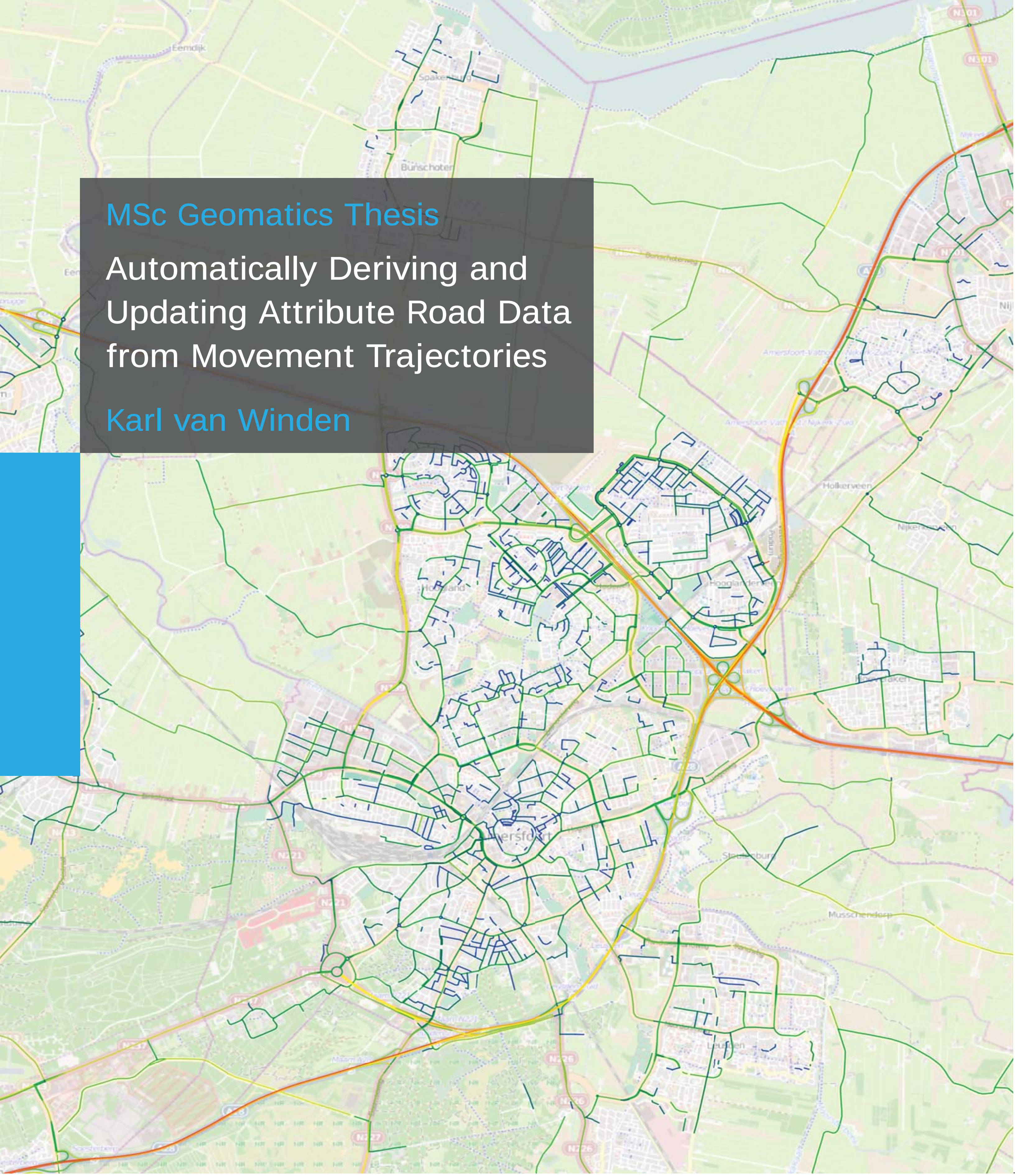
MSc Geomatics Thesis

Automatically Deriving and Updating Attribute Road Data from Movement Trajectories

Karl van Winden



Delft University of Technology

Challenge the future

Automatically Deriving and Updating Attribute Road Data from Movement Trajectories

M.Sc. Geomatics Thesis
19-06-2014

By

Karl van Winden

Delft University of Technology

van Winden, K. Automatically Deriving and Updating Attribute Road Data from Movement Trajectories. *MSc Thesis in Geomatics,* Delft University of Technology, June 2014.

Section GIS Technology
OTB Department for the Built Environment
Faculty of Architecture
Delft University of Technology
The Netherlands

Main Tutor
Ir. Filip Biljecki

Graduation Professor
Dr. ir. S.C. (Stefan) van der Spek

Tutor
Dr. D.M.J. (David) Tax

Delegate of the Board of Examiners
Drs. D.J. (Dirk) Dubbeling

Cover Illustration
The front cover illustrates the value of the average speed attribute in the area of the city of Amersfoort, the Netherlands projected onto OpenStreetMap. The average speed was calculated for the roads containing more than 10 GPS points.
The back cover illustrates the value of the same attribute for a larger area around Amersfoort.

Abstract

There are many applications that use maps, and more detailed the maps are, more applications can benefit from the map. Many maps are still manually created and also the underlying attributes are informed in a manual process. This thesis presents a method to automatically derive and update attribute road data by mining and analyzing movement trajectories.

The method used for this thesis implemented OpenStreetMap as the underlying map which will be updated. GPS tracks are the movement trajectories that will be used to derive the information for the road attributes. There are attributes that are already conceptually present in OpenStreetMap but are in practice rarely filled. The attributes that will be investigated to derive are: whether the road is a one or two way road, the speed limit of the road, the number of lanes of the road and which vehicles have access to the road. Also, new attributes are introduced in this thesis. These are the average speed of the road, the hours in which the road is congested, the importance of the road and whether the road has a certain geometrical error.

Preprocessing is performed before the attributes can be derived. Important are the classification of the transportation mode of the GPS tracks and the map matching of the GPS points to the roads they are on. When the IDs of the roads, where the GPS points are on, are known the attribute extraction algorithms can be applied. These algorithms all have different methods for deriving their attributes. There are attributes that use the speed of the GPS point, the distance from the point to the road or the heading of the point. For some attributes, a hierarchical code list is created to provide different perspectives on the error of the attributes. The code list consists of the values of the attributes and the hierarchy between these lists describes the level of detail and the granularity of the values.

While some attributes were classified correctly in almost 100% of the cases, the extraction of the attributes were not all successful. The number of lanes proved to be too difficult to derive out of the available data and the importance of a road relies on a complete coverage of data which was not the case. Although, the latter is applied in this research. The other attributes had different results, the accuracy of the classification of the speed limit was 69,2%. However, when taking into account speed limits that are only one step away (e.g. 60 km/h instead of the classified 50 km/h) the classification increases to 95%. The classification of the roads that allow bicycles was 74% and the attribute to determine whether a road is a one or two way road has a classification accuracy of 99%.

In the future, the attribute extraction algorithms could be improved or expanded. More detailed levels in the hierarchical code list could be added and constraints could be added to improve the attributes. Also, some techniques might be enhanced for better results. Finally, the ideal application of this method would be deriving and updating the attributes in real-time. This could lead to live maps which would change real-time with the changes on the road.

Contents

Abbreviations

AIC	Akaike Information Criterion
ANN	Artificial Neural Network
API	Application Programming Interface
BIC	Bayesian Information Criterion
CDF	Cumulative Distribution Function
CDOP	Correction Dilution of Precision
CRS	Coordinate Reference System
CSV	Comma-Separated Values
DGNSS	Differential Global Navigation Satellite System
DTM	Digital Terrain Model
DR	Deduced Reckoning
EM	Expectation Maximization
GIS	Geographical Information System
GLONASS	Global Navigation Satellite System
GMM	Gaussian Mixture Model
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPX	GPS Exchange Format
HMM	Hidden Markov Model

MNN	Modular Neural Network
ODbL	Open Data Commons Open Database License
OGC	Open Geospatial Consortium
OSM	OpenStreetMap
PDF	Probability Density Function
RD	Rijksdriehoek (Dutch Coordinate Reference System)
REST	Representational State Transfer
SVC	Support Vector Classifier
SVM	Support Vector Machines
TOA	Time of Arrival
UML	Unified Modeling Language
VCR	Velocity Change Rate
VGI	Volunteered Geographic Information

CHAPTER 1 Introduction

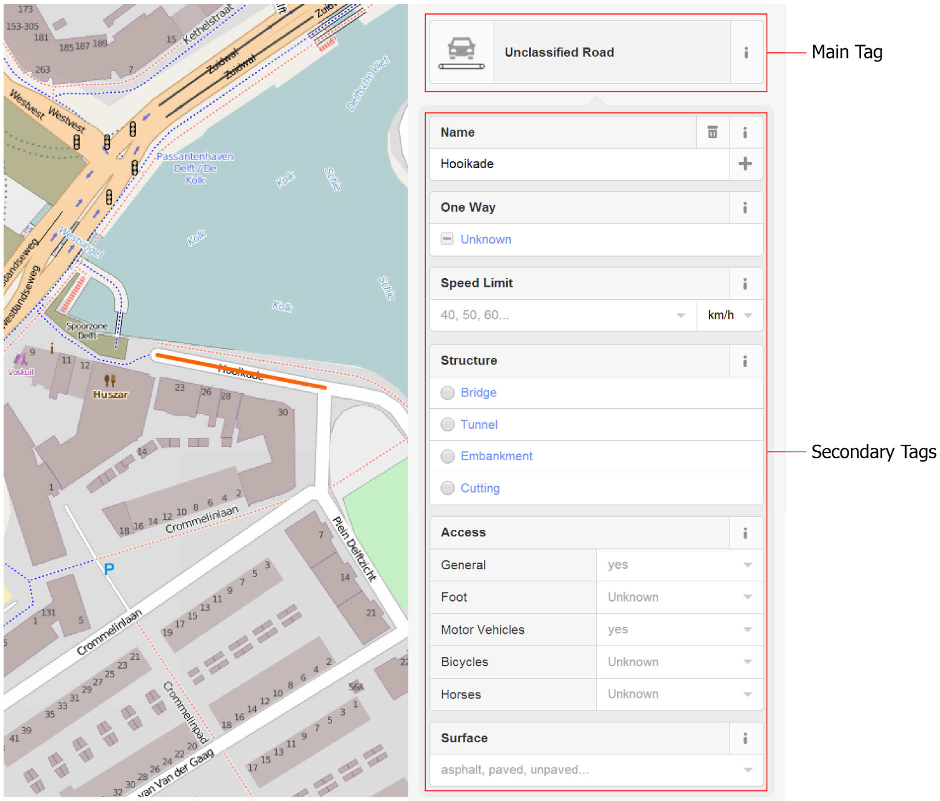
1.1 Motivation

Nowadays, there are many applications which make use of maps. Therefore there are different demands about the detail of the map and the information that is stored in the map. Still, much work is done manually or semi-automatically by map makers. This results in research of automated methods for map making. This kind of research mostly focus on the visualization aspect, rather than the information behind the maps. Much of the data is still acquired manually and little research is performed on the automatic extraction of attributes for maps.

