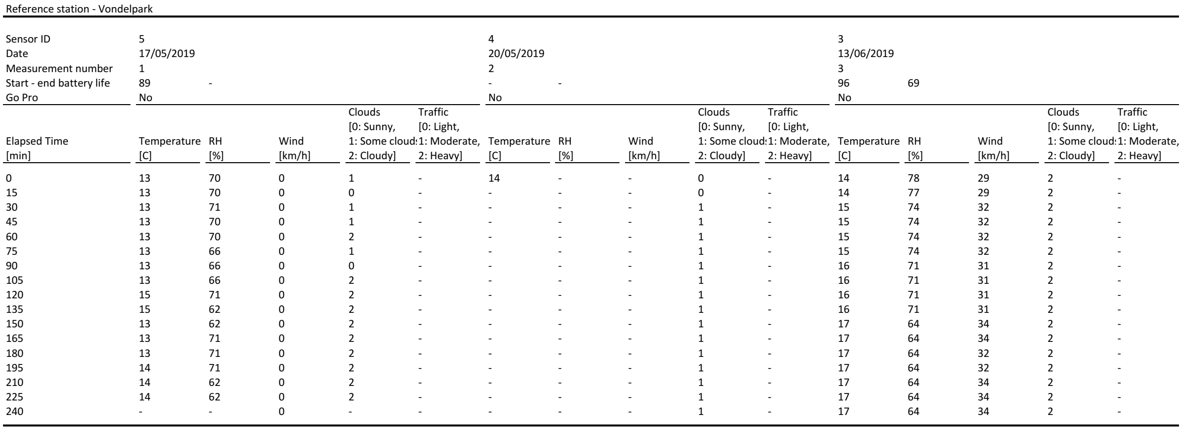
Figure D.8: The notes made during the three measurement days for street type 3 - Frederiksplein.



Street type 4 - Entrance of Vondelpark

Sensor ID	4					3					2				
Date	17/05/2019					20/05/2019					13/06/2019				
Measurement number	1					2					3				
Start - end battery life	-	-				94	70				86	75			
Go Pro	No					Yes					Yes				
Elapsed Time [min]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some cloud 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some cloud 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some cloud 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]
0	15	-	-	0	2	13	90	14	2	2	14	77	31	2	1
15	-	-	-	0	2	13	90	14	2	2	14	77	31	2	1
30	-	-	-	0	2	14	89	18	2	2	15	74	32	2	1
45	-	-	-	1	2	14	89	18	2	2	15	74	32	2	1
60	-	-	-	2	2	14	89	18	2	2	15	74	32	2	1
75	-	-	-	2	2	14	89	18	2	2	16	72	31	2	1
90	-	-	-	1	2	14	87	19	2	2	16	72	31	2	1
105	-	-	-	2	2	14	87	19	2	2	16	69	32	2	1
120	-	-	-	2	2	14	89	18	2	2	17	65	34	2	1
135	-	-	-	2	2	14	89	18	2	2	17	64	34	2	1
150	-	-	-	2	2	14	91	18	2	2	17	64	34	2	1
165	-	-	-	2	2	14	91	18	2	1	17	63	34	2	1
180	-	-	-	2	2	14	91	18	2	2	17	64	32	2	1
195	-	-	-	2	2	14	91	18	2	2	17	65	34	2	1
210	-	-	-	2	2	14	91	18	2	2	17	65	34	2	1
225	-	-	-	2	2	14	87	18	2	1	17	65	32	2	1
240	-	-	-	2	2	14	87	18	2	2	17	66	31	2	1

Figure D.9: The notes made during the three measurement days for street type 4 - Stadhouderskade.