With new technologies like smart phones, more and more movement data becomes available. Still, more can be done with this data. This research investigates if movement trajectories can be used for automatically deriving and updating attribute road data. Automatically deriving this kind of information saves a vast amount of time and money compared to manual extraction. The results could be used for various applications as navigation systems, traffic analyses, urban development and civil engineering and more. This research uses OpenStreetMap as a base map due to its free usage of data and Global Positioning System (GPS) tracks as movement trajectories because of its availability and usability.

The concept of Volunteered Geographic Information (VGI) has recently changed the mapping process. OpenStreetMap (OSM) is the most significant example of a system based on VGI (Girres and Touya, 2010). The aim of OSM is "to create a free digital map of the world and is implemented through the engagement of participants in a mode similar to software development in Open Source projects" (Haklay, 2010). OSM follows the same model as Wikipedia, i.e. the map should be free to use, editable and licensed under new copyright schemes (Haklay, 2010). When the community, or crowd, contributes in one way or another to solving a problem or performing a task specified by a requester, this can be called crowdsourcing (Nakatsu and Iacovou, 2014). OSM is an example of crowdsourcing, but the acquisition of movement trajectories could also benefit from crowdsourcing.

Research on the quality of OSM data in the United Kingdom shows that the data has a reasonable accuracy of about 6 m and a coverage of 100% for Ordnance Survey digitised motorways, A-roads and B-roads (Haklay and Weber, 2008). However, the attribute accuracy of OSM data often lacks. Attributes in OSM are described using tags, each tag describes an attribute of the feature. OSM has a freeform tagging system with key-value pairs, which allows the contributors to add an unlimited number of attributes to a feature (Haklay and Weber, 2008). There are, however, certain tags which act as informal standards agreed upon by the OSM community. The attributes



Unclassified Road

Name	
Hooikade	
One Way	
▭ Unknown	
Speed Limit	
40, 50, 60...	km/h
Structure	
○ Bridge	
○ Tunnel	
○ Embankment	
○ Cutting	
Access	
General	yes
Foot	Unknown
Motor Vehicles	yes
Bicycles	Unknown
Horses	Unknown
Surface	
asphalt, paved, unpaved...	

```
▼<way id="39941510" visible="true" version="4" changeset="12477035"
 timestamp="2012-07-24T21:33:59Z" user="datalogg" uid="105512">
   <nd ref="479577554"/>
   <nd ref="1391531638"/>
   <tag k="highway" v="unclassified"/>
   <tag k="name" v="Hooikade"/>
```

Figure 1.1: Missing OpenStreetMap attribute data for a road (orange line) showing its map, attributes and corresponding XML. Courtesy of OpenStreetMap (2013a)

are divided in main tags and secondary tags. The main tag describes the most important attribute and the secondary tags provide additional attributes. Quantitative results for the French OSM dataset shows that the main tag for roads which describes the type of the road is informed at 85% of all the roads (for other types this is mostly more), but the secondary attributes (e.g. number of lanes and speed limit in the case of roads) are rarely informed (Girres and Touya, 2010). This can be seen in Figure 1.1, the Figure shows the information of a road in OSM consisting of the main tag, which describes the type of the road, and the most used secondary tags. The secondary attributes can add valuable information to a number of applications, e.g. when the OSM data is used for spatial analyses or if OSM is used for navigation applications. These attributes are probably missing due to the unwillingness of the person adding the data or simply because the attributes are not known. However, some attributes such as maximum speed and one or two way direction may be derived automatically from the GPS tracks added by the user. This research will investigate the possibility of deriving missing OSM attribute road data from GPS tracks and updating OSM with these attributes. The research tries to implement this on real GPS tracks.

VGI relies on the input of people and the willingness of people to provide information to other people. This is also the case for the attributes, however not many people are willing to manually provide this information for the roads. Therefore, only a small amount of roads have the attributes informed. This could be much easier using the movement trajectories of people and an automated method to derive the attributes from these trajectories. This is the strength of this research. When people get more involved in the map making process by only providing their movement trajectories, the accuracy and detail of the map could improve significantly.

In the past, no research addressing this problem have come to the attention. There is, however, some similar research performed on the enrichment of data using GPS traces or OSM data. These research projects can be placed in the spatial data mining discipline in which different spatial data mining techniques can be distinguished. The definition of data mining according to Giannotti et al. (2008) is "the methods for extracting models and patterns from (large) volumes of data". Spatial data mining can then be described as extracting certain models and patterns from mass data that contain spatiotemporal aspects, such as GPS trajectories. The real development of data mining for spatiotemporal data started just recently, around 2007 (Giannotti et al., 2008). This field has much to offer, but the fact that there is a lack of related work for this project exposes that there still is a lot to be done.

1.2 Relevance and Contribution

The research focuses on automatically deriving attribute road data from movement trajectories. This Section discusses the relevance and the value of the research. What is new about this research? What does it contribute? And for what can we use it? All these questions are answered below.

As stated in Section 1.1, there are not many similar research projects that have been performed in the past. This is the case for extracting attribute information from movement trajectories, but also for improving maps in an automated way. For OSM, this is partially due to the unwillingness of the OSM community for automated methods. Most OSM contributors see updating OSM as a hobby and spend a great deal of time to improve the map in their way. Introducing automated methods will perhaps change their maps, in which they put much time. However, the method presented in this thesis updates only missing attributes and does not change the geometry in the map. Basically, it complements the data by adding/updating missing attributes. Thus it can be stated that the attribute extraction as well as the updating of OSM is novel and introduces a new method to enrich the OSM data from GPS tracks in an automated manner.

Now that the novelty of this research is stated, the contribution of the results can be discussed. Important to know is the rise of open data in the past few years. Open data is defined as free to use data or data that does not exceed the costs of reproduction. Not only the data is free or at the cost of reproduction, but everyone can also use, re-use the data and it is free from rights (Van Loenen, 2003). OSM is also open data, licensed under Open Data Commons Open Database License (ODbL) (OpenStreetMap, 2014). Nowadays, open data is especially popular for governmental data. This is due to the financial benefits open data has, although it may sound unlikely. When selling data at a price exceeding the costs of reproduction and with a strict copyright license, only primary users will use the data. Primary users are the users that frequently use the data for the goal that it was meant by the producer (van Loenen and Crompvoets). When providing the data for free or at the cost of reproduction and without strict copyright licenses, the data will be used by primary, secondary and tertiary users. Secondary users only use the data incidentally, but the tertiary users add value to the product and provides the data to the end-user. In this way more people will benefit from the data, making it possible to add value to the data and create new products. In fact "lowering the price of public sector geographic data by 60% would lead to a 40% annual turnover growth" (Pluijmers and Weiss, 2002). These facts also explain the importance of OpenStreetMap in the map market. Alternatives for OSM are either available at prices exceeding the costs of reproduction, e.g. HERE Maps, and/or the raw data is not available to the public, e.g. Google Maps. Alternatives like HERE Maps are, of course, much more detailed in certain ways, having more resources to produce the map. HERE Maps has one of the most detailed maps of the world and provides this data to clients who demand this quality. However due to the costs, this map is not attractive to tertiary users. Therefore OSM is an important source for tertiary users. Tertiary users can use OSM freely to add value to the map, like cycling navigation applications for smartphones. This research contributes to the level of detail of OSM and enriches the data to become more complete. Therefore OSM can be used for more and different purposes. Possible attributes that can be extracted are for example average speed and maximum speed which can contribute to navigation applications. Thus a more complete map/dataset can lead to more applications, which means more users. This research focuses on OSM and GPS tracks, but the final methods could also be used on different maps and with different data types. The companies behind other maps could also benefit from the result, since automated extraction is cheaper than manual updating attribute road data which is nowadays done by driving through the entire country and taking pictures every 5 meters.

1.3 Objectives

The objective of this research is to develop a method that automatically derives and enriches attribute road data from movement trajectories. Therefore the research question of this project is:

- Is it possible to derive and update attribute road data by using movement trajectories and to what extent?

To answer this research questions, additional research has to be performed. The research sub questions below facilitate answering the main question:

- Which road attributes in OpenStreetMap data can be enriched by GPS tracks (movement trajectories)?
- Can GPS tracks provide information for new attributes to OpenStreetMap data?
- What is needed to extract the information for the missing attributes?
- How can the method automatically update OpenStreetMap?
- How can the results be validated?

1.4 Scope and Outline

This research could be conducted on different maps and with different inputs and outputs. However, for this research a choice has been made for OpenStreetMap as the updated map, GPS tracks as the input and speed, direction, access and number of lanes as the output together with some new attributes. This Section will elaborate more on why these decisions have been made and also defines the scope of the research. Finally, it will conclude with the goal off this thesis.