Reference station - Vondelpark

Sensor ID	5					4					3				
Date	17/05/2019					20/05/2019					13/06/2019				
Measurement number	1					2					3				
Start - end battery life	89	-				-	-				96	69			
Go Pro	No					No					No				
Elapsed Time [min]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some cloud 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some cloud 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]	Temperature [C]	RH [%]	Wind [km/h]	Clouds [0: Sunny, 1: Some cloud 2: Cloudy]	Traffic [0: Light, 1: Moderate, 2: Heavy]
0	13	70	0	1	-	14	-	-	0	-	14	78	29	2	-
15	13	70	0	0	-	-	-	-	0	-	14	77	29	2	-
30	13	71	0	1	-	-	-	-	1	-	15	74	32	2	-
45	13	70	0	1	-	-	-	-	1	-	15	74	32	2	-
60	13	70	0	2	-	-	-	-	1	-	15	74	32	2	-
75	13	66	0	1	-	-	-	-	1	-	15	74	32	2	-
90	13	66	0	0	-	-	-	-	1	-	16	71	31	2	-
105	13	66	0	2	-	-	-	-	1	-	16	71	31	2	-
120	15	71	0	2	-	-	-	-	1	-	16	71	31	2	-
135	15	62	0	2	-	-	-	-	1	-	16	71	31	2	-
150	13	62	0	2	-	-	-	-	1	-	17	64	34	2	-
165	13	71	0	2	-	-	-	-	1	-	17	64	34	2	-
180	13	71	0	2	-	-	-	-	1	-	17	64	32	2	-
195	14	71	0	2	-	-	-	-	1	-	17	64	32	2	-
210	14	62	0	2	-	-	-	-	1	-	17	64	34	2	-
225	14	62	0	2	-	-	-	-	1	-	17	64	34	2	-
240	-	-	0	-	-	-	-	-	1	-	17	64	34	2	-

Figure D.10: The notes made during the three measurement days for Vondelpark.

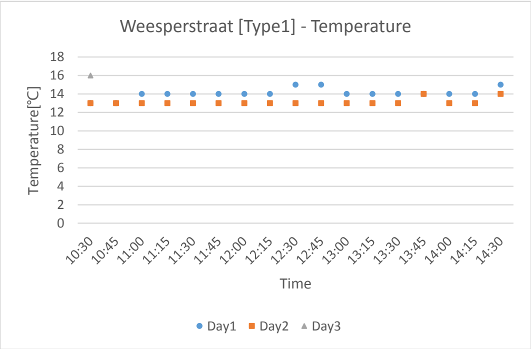


Figure D.11: The temperatures for street type 1.

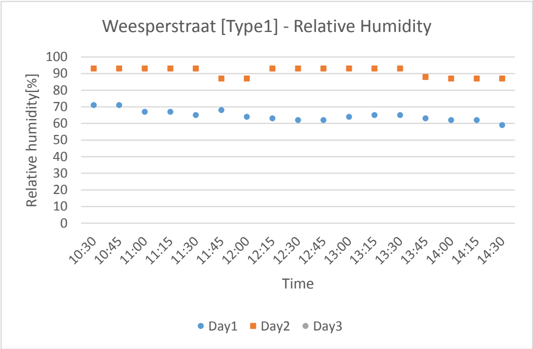


Figure D.12: The relative humidity for street type 1.

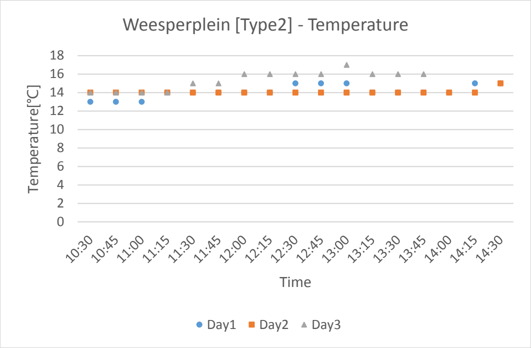


Figure D.13: The temperatures for street type 2.

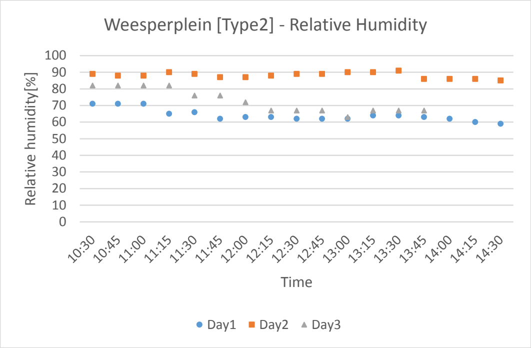


Figure D.14: The relative humidity for street type 2.

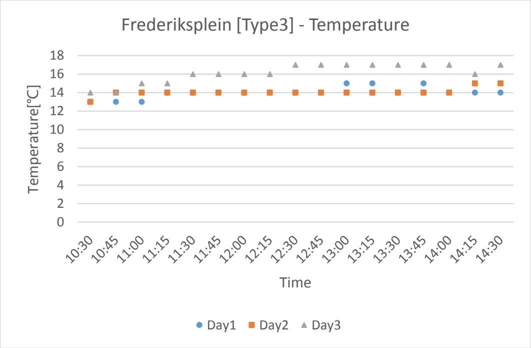


Figure D.15: The temperatures for type 3.

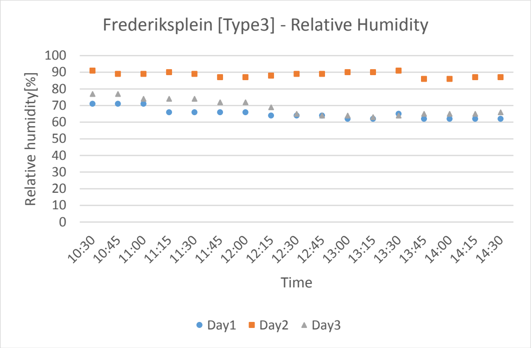


Figure D.16: The relative humidity for type 3.

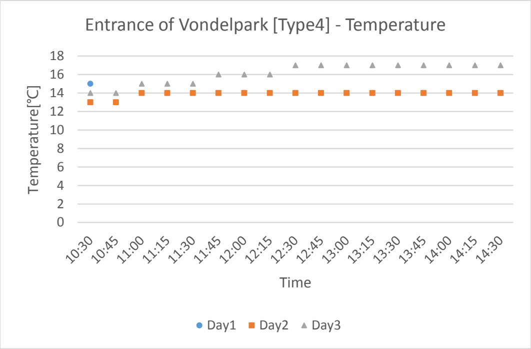


Figure D.17: The temperatures for street type 4.

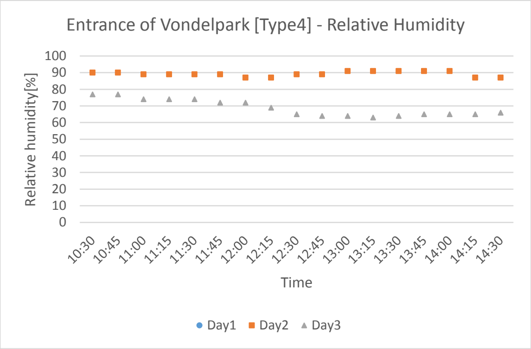


Figure D.18: The relative humidity street type 4.

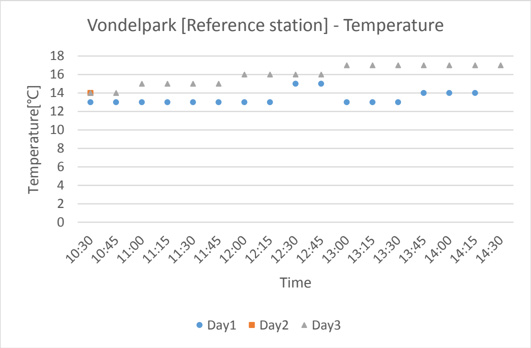


Figure D.19: The temperatures for Vondelpark.

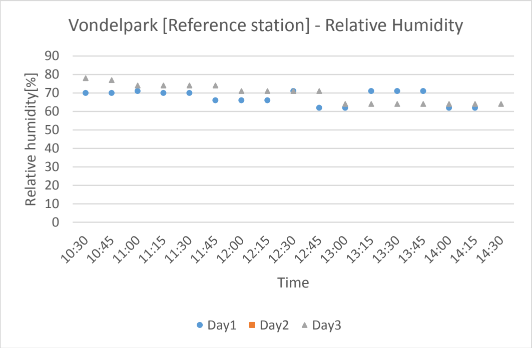


Figure D.20: The relative humidity for Vondelpark.

Bibliography

[1] European Environmental Agency. Air pollution sources. https://www.eea.europa.eu/themes/air/air-pollution-sources#tab-related-infographics. Accessed: 15-06-2019.

[2] European Environmental Agency. Every breath we take - Improving air quality in Europe. https://doi:10.2800/82831, 2013.

[3] United States Environmental Protection Agency. Nitrogen Oxides (NOx), Why and How They Are Controlled. 11 1999.

[4] Gemeente Amsterdam. Landelijk en Europees beleid. https://www.amsterdam.nl/parkeren-verkeer/luchtkwaliteit/landelijk-europees/#h4b254bc5-ce18-4404-8e85-0c611f109dfc. Accessed: 18-06-2019.

[5] Franz Aurenhammer and Rolf Klein. Voronoi Diagrams. https://pdfs.semanticscholar.org/b14c/a312a69498911556a03a5ae886a91191d774.pdf.

[6] European Commission. Air quality - Introduction. http://ec.europa.eu/environment/air/quality. Accessed: 15-06-2019.