There are many different maps available on the internet, of which Google Maps (Google) is most known. Unfortunately, the data behind Google Maps is not available to the public. OSM data is available to the public, but this is not the only reason OSM is chosen for this thesis. OSM has 1,5 million users worldwide, who are contributing to the map or using it. These users uploaded almost 4 billion GPS points, resulting in more than 2 billion nodes and over 200 million ways (OpenStreetMap, 2013b). These nodes and ways construct the map of the world, showing different features like different amenities, buildings and highways. It should be noted that OSM does not make use of Open Geospatial Consortium (OGC) standards (OpenStreetMap, 2013b). Object descriptions differ therefore significantly from other GIS applications. There are a great deal of features, which are all making OSM one of the most detailed maps. Therefore more and more people are using OSM and the more detailed it becomes, the higher the increase of users and contributors will become. With its increasing popularity, this research can contribute to OSM to make it more detailed and accurate.

There are many different types of movement trajectories. There are satellite based movement trajectories such as United States' GPS, Russia's Global Navigation Satellite System (GLONASS), Europe's Galileo and China's Beidou. These systems all fall under the Global Navigation Satellite Systems (GNSS). A different, non-satellite based movement trajectory can be for example trilateration of mobile devices. This thesis will investigate the possibility of using GPS tracks to derive attributes for roads. GPS tracks are chosen because GPS is the most commonly used movement trajectory and it is the original and most used way for updating OSM. Roads have been chosen, because the movement trajectories usually follow roads. People tend to move over roads and therefore most information can be derived for the roads people are on. Therefore, the final algorithm of this thesis will be most used by contributors of OSM. For the scope of this thesis (8 months), it is not feasible to address all features and tags in OSM. Next to that, it is also not possible to derive all information only from the GPS tracks. Therefore this research needs to be narrowed down to only certain features and tags.

It makes sense that GPS tracks of people travelling have a direct link to the road network in OSM. Therefore it will be most feasible to direct this research to the "highway" feature of OSM. "The highway tag is the main tag used for identifying any kind of road, street or path." (OpenStreetMap, 2013b). Not only does the "highway" feature match the GPS tracks, it is also the most updated and most used feature in OSM (Girres and Touya, 2010).

There are multiple attributes for the "highway" feature, but not all of these attributes can be updated by using GPS tracks. Therefore this research will investigate the possibility of deriving the information of four attributes, of which three of them are listed in the useful combinations by OSM for "highway" (OpenStreetMap, 2013b). These three attributes are "maxspeed", "oneway" and "access". The "maxspeed" tag provides information about the maximum speed that is allowed on the road. Currently 26,3% of all the OSM highways in the Netherlands have a filled in "maxspeed" tag. It should be noted that not all of these tags are correct and that the way of tagging varies, e.g. only the number 30 or 30mph or 30 mph. By updating this data automatically these errors cannot be made anymore. The "oneway" tag is used to indicate the access restriction on the road, describing either a one way road or a two way road. This tag occurs 13,9% of the time in the OSM road network. The "access" tag provides information on the vehicles that are allowed on the road, e.g. car, bus, bicycle and pedestrian. The fourth attribute is "lanes", describing the number of lanes of the "highway".

Finally, the goal of this research is to research the possibilities of automatically deriving and updating the four existing attributes in OSM using GPS tracks in combination with the OSM data and creating a software prototype/algorithm to do so. Also the possibility of adding new attributes to the methods will be investigated. The attributes for roads that will be investigated in this thesis are:

- one or two way road
- speed limit
- number of lanes
- access of type of vehicles
- average speed
- hours of congestion
- importance
- geometrical error

This thesis will first address some related work in Chapter 2. In Chapter 3 the methodology of this research will be provided. Chapter 4 discusses and explains the preprocessing of the data. In Chapter 5 the extraction of the different attributes is investigated and developed after which the results are validated in Chapter 6. Finally, Chapter 7 presents the conclusions and provides ideas for future work.

CHAPTER 2
Related Work

In order to automatically derive and update OSM attributes, closely related domains in the information science should be investigated. First is the domain of data mining as mentioned in the Introduction. To extract these models and patterns from large volumes of data, two domains are important: statistics and pattern recognition. Finally, the last Section will provide related work to the topic of this research.

2.1 Data Mining, Pattern Recognition and Statistics

This Section will define the three domains of data mining, pattern recognition and statistics and discusses some of the techniques that are commonly used.

2.1.1 Data Mining

The past years there is a vast increase in the amount of data. Datasets became bigger and bigger and cannot be analysed by humans anymore, because of the time and costs that would take. This causes the domain of data mining to grow rapidly in the past years. Data mining can be applied to various fields, like: scientific data, medical data, demographic data, financial data and marketing data (Han et al., 2006). For this research, the focus is on the field of spatial or geographic data. The difference between the terms spatial and geographic is that:

> "Spatial" concerns any phenomena where the data objects can be embedded within some formal space that generates implicit relationships among the objects of which geographic is the specific case where the data is georeferenced and relates to a location or position on or near the Earth's surface. "Geographic" refers to the specific case where the data objects are georeferenced and the embedding space relates (at least conceptually) to locations on or near the Earth's surface. (Miller, 2008)

Spatial data mining focuses on extracting models and patterns from mass data that contain spatiotemporal aspects. GPS tracks contain both the spatial aspect (x, y, z) and the temporal aspect (time). The book "Mobility, Data Mining and Privacy" (Giannotti et al., 2008) provides an overview on movement data, spatial data mining and concerning privacy issues. In this thesis, privacy issues will be discussed in Chapter 3.4 and movement data or GPS data will be discussed in Subsection 3.1.1.

Spatial data mining can be used for deriving knowledge from data using models and patterns.

This is not the only way to derive knowledge from spatial data, one can also extract spatial features from the data. Spatial features describe a characteristic of a geographic object. This object has a spatial component in the form of a position, but also non-spatial attributes like maximum speed, lanes and oneway direction for example. Non-spatial attributes can be treated like normal information data, where the spatial features should be extracted differently. Spatial feature extraction can be divided into two different groups: relational features and aggregation-based features (Giannotti et al., 2008).

Relational features describe, as its name implies, the relation between the spatial components between two or more different geographic objects. The main example of a relational feature is distance. More relational features can be derived using topological relations and directional relations. Topological relations can be extracted using the 9-intersection model. This model is a 3x3 matrix representing the relations between the interior, boundary and the exterior of a geographic object. The matrix can define 8 topological relations: disjoint, contains, inside, equal, meet, covers, coveredBy and overlap (Clementini et al., 1994). Directional relations define the relation between a point and its direction to another point. Directional relations can be expressed using the cardinal directions north (N), east (E), south (S), west (W) and its combinations, e.g. north-east (NE) or south-south-west (SSW), according to the cone-shaped areas. When using the projection-based method, two orthogonal projections from the first object divide the space in nine partitions making it possible to derive the direction to the second object.

Aggregation-based features describe the aggregation of data inside a certain area of which it is not possible to assign a single location, e.g. birth rate data. Determining the spatial resolution is important for this feature, since a too large resolution results in a single result where a too small resolution separates all the data in a way no pattern can be discovered anymore.

To improve spatial feature extraction, one can use background knowledge. In the case of this thesis it is possible to check whether a road is located on land or on water. If it is located on water it can be stated that it is probably a bridge or a tunnel depending on the height of the road.

The spatial data mining domain consists of different techniques to derive patterns from mass data. The techniques that will be discussed here are clustering, classification and regression, association rules and subgroup discovery.

"Clustering divides a given set of objects into non-overlapping groups, such that similar objects are within the same group and objects of different groups are most heterogeneous." (Giannotti et al., 2008). For geographic objects, this means that objects that are in the neighbourhood of each other can be clustered. Objects are thus divided into groups.

A different technique is classification and regression. This technique predicts unknown target values of objects using a set of training data. It is, like clustering, an important part of the pattern recognition domain. Therefore details of these two techniques will be discussed in Subsection 2.1.2.

The third technique is association rules. Association rules over relations are of the form $X \rightarrow Y$. X and Y represent the sets of items that co-occur in the dataset. If the rule is exact, it can be stated that all tuples in a dataset that satisfy X also satisfy Y. The confidence factor is then 1. But if it is lower it means that all tuples that satisfy X, not always satisfy Y (Miller and Yang). For spatial association rules the X can be a spatial predicate closeToRoad that is calculated by the spatial relationship distance. The rule should meet a certain threshold regarding the confidence region to be reliable.

Finally, the fourth technique is subgroup discovery. It tries to detect subgroups in the whole dataset that deviate significantly regarding a target value. Basically, this technique discovers the subgroups that are statistically the most interesting. This can be done by finding the groups that have a significantly high or low share with respect to the target value, thus deviates the most from it. (Herrera et al., 2011)

2.1.2 Pattern Recognition

Pattern recognition and data mining are closely related. "Pattern recognition is the scientific discipline whose goal is the classification of objects into a number of categories or classes." (Theodoridis and Koutroumbas, 2008). These objects, or patterns, can vary from images to any other type of measurements. Data mining and knowledge discovery is one of the key application areas of pattern recognition. Pattern recognition is commonly used for classification and clustering. This Subsection will elaborate more on these two methods, which are commonly used in the data mining domain as mentioned in Subsection 2.1.1. It should be noted that pattern recognition divides user cases into two categories: supervised and unsupervised cases. In the first case there is a set of training data available with known class labels, which can be used to learn classifiers in a supervised way. In the second case, this kind of training data is not available and the goal is to reveal the underlying similarities of the data and cluster the data that is similar to each other. This is known as unsupervised pattern recognition. Figure 2.1 depicts the basic stages that are involved in the design of a classification system. The sensor provides the measurements for the classification, this can be satellite imagery, handwritten digits, etc. From these measurements, features can be generated and selected. Features are basically the measurements that are used for the classification. The entire set of features form the feature vector. There are different features that can be used from a set of measurements. E.g. features of an image can be only the pixels, but it can also be the mean and standard deviation of the intensity of the pixels. After the best features are generated and selected, one can design the corresponding classifier.

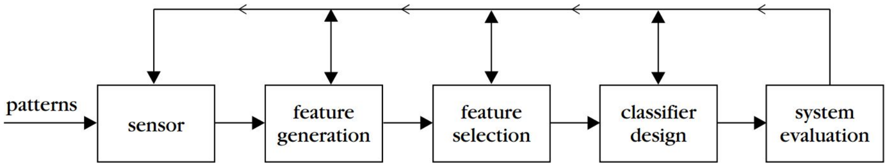


Figure 2.1: The basic stages involved in the design of a classification system. Courtesy of Theodoridis and Koutroumbas (2008)

The goal of classification for supervised cases is to identify the input as a member of a predefined class. For unsupervised cases it is to assign the input data to, till then, unknown classes (Jain et al., 2000). To do this, there are several methods for both the supervised as the unsupervised cases.

Supervised classifiers can be divided into three main groups: classifiers based on Bayes decision theory, linear classifiers and nonlinear classifiers.

The first group of classifiers is based on the Bayes rule. The Bayes rule uses the a priori probabilities, the class-conditional probability density functions and the probability density function of the pattern to calculate the posterior probability. The Bayes classification rule can then be defined as the class with the highest posterior probability can be assigned to the object. "The Bayesian classifier is optimal with respect to minimizing the classification error probability" (Theodoridis and

Koutroumbas, 2008). This means that when everything is known, the Bayes classifier would always have the lowest possible error. Classifiers that can be grouped under the probabilistic classifiers based on Bayes rule are Naïve Bayes Classifier, Bayesian Networks, Linear Discriminant Analysis and Quadratic Discriminant Analysis.

Linear classifiers classify the data without taking into account the underlying distributions describing the training data. Linear classifiers divide the l-dimensional feature space in a linear way that all (or most) points are classified correctly. It is the most simple and computationally effective classifier. Classifiers that can be grouped under the linear classifiers are the Perceptron Algorithm, Least Squares Methods and Linear Support Vector Machines.

Nonlinear classifiers divide the space in a nonlinear way that all (or most) points are classified correctly. Nonlinear classifiers are closely related to linear classifiers, which means that linear classifiers can also be used as a nonlinear classifier with some adjustments. Nonlinear classifiers are the Two- and Three-Layer Perceptron, Nonlinear Support Vector Machines and Decision Trees. There is also the possibility to combine different classifiers to improve the classification.

In unsupervised cases, one cannot speak of a real classification since the class labels are not known. Therefore unsupervised pattern recognition is defined as clustering, which can be defined as revealing the organization of patterns into clusters. Clustering algorithms can be divided into sequential algorithms, hierarchical clustering algorithms, clustering algorithms based on cost function optimization and other.

Sequential algorithms produce a single clustering. The number of clusters is not known in advance, therefore a threshold needs to be set. Every new vector is considered by the algorithm, after which is assigned to an existing cluster or a new cluster is created. It all depends on the distance to the other clusters. The number of clusters can not exceed the defined threshold. Different sequential algorithms are the Basic Sequential Algorithmic Scheme (BSAS), the Modified BSAS and the Two-Threshold Sequential Scheme.

Hierarchical clustering algorithms do not produce a single clustering, but produce a hierarchy of clusterings. The entire dataset is considered at the beginning of the algorithm. The two closest objects in the feature space are clustered depending on the distance that is chosen, e.g. Euclidean distance or city-block distance, and the method of linkage, e.g. single linkage or complete linkage. After the clustering, all distances are calculated again with the cluster included. Again, the closest objects are clustered and this continues until there is only one cluster. The way of clustering can be depicted in a dendrogram, of which an example is given in Figure 2.2. Cutting the dendrogram at the biggest leap offers often the best clustering.

Clustering algorithms based on cost function optimization relies, as the name already implies, on the optimization of a cost function using differential calculus techniques. Four major categories can be defined for this clustering algorithm: the mixture decomposition algorithm, the fuzzy algorithm, the possibilistic algorithm and the hard clustering algorithm. The most well-known clustering algorithm, k-means clustering, falls under the latter category. The goal of the k-means clustering algorithm is to determine a set of k points in the feature space to minimize the mean squared distance from each data point to the nearest k (Kanungo et al., 2002).

2.1.3 Statistics

Data mining and pattern recognition benefit of the field of statistics. According to Friedman (1998) statistics can "be defined as the set of problems that can be successfully addressed with these and

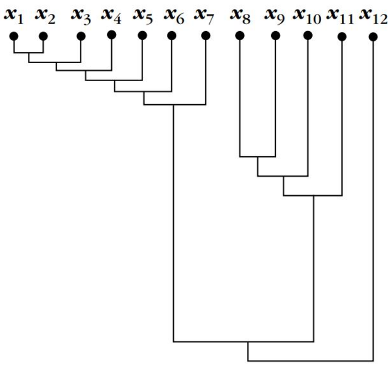


Figure 2.2: An example of a dendrogram. Courtesy of Theodoridis and Koutroumbas (2008)

related tools [...]:

- Probability theory
- Real analysis
- Measure theory
- Asymptotics
- Decision theory
- Markov chains
- Martingales
- Ergotic theory
- etc..."

Subsection 2.1.2 about pattern recognition already explained methods in that field that uses the probability and decision theory from the statistics field. It can be stated that both the data mining and the pattern recognition domain benefit from the statistics domain. The mathematical core of statistics is used as a ground layer in both domains.

2.2 Attribute Extraction from Movement Trajectories

Two related works are similar to this research, the work by Zhang et al. and the work by Chen and Krumm. This Section will go more into detail in the methods that are used for their research.

Zhang et al. "describe a process which incrementally improves existing road data with incoming new information in terms of GPS traces". The GPS traces that are used by Zhang et al. are downloaded from OSM. These GPS traces have a accuracy of 6 to 10 meters and the coverage of the traces vary from road to road. It is stated in the paper that highways tend to have a density of 30 to 80 GPS traces, where a busy city road has less than 20 and a local road has a few or less and sometimes none. TeleAtlas-data is used for an independent quality analysis.

In the preprocessing stage GPS points are linked using the timestamps and different trips are split into separate trips. The threshold for splitting trips is when the distance between two points is larger than 300 meters or if the heading changes more than 45 degrees. Also the speed is derived from the GPS traces. Thresholds of 250 km/h in highway areas and 100 km/h in urban areas are set.

For extracting the centerlines of the roads the OSM road map is used as initial prior information. First the GPS traces need to be matched to these roads. Three conditions are used to find the corresponding road to a GPS trace: distance to the road, direction and the angle between the trace and road. The first step in the process is to draw perpendicular lines of 30 meters along the roads to select possible traces, these lines can be seen in Figure 2.3. Traces that intersect with the lines are selected and then checked if the heading is also within 20 degrees of the direction of the road. This method is not able to separate traces from roads that are parallel and close to each other, e.g. the two roads in Figure 2.3. Therefore the authors created a clustering method, which also makes use of the difference in speed between traces. The clustering method used in the paper is a fuzzy c-means algorithm. This algorithms sets two initial cluster centers at the beginning and the end of a perpendicular line to achieve the maximum separation of the two clusters. Traces with a membership degree higher than, at least, 0.5 are then assigned to the cluster. When the map matching is done, the new centerline can be determined. The authors select the points within 95% confidence interval and calculate the center of these points. This center is then added to the centerline vertices.

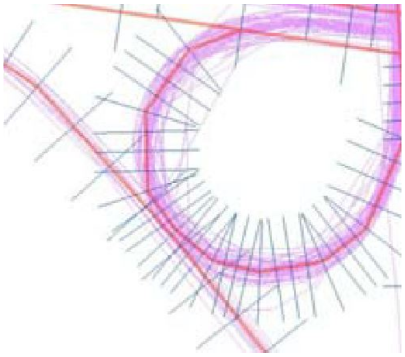


Figure 2.3: Perpendicular lines along the roads to select possible traces. Courtesy of Zhang et al.

The paper addresses not only the centerline estimation, but also extracts some attributes. An estimation of number of lanes has been made by using the standard deviation of the distribution of the GPS traces with respect to the road. The method assumes that two or more lane roads have a width of 3.5 meters and one way roads have a width of 5 meters. Therefore, when two times the standard deviation is smaller than 5.5, $2\sigma<5.5$m, the road is considered a one way road. When two times the standard deviation is larger than 5.5, $2\sigma>5.5$m, the road is considered a $2\sigma/3.5$m lane road. Besides number of lanes, the paper also addresses turning restrictions as possible attributes. It considers 6 types of turn restrictions: straight on, left turn, right turn, no straight on, no left turn and no right turn.

Finally, the paper validates the results of the method. The TeleAtlas dataset is used for validating the positional accuracy of the roads, having an accuracy of 2 to 10 meters. The data is validated by checking the number of roads that are within 2, 5 and 7 meters from the TeleAtlas data. OSM roads have a 14.8% rate for a 2 meter buffer against 27.4% for the proposed method. For 5 meters it is 46.8% (OSM) and 61.7%, for 7 meters it is 65.8% against 73.9%. The method proves to be successful in improving the road data. Also the number of lanes where validated and it turned out that out of 42 roads, 25 roads are given the correct number of lanes. This means an error of 40.48%. (Zhang et al.)

Where Zhang et al. focus more on the improvement of the geometry of roads than the extraction of attributes, Chen and Krumm focus on the latter aspect. In their paper they "introduce the idea of using a Gaussian mixture model (GMM) to model the distribution of GPS traces across multiple traffic lanes" and in that way "automatically compute the number and locations of driving lanes on a road". This automated method could, in the future, maybe replace the time consuming and costly alternatives, e.g. using aerial imagery. The GPS data that is used by Chen and Krumm is collected from 55 shuttle vehicles driving around their institution collecting a total of about 20 million GPS points. The authors concentrated on intersections for the testing of the method.

Again, perpendicular lines are drawn along the road to find intersections between the lines and the GPS tracks. This will result in a one dimensional dataset representing the spread of the GPS traces. The paper makes use of the assumption that GPS tracks cluster near the center of the lanes, however certain errors will cause the points to be spread. Therefore they assume that the spread within a lane can be modelled as a Gaussian distribution. First, the expectation-maximization (EM) algorithm is used to learn the parameters of a GMM. Some constraints are added to the EM algorithm. The first constraint is that the width of the roads are equal, resulting in similar distances between the mean of different lanes. The second constraint is that the variance of the Gaussian components remains the same between different lanes. The result of the constraint GMM on a road with two lanes can be seen in Figure 2.4. Akaike information criterion (AIC) and Bayesian information criterion (BIC) were used for model selection, which means automatically selecting the number of lanes to fit the parameters of the GMM. However, these two methods did not work well for the problem. Therefore a different regularizer is used, called Lane Spread.