[7] The European Commission. Commission refers POLAND to the Court of Justice of the EU over poor air quality. http://europa.eu/rapid/press-release_IP-15-6225_en.htm, 2015. Accessed: 18-06-2019.

[8] The European Commission. Air quality: Commission takes action to protect citizens from air pollution. http://europa.eu/rapid/press-release_IP-18-3450_en.htm, 2018. Accessed: 18-06-2019.

[9] 3D Geo-Information Group TU Delft. 3dfier. https://github.com/tudelft3d/3dfier. Accessed: 16-06-2019.

[10] Balász Dukai. 3D Registration of Buildings and Addresses (BAG). https://doi.org/10.4121/uuid:f1f9759d-024a-492a-b821-07014dd6131c, 2018. 4TU.Centre for Research Data. Dataset.

[11] Rijkswaterstaat Environment. Dutch policy and regulations for air quality. https://rwsenvironment.eu/subjects/air/air-quality/. Accessed: 16-06-2019.

[12] Python Software Foundation. About Python. https://www.python.org/about/. Accessed: 16-06-2019.

[13] Sean Gillies. The shapely user manual. https://shapely.readthedocs.io/en/stable/manual.html. Accessed: 20-06-2019.

[14] Max Roser Hannah Ritchie. Air pollution. https://ourworldindata.org/air-pollution. Accessed: 15-06-2019.

[15] Plume Labs. Flow, by Plume Labs | The First Smart Air Quality Tracker. https://plumelabs.com/en/flow/, . Accessed: 20-06-2019.

[16] Plume Labs. What you need to know about air pollutants. https://blog.plumelabs.com/2016/05/13/what-you-need-to-know-about-air-pollutants/, . Accessed: 20-06-2019.

[17] Plume Labs. Evaluation of Flow, a personal air quality sensor. 5 2019.

[18] Alexandre Leonardi and Boris Quennehen. Plume AQI: An Air Quality Index aligned with health recommendations. 5 2016.

[19] European Court of Auditors. Air pollution - Our health still insufficiently protected. `http://publications.europa.eu/webpub/eca/special-reports/air-quality-23-2018/en/#A1`. Accessed: 15-06-2019.

[20] World Health Organization. World Health Organization - About. `https://www.who.int/about`. Accessed: 16-06-2019.

[21] World Health Organization. Who air quality guidelines for particulate matter, ozone, nitrogen dioxide and solfur dioxide. `https://www.who.int/phe/health_topics/outdoorair/outdoorair_aqg/en/`, 2005.

[22] The European Parliament and the Council of the European Union. DIRECTIVE 2008/50/EC on ambient air quality and cleaner air for Europe. `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32008L0050&from=EN`, 2008. Accessed: 18-06-2019.

[23] PDOK. Publieke Dienstverlening Op de Kaart. `https://www.pdok.nl/`. Accessed: 17-06-2019.

[24] QGIS. Quantum GIS. `https://qgis.org/en/site/`. Accessed: 16-06-2019.

[25] Rijksoverheid. NSL Monitoringstool - Inleiding. `https://www.nsl-monitoring.nl/informatie/inleiding`, . Accessed: 18-06-2019.

[26] Rijksoverheid. Nieuwe omgevingswet maakt omgevingsrecht eenvoudiger. `https://www.rijksoverheid.nl/onderwerpen/omgevingswet/vernieuwing-omgevingsrecht`, . Accessed: 18-06-2019.

[27] Rijkswaterstaat. NSL-monitoringstool. `https://www.nsl-monitoring.nl/viewer/`, . Accessed: 16-06-2019.

[28] Kenniscentrum InfoMil Rijkswaterstaat. Nationaal Samenwerkingsprogramma Luchtkwaliteit (NSL). `https://www.infomil.nl/onderwerpen/lucht-water/luchtkwaliteit/regelgeving/wet-milieubeheer/nsl/`, . Accessed: 16-06-2019.

[29] RIVM. Toelichting. Digital scan, 1980s. Viewed: 19-06-2019. See Appendix for scanned document.

[30] Rijksinstituut voor Volksgesondheid en Milieu. Technisce beschrijving van standaardrekenmethode 1 (SRM-1). `https://www.rivm.nl/bibliotheek/rapporten/2014-0127.pdf`, 2015.

[31] Rijksinstituut voor Volksgezondheid. Luchtmeetnet. `https://www.luchtmeetnet.nl/stations`. Accessed: 16-06-2019.

[32] Wikipedia. Alpha shape. `https://en.wikipedia.org/wiki/Alpha_shape`. Accessed: 20-06-2019.