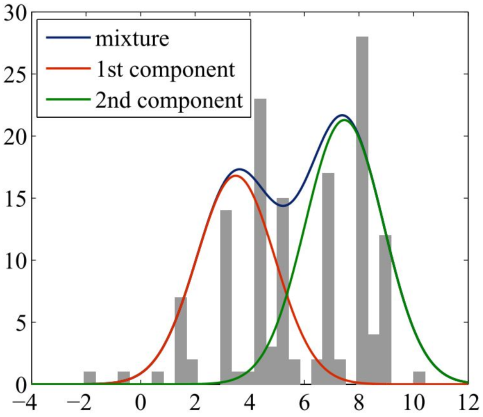


Figure 2.4: Result of a constraint GMM on a road with two lanes. Courtesy of Chen and Krumm

Cross-validation was used for testing, using 80% to train and 20% to test the data. For the three intersections, that this paper concentrated on, the best result was with the restricted (constraint) GMM using the Lane Spread regularizer. This combination led to an error of 28.06%. The new methods proposed in the paper give better results than previous methods, but it also naturally still models the inherent noise in GPS data.

Both related works show there is still work to be done regarding attribute extraction and, in particular, lane detection. Probably there are more attributes that can be derived from GPS data that are not treated in these two papers.

CHAPTER 3

Methodology

The outcome of this research is the updated OSM data, containing the missing attributes for roads. To come to this output, a few steps need to be taken. The input of the system is the data that is already available: the raw GPS data and the OSM data. First a segmentation and classification need to be performed on the GPS data in order to filter out tracks made with transportation modes that are not useful in this thesis and to derive multiple elements like speed and heading. These will later be used to derive the attributes. Then a map matching algorithm will be performed on the GPS dataset with the OSM data to match each GPS point to the correct road. After the map matching the attributes for every road can be extracted by using different attribute extraction algorithms. The resulting dataset can then finally be validated and updated into OSM. All the steps are depicted in Figure 3.1. This Chapter gives insight in the different steps towards the final result. First the available data will be described. Secondly, an overview is given on the different available map matching algorithms. After that the various attribute extraction algorithms and their implementation for updating OSM will be treated. Finally, the possible privacy issues concerned with this research are discussed in the final Section.

3.1 Data

The raw GPS data is acquired in February 2013 for an urban analysis project by Paul van de Coevering of the Urban and Regional Development Section of Delft University of Technology, one of his works is the research on public transport from the perspective of urbanization and mobility (Hilbers et al., 2009). The data contains GPS points containing coordinates (x, y, z) and time (t) recorded every 5 seconds by more than 800 people in Amersfoort, Veenendaal and Zeewolde during a certain timespan. The data consists of 40 million GPS points in total of which 3,7 million are made by car and 1,5 million by bicycle. These GPS points result in tracks with a total length of 385.000 kilometers of which 272.000 kilometers are made by car. This raw data is imported in a database and preprocessed for calculating various statistical values, after which a classification and segmentation algorithm is performed as explained by Biljecki et al. (2013). This data contains, besides the raw data elements, for instance: speed, heading and transportation mode.

The OSM data is freely downloadable for everyone and thus can be viewed online, but also offline in a Geographical Information System (GIS) and it also can be stored in a database. For the OSM "highway" data in a database the table contains, for example: the id, type, one way street (yes/no), maximum speed and access. The last three columns contain the attributes that this thesis tries to derive in an automated way from the GPS tracks, adding also the number of lanes. Both the OSM data as the GPS data are available in a PostGIS database.

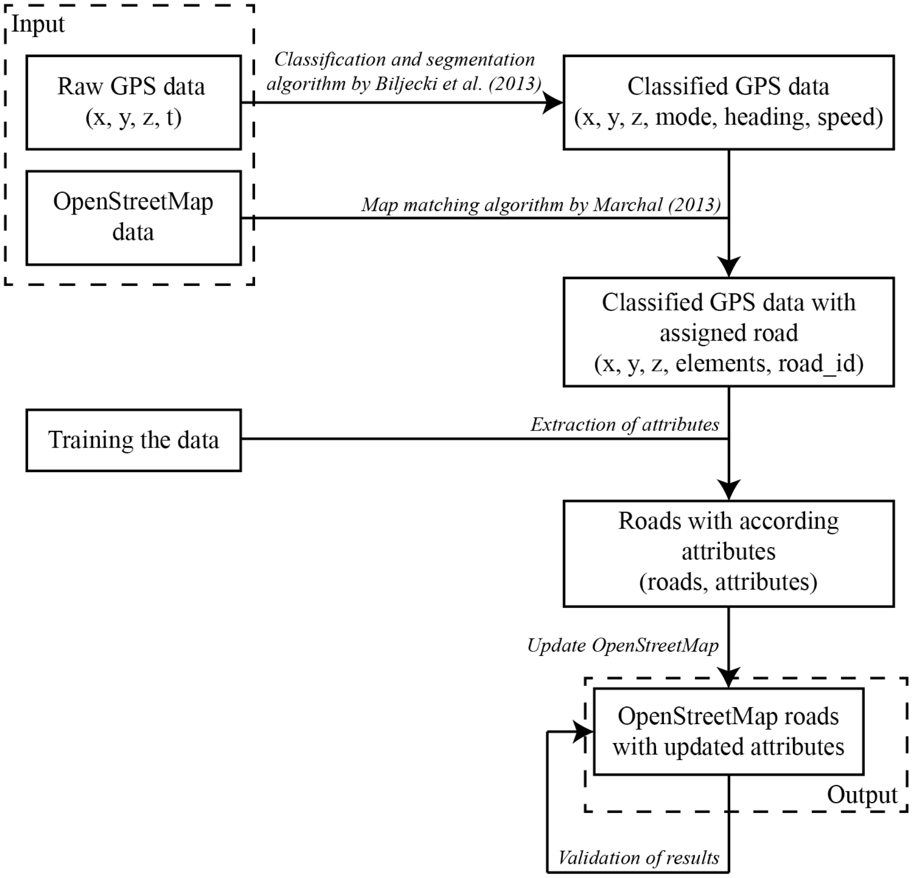

Input
Raw GPS data
(x, y, z, t)
Classification and segmentation algorithm by Biljecki et al. (2013)
Classified GPS data
(x, y, z, mode, heading, speed)
OpenStreetMap data
Map matching algorithm by Marchal (2013)
Classified GPS data with assigned road
(x, y, z, elements, road_id)
Training the data
Extraction of attributes
Roads with according attributes
(roads, attributes)
Update OpenStreetMap
OpenStreetMap roads with updated attributes
Output
Validation of results

Figure 3.1: Diagram explaining the methodology of the research

When visualizing these two datasets, it becomes visible that there has to be dealt with inaccuracies (Figure 3.2). Both datasets have their errors. The GPS tracks have an error due to the accuracy of the GPS device. These GPS tracks are also used to map OSM, thus these errors will also occur in OSM. OSM has an average error of approximately 6 meters compared to the positions recorded by the Ordnance Survey in the United Kingdom (Haklay and Weber, 2008).

The two datasets together can already be used for extracting some information that can be useful for the road attributes. A cross-section of a certain road can be made with all nearby GPS points. The distances from the GPS points to the centerline of the road can then be calculated. After this the distances can be plotted in a graph to see if there is a relation between high densities of points at a certain distance and the number of lanes of that road.

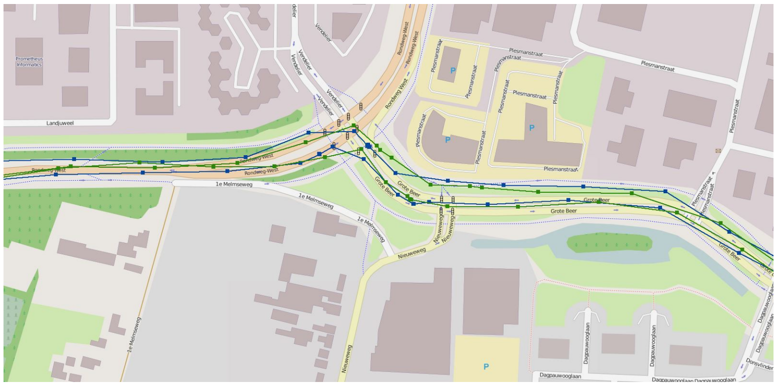


Figure 3.2: An example of GPS tracks visualised on OSM

3.1.1 GPS Data

"GPS provides accurate, continuous, worldwide, three-dimensional position and velocity information to users with the appropriate receiving equipment."(Kaplan and Hegarty, 2005). The GPS constellation contains at least 24 satellites, which are arranged in groups of 4 in 6 different orbital planes. To determine an accurate location, at least 4 measurements from different satellites are needed. The time of arrival (TOA) method is used to determine the latitude, longitude and height of the user. GPS falls under the more general name Global Navigation Satellite Systems (GNSS) together with other systems. Where GPS is a product of the United States, Galileo is the satellite system of the EU, GLONASS of Russia and Beidou is the satellite system of China. These systems, apart from GPS, are not fully operational yet or have less satellites in the air which means less coverage around the world. Therefore GPS is the most popular and most used satellite system, thus also used for this research.

From the results derived from the GPS devices, data can be enriched using calculations on the information of consecutive points (e.g. speed or heading). The following list contains all the available data that is used from GPS:

- Point ID (pid)
- User ID (userid)
- Time (time)
- Acceleration (acc)
- Speed (speed)
- Calculated speed (calcspeed)
- Heading (heading)
- Geometry (geom), Latitude (lat) and Longitude (lon)

The GPS device is not 100% accurate, there are different error sources. Lots of these errors can be decreased by taking precautions. Differential GNSS (DGNSS) is one of these precautions, using a second receiver on a fixed position to increase the accuracy by correcting the errors of the measurements of the receiver. However, there are also errors which are more difficult to take care of, e.g. multipath and availability problems in urban canyons.

"Multipath is caused by multiple reflections of the microwave signal emitted by the GNSS satellite."(Lemmens, 2011). When these signals are reflected, the receiver can receive the same satellite signal multiple times. This can distort the measurements of the GPS receiver. Multipath tend to happen more often in urban areas than in rural areas, since there are more reflecting surfaces in an urban area. This causes the precision of GPS to decrease in these areas.

Next to multipath problems in urban areas, there are also GPS constellation problems for the built environment. Surrounding high buildings in an urban area can affect the availability of GPS satellites. An urban canyon model from Kleijer et al. proves the lack of satellites in the North, as can be seen in Figure 3.3. This causes streets that have a North-South orientation to have a limited availability of GPS satellites, where streets with an East-West orientation suffer less from this problem.

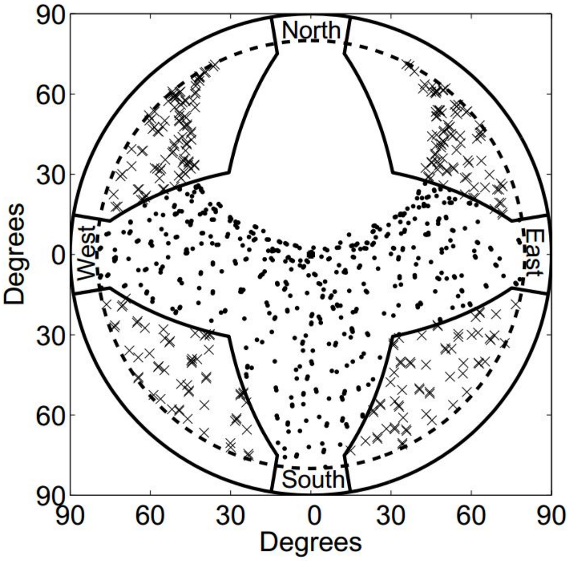


Figure 3.3: GPS skyplot for the urban canyon model. Dots represent visible satellites, crosses represent non-visible satellites. Courtesy of Kleijer et al.

It can be stated that GPS is not 100% accurate. Especially in urban areas there are more errors which can lead to a decrease of accuracy. This is something that should be taken into account.

3.1.2 OpenStreetMap Data

As already discussed in Section 1.1, OSM has a concept that follows the model of Wikipedia. This means that the user is important and responsible for the input and maintaining the data. Users cannot only add GPS tracks and tags, but they can also edit the existing map. They can add or edit roads, but also other elements of OSM. In Section 1.4 is already discussed what kind of errors result out of the freeform tagging system. Different contributors tend to insert the tags in a different way. This also counts for the way OSM is drawn. As already stated above, contributors can add and edit the geometry of the OSM roads. Different contributors lead, again, to different results. The detail of drawing roads differ between different contributors. Where one could draw a slightly bended road as a straight line, another could draw it as a multilinestring with multiple nodes following the curve of the road. These ways of drawing can have an influence on the results of this thesis and should be taken into account. It is not only the way of drawing, but also the level of detail per country that is different. The Netherlands is represented relatively well in OSM, in detail and in coverage. Other countries may already be satisfied with a decent percentage of coverage, not to speak of the detail.

The UML (Unified Modeling Language) diagram of OSM data can be seen in Figure 3.4 (Hahn, 2014), this diagram shows the relations and contents of the database in a schema. The relations between all the different components are depicted in this diagram. The top right side of the diagram is the most important part for this thesis, the ways and their tags. A tag consists of a key (e.g. "maxspeed") and a value (e.g. "30").

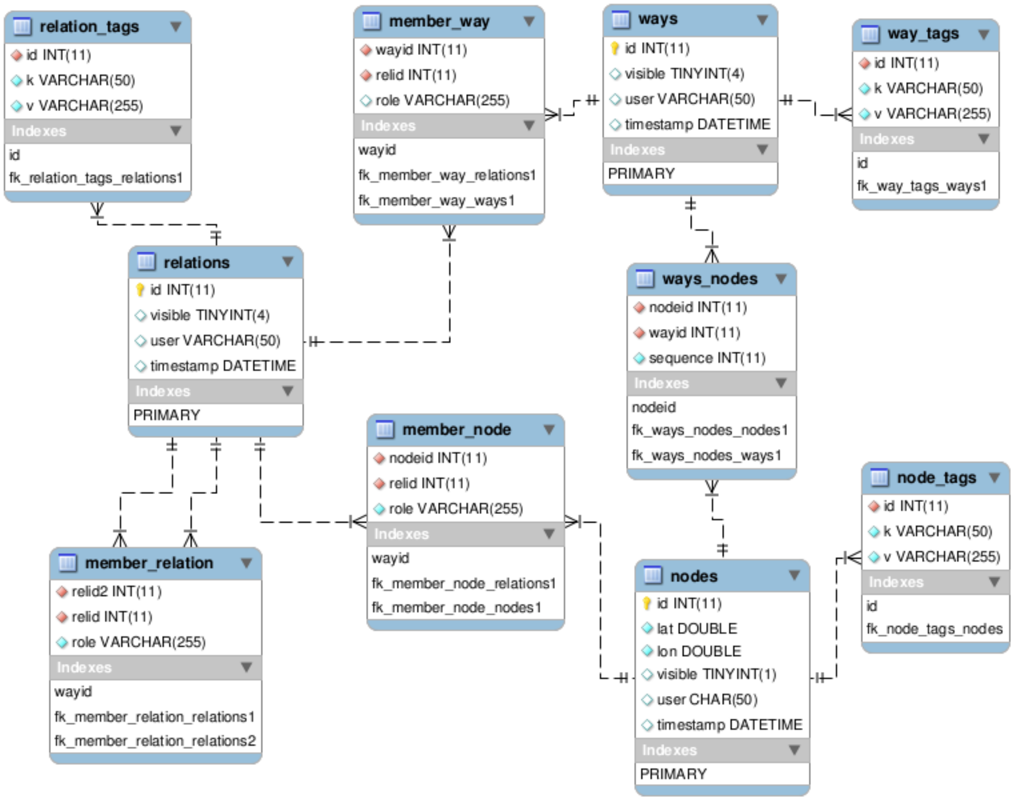


Figure 3.4: UML diagram of OpenStreetMap data. Courtesy of Hahn (2014)

3.1.3 Analysis of Input Data

As already stated in the previous subsections, there are always standard errors with data. GPS data suffers from inaccuracy, availability of GPS satellites and urban canyons, where OSM data relies on users/contributors who are responsible for the data. These are not the only causes for errors, there are more. This Subsection discusses the analysis of the input data.

During the analysis of the GPS data, one problem arose: stationary points. The GPS dataset contains many stationary points, i.e. points with a speed around 0. Analysis of the GPS data show that 709.127 points out of a total of 3.660.561 points made by car have a speed lower than 5 km/h. One of the causes of these stationary points can be seen in Figure 3.5. Apparently, the carrier of the GPS device forgot to turn of the GPS device before going into his/her house, resulting in a large scatter of GPS points near the house. This does not only happen when going into buildings, but also when the carrier of the device forgot the GPS device when leaving the car. Multiple stationary points have been found on parking lots.

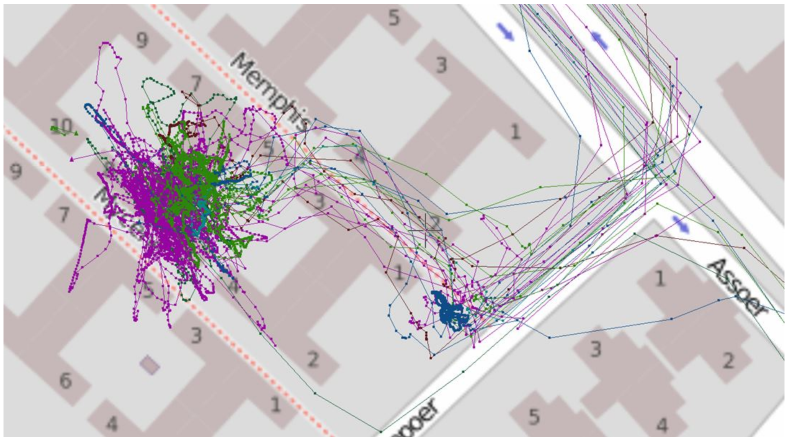


Figure 3.5: Stationary points of a random GPS track near a house

As mentioned in Subsection 3.1.2, the way of drawing can differ in OSM. There are some cases where this way of drawing can lead to errors. Roundabouts, for example, are considered roads in OSM. Thus when working with road data it is possible that you encounter a roundabout. If the roundabout is treated as a normal road it could cause errors in specific cases, because a roundabout is drawn as a closed linestring in some cases in OSM. Also, analysis of some OSM linesegments show the possibility that they contain two consecutive points that are the same. This could cause problems when splitting the linesegments of a road into smaller segments, resulting in a segment which contains exactly the same points.

3.2 Map Matching

Matching the GPS points to the OSM data is an important step in the research. If a GPS point is matched to the wrong road, the attributes will also be wrongly updated. Therefore, the error of the map matching determines also a part of the error of the assignment of the attributes. It should be noted that this research is based on post processing, not on real-time data mining.

Different map matching algorithms can be defined. These map matching algorithms can be categorised into 4 groups: geometric algorithms, topological algorithms, probabilistic algorithms and advanced algorithms (Quddus et al., 2007). In this Section, all four categories will be explained shortly and some related work will provided.

3.2.1 Geometric Algorithms

Geometric map matching algorithms use, as the name already implies, the geometric information of the road network. Geometric map matching can be implemented in a few different ways as researched by Bernstein and Kornhauser (1998). The first method is point-to-point matching, this method makes use of the closest node search algorithm. For every GPS point its closest road network node is mapped. This method is highly error sensitive, as both the inaccuracy of the GPS points and the number of nodes per line or arc causes severe differences in performance. The second method that is proposed by the authors is the point-to-curve matching. Distances are calculated between the GPS points and the edges in the network, after which the edge with the shortest distance is assigned to the GPS point. This method performs already better than point-to-point matching, however there are still problems like with a high density of roads or at junctions. At junctions the GPS point can easily be assigned to the road perpendicular to the road that the vehicle is actually on. With a high density of roads, the GPS point can easily be assigned to a road parallel to the road that the person was on. The visualization in Figure 3.6 shows a GPS track on a railroad, however there are some points that are closer to the highway next to it. These points would be assigned incorrectly and could cause the enrichment to enrich the highway with railroad attributes.

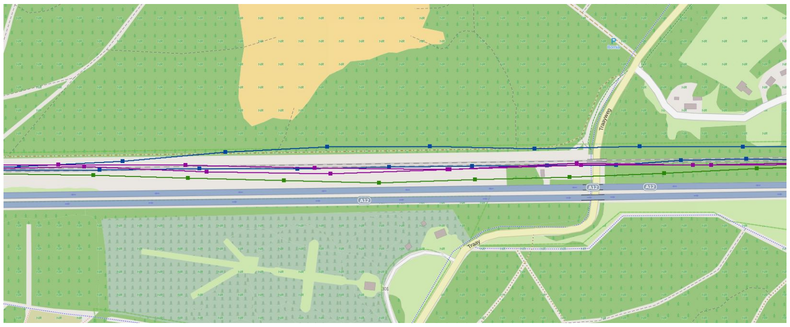


Figure 3.6: GPS track on a railroad with a highway parallel to it

The third method is curve-to-curve matching. In this method linear curves from the vehicle's trajectory are matched with the edges of the road network. The edge of the road that is closest to the edge of GPS tracks will be assigned to the corresponding GPS points. This method still has some errors due to outliers and because it is still based on point-to-point matching.

The geometric map matching algorithms probably are the most simple map matching algorithms, however these algorithms are also quite sensitive to errors and therefore is the performance of these algorithms most likely less than other algorithms.

3.2.2 Topological Algorithms

Topological algorithms are about the topology of the road network. "In GIS, topology refers to the relationship between entities (points, lines, and polygons)." (Quddus et al., 2007). These relationships are adjacency, connectivity and containment. Therefore topological algorithms make use of the connectivity and contiguity of links in a network. Recent work on topological algorithms are the enhanced weight-based topological algorithm by Velaga et al. (2009) and an efficient map matching algorithm of large GPS data sets by Marchal et al. (2005).

The first algorithm uses link data (including ID, start node and end node), node data (including ID, easting and northing coordinates), positioning and navigation data from a navigation sensor, vehicle heading and speed and turn restriction data for junctions. Next, the map matching process is divided into three stages: the initial map matching, map matching on a link and map matching at a junction. The first stage is to identify the first correct link for the first positioning point. The next fix is then determined by checking three criteria. First it checks whether the vehicle is in a stationary condition, in this case it will be matched on a link. Second, the algorithm checks whether the vehicle is travelling on the previously matched link which would also cause the algorithm to match it on a link. Third, it checks whether the vehicle is near to a junction, in this case it will be matched at a junction. The algorithm makes also use of weights when selecting candidate links. For the initial map matching the heading and proximity are used. For map matching at a junction, two additional weights are used: turn restrictions at junctions and link connectivity (Velaga et al., 2009).

The second algorithm by Marchal et al. (2005) relies only on GPS coordinates and the network topology. First the distance between a point and a road segment is calculated. During the initialisation the closest set of links to the first data point are found. Due to signal losses and other types of noise the GPS tracks will be split into paths. Next, the network topology is used to determine the subsequent points. Scores are calculated for each path, which are then ranked. These paths are all stored and finally the best ranked path can be chosen (Marchal et al., 2005).

3.2.3 Probabilistic Algorithms

"The probabilistic algorithm requires the definition of an elliptical or rectangular confidence region around a position fix obtained from a navigation sensor." (Quddus et al., 2007)

Basically, the error variances are used to derive an error region which tries to identify a road segment on which the vehicle is travelling. Ochieng et al. (2009) developed an improved probabilistic map matching algorithm. This algorithm takes into account the error sources associated with the positioning sensors and the historical trajectory of the vehicle. They have also made use of topological information on the road network and the heading and speed of the vehicle, like the enhanced weight-based topological algorithm of Velaga et al. (2009). The algorithm starts, like most map matching algorithms, with the initial map matching process followed by the subsequent matching process. A confidence region is created to select candidate road segments. The earlier mentioned input help selecting the road segment of which the probability that the vehicle is on that road is the highest.

3.2.4 Advanced Algorithms

Advanced map matching algorithms are usually the more complex algorithms, like Kalman Filters and fuzzy logic models. They are based on different methods and a large variety of advanced algorithms are therefore available.

Quddus et al. (2006) describes a high accuracy fuzzy logic based map matching algorithm for road transport. This algorithm is based on fuzzy logic theory, which let the algorithm deal with qualitative terms like likeliness to identify a correct link. The inputs of the system are GPS data and deduced reckoning (DR) data. These two data sets combined result in a data set containing

Easting, Northing, vehicle speed, heading, and the associating error variances. Fuzzy logic is not only used to identify the correct link among candidate links in the initial map matching process, but it is also used in the subsequent map matching process. Again, the subsequent map matching process consists out of matching on a link or matching at a junction.

Winter and Taylor developed a modular neural network (MNN) approach to improve map matched GPS positioning. "The research deals with designing and developing a MNN technique that autonomously chooses the appropriate expert from a number of locally trained ANN (artificial neural network), based on road shape." (Winter and Taylor). An advantage of this system is that, once the neural network is trained, the algorithm can achieve a high performance for real-time map matching. The algorithm uses CDOP as a road shape indicator to categorize the input in four different categories that are based on CDOP value ranges. Each category has its own ANN. A disadvantage of this method is that for the training and testing of the individual ANNs and the resulting MNN, a large amount of GPS data is needed.

Newson and Krumm propose a Hidden Markov map matching algorithm. The Hidden Markov Model (HMM) is used in this algorithm to integrate noisy data and path constraints smoothly. The algorithm uses measurement probabilities and transition probabilities to derive the optimal path, which is the most probable path.

Jawad and Kersting (2010) describe a kernelised map matching algorithm, which make use of kernels. First the consistency between the similarity measures are captured by the kernel matrices of the trajectory and the relevant part of the road network. Then the resulting relaxed assignment is rounded into a hard assignment fulfilling the map constraints.

There are different map matching algorithms, which all have their advantages and their disadvantages. The easiest method is to take the shortest distance to road, but this method also gives a high error rate. A wrong classification of a GPS point would, in the case of this thesis, also mean wrong attribute information for the assigned road. It should also be noted that not the entire track has to be completely matched. Outliers in GPS data can easily be left out, since not all points are needed for the attribute extraction.

3.3 Attribute Extraction and Updating

This Section discusses the methodology for extracting and updating the attributes. Every attribute has its own method and therefore for every attribute a short description of the methodology is given. All attribute extraction methods will use PostgreSQL (PostGIS) and Python to derive the attribute information. Both, the OSM data and the GPS data, was already available in PostgreSQL in advance of this thesis. Therefore it would be logical to continue in this database. An advantage of this database is the PostGIS extension which makes it possible to use geometries in the database. As a programming language, Python is used. Python is an open-source programming language which can be extended with multiple libraries. Therefore it is possible to connect to the PostgreSQL database and derive information from it. It is also possible to implement mathematical and scientific libraries, and machine learning libraries. This makes Python a complete language which is beneficial for this research. All attributes will be trained on the 10 most used roads in the dataset to get the best test results.

Different attributes are used in OSM for the "highway" tag. The main tag is the type of the road. There are multiple secondary tags (OpenStreetMap, 2013b):

- name
- one way
- speed limit
- structure
- access
- lanes
- surface
- reference
- abutters
- maximum height
- maximum width

For the type of road, the name of the road and its reference (e.g. A1) it is not possible to derive from only GPS tracks. The same is for the "abutters" tag, which describes the predominant usage of land along the road. It is also impossible to derive information on the surface of the road and without the knowledge of the vehicle in which the GPS tracks are acquired it is also not possible to extract the maximum height and width of the road. Extracting the structure, e.g. bridges and tunnels, might be possible using signal shortages and the underlying landuse. However, for other structures it might be more difficult and this attribute is not as common as the remaining 4 secondary tags. Therefore the existing attributes in OSM that will be derived are mentioned below. Between quotation marks are the tag names as they are used in OSM.

"oneway"
For determining whether a road is a one or two way road it is important to have the heading of the GPS point and the direction of the road. Both can be derived from the data. Next, a threshold needs to be carefully chosen to indicate which points are going in the same direction of the road. This threshold will determine whether a GPS point moves in one direction of the other. Finally there is always the possibility to have outliers, therefore a ratio has to be chosen to determine whether a road is one way or two way.

"maxspeed"
The most important thing to derive the maximum speed is to anticipate the behavior of the drivers. Chosing the correct thresholds for assigning the speeds on a road to a specific speed limit is the key to success for this attribute.

"lanes"
Deriving the number of lanes from GPS tracks has been proven to be a difficult task. Two different methods are already introduced in Section 2.2. An attempt will be made to implement these methods in this thesis. To be able to implement these methods a few steps need to be taken first. The distance from the GPS point to the road should be calculated and also on which side of the road the point lies. This will give a relative position of the point to the road and after that the methods can be implemented.

"access"
The access attribute relies on map matching. The available GPS data contains different transportation modes. If these transportation modes are matched correctly to the roads, the according vehicles can be added to the access list. However, the OSM road network should have a full coverage and contain all different roadtypes (e.g. buslines, cycleways, footpaths) to make this method 100%

accurate.

This research not only extracts existing attributes of OSM, but also introduces new attributes. These attributes might contribute to a more complete and detailed map. Commercial map makers like Google Map Maker (Google, 2014) and HERE Map Creator (HERE, 2014) have more attributes implemented compared to OSM. HERE has the average speed attribute, whether the road is well lit and whether it is a crowded road incorporated in their maps and Google adds the priority of the road, quality of the road and the house numbers along the road. From these attributes, the average speed could also be implemented in this research and an attribute to measure how crowded the roads are could be developed. However, the other attributes are difficult to derive from GPS tracks.

Below, the new attributes that are investigated but that are not available in OSM are listed. Some of these attributes are already implemented in commercial maps. However, not all of them are extracted automatically at these companies and it might be of interest to introduce such attributes in OSM.

"averagespeed"
Deriving the average speed is not that hard, all speeds of a road should be summed and divided by the number of GPS points. This will result in the average speed of the specific road.

"congestion"
Congestion is an attribute that is based on the "averagespeed" attribute. The average speed can be calculated for every hour of the day. The results per hour can then by compared and hours in which the average speed is significantly lower than the other averages can be added to the congestion tag. This makes it easy to see in which hours of a day you need to avoid certain roads.

"importance"
Importance can be described as the usage of a road compared to other roads. A percentage of the usage of the road can be calculated. Chosen thresholds for the different categories can then categorize the roads in importance.

"geometryerror"
This attribute will define an error given to the geometry of the road. By comparing the mean distance to the centerline of the road with the actual centerline of the road given by OSM, one can define a certain error. If this error is higher than a certain threshold a warning can be given for that road that a change of geometry might improve the accuracy of the geometry.

These attributes, or tags, are all connected to one specific road. These roads are called ways in OSM. Every way has its own tags as could be seen from the UML diagram in Figure 3.4. As already stated, these tags have their own keys and values. In the UML diagram of Figure 3.7 these keys and their values are depicted with their link to the ways. The values for the existing attributes are from OSM and the new attributes contain the values as they are expected in the final application.

Finally, the updating of these attributes in OSM directly might be a difficult case. It should be investigated if directly updating OSM is possible, otherwise other alternatives could be used. OSM data is freely available for everyone, this means it can be downloaded and stored in a database.

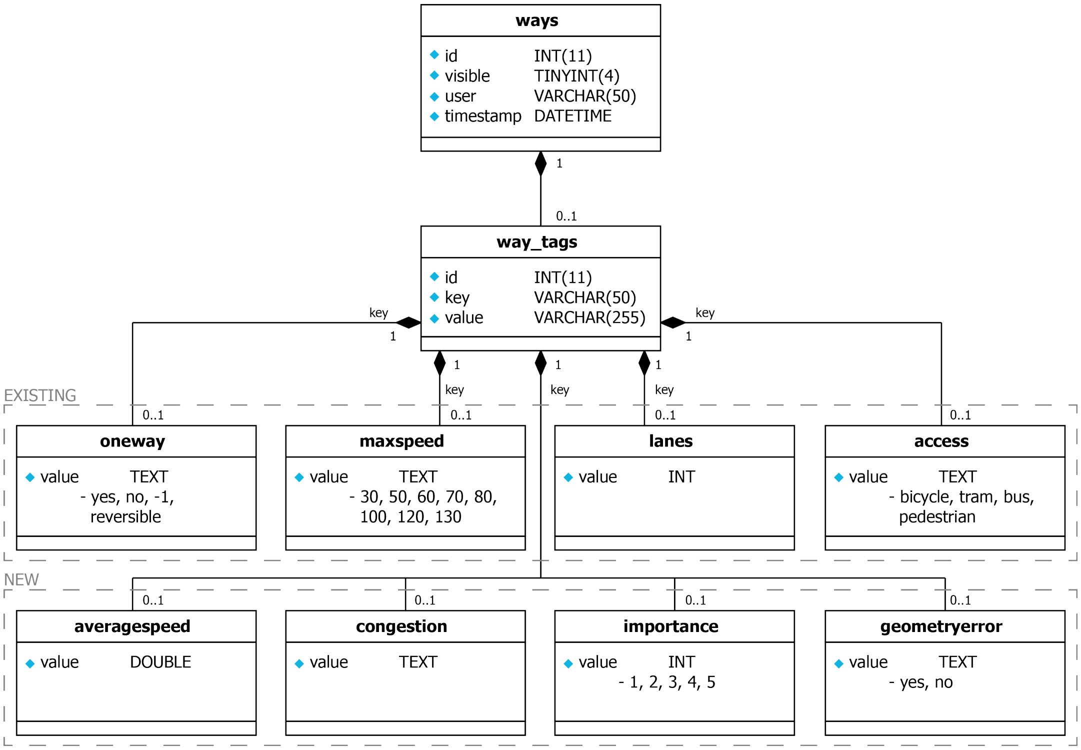


Figure 3.7: UML diagram of the different tags and their values

The database can be updated with the new or improved attribute information.

3.4 Privacy

Privacy is an everyday returning issue, also in the field of Geomatics. Privacy is often described as "the right to be left alone" (Brandeis and Dossick, 1890). Like in many other applications, this research also concerns privacy. In this case, it is information privacy. Information privacy describes the privacy of personal data. GPS tracks of moving people is personal data. This means that with the data people can be identified, directly or indirectly.

However, there are no privacy issues concerning the GPS data used in this thesis. First the data that is used is anonymized, this means that no direct link between the person that carried the device and the data is saved. This has been done by changing the ID of the GPS device to a random ID of which the original ID cannot be retrieved. Therefore, the data converted from personal data to impersonal data.

Secondly, the final result of this thesis will be derived from mass data from which individual tracks cannot be reconstructed. Only the result of the attributes are stored leaving no trace to the original GPS data. This last point is an important point for future implementations. Basically this makes sure that no link to the GPS tracks or devices can be found. However, usually people need to give permission to acquire their location. For future implementations people can be asked if their location can be used for "enrichment of the dataset" or for "the improvement of the traffic flow". There can be multiple purposes to use the location of the user. In OSM, the contributors already agree that their GPS tracks are used by OSM when uploading them. Therefore the uploaded GPS tracks of OSM can already be used for attribute extraction. It should be noted that these GPS tracks are different from the GPS tracks used in this research. The GPS tracks in this research are from real-life situations, where OSM contributors deliberately drive a certain route which causes the driving behavior to change.

CHAPTER 4

Preprocessing the Data

This Chapter describes the preprocessing steps that have been made in the research process. These are multiple steps from applying the map matching and classification algorithms to filtering the datasets. The first Section describes the TrackMatching algorithm which provides the map matching for this research. The second Section elaborates on filtering the GPS and OSM data and in the last Section the creation of the training dataset will be explained.

4.1 TrackMatching

The final map matching algorithm that is be used for this project is based on the topological algorithm by Marchal et al. (2005). A cloud based web service called TrackMatching is developed in 2013 by Fabrice Marchal. This web service matches GPS data on the OpenStreetMap road network. It is accessible through a REST API which uses four HTTP methods (GET, POST, PUT and DELETE) to execute different operations and can be integrated within the final software prototype. The service is also free to use for developers that use the API to build early stage applications. TrackMatching uses GPX-files or custom CSV/Text format containing only the positions and the timestamps as an input. The output is the routes that are matched from GPS points. These can be used for further processing to acquire the id of the road per GPS point. This Section will give insight in the map matching process performed for this research. First the preprocessing steps before the actual map matching will be described, after which the map matching and the results are presented.

4.1.1 Preprocessing

The input of the TrackMatching algorithm are the consecutive points of a GPS track consisting out of the position of the GPS point (x, y, z) and the timestamp (t). This input can be inserted in the form of a GPX-file or a custom CSV/Text format. The latter will be used for this research, because this format can handle large bunches of data better than the GPX-files. Before the map matching can be performed, the CSV files should be created.

The first step is therefore to extract the data from the database into a CSV file. Python is used to automatically extract all the data from the PostgreSQL database and write it into a CSV file. It is important that the GPS points are ordered by time and by user ID to retain the correct sequence of the points. A large CSV file containing all the GPS points is the result of this first step. This file should be separated into smaller files of which each having their own user ID. Therefore first the all the distinct user IDs are extracted to acquire all different users. These user IDs are then used to split the large CSV file into smaller CSV files per user ID.

4.1.2 Map Matching using TrackMatching

Now that all different tracks have their own CSV file, they can be map matched one by one. TrackMatching can be accessed via a REST API from Python, a for loop will make sure that all the different files will be matched one at a time. The Python code that is used consists out of an URL that calls the REST API and gives a payload containing a personal id and key and some options regarding the map matching. Additional headers contain the content types of the input and of the output. Finally, all the output will be written into different XML files that still separate the tracks.

The returned XML file contains a route class that describes the matched route of the GPS track, as can be seen in the code below:

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
 - <result>
  - <diary>
    - <entries>
      - <entry>
        - <route>
          - <link dst="343639788" id="185815954" src="307697863">
              <wpt id="55363679"/>
              <wpt id="55363680"/>
              <wpt id="55363681"/>
              <wpt id="55363682"/>
              <wpt id="55363683"/>
              <wpt id="55363684"/>
              <wpt id="55363685"/>
              <wpt id="55363686"/>
            </link>
          - <link dst="307697873" id="185815945" src="343639788">
              <wpt id="55363687"/>
              <wpt id="55363688"/>
```

This route lists the driven roads, called 'links'. These links contain an OSM ID and the corresponding start and end node of that road. Also, the matched GPS points are listed underneath the links by using the IDs of the GPS points. The next step is to derive the IDs of the GPS points and the corresponding road ID. A Python script parses the XML file and writes all the GPS point IDs and corresponding roads to a new CSV file. After parsing all the XML files and writing them to a single CSV file, the latter can be imported into the PostgreSQL database. All data from the GPS points can then be joined with the GPS point ID and all data from the OSM roads can then ben joined with the road ID. This will result in a table containing all information regarding the map matched GPS points.

4.1.3 Analysis

Analysis of the final map matched data show that there are still some errors in the data. One of the most common errors is caused by stationary points, these are points where the GPS device is not moving over a long period of time. This can be for example caused by leaving the GPS device inside a parked car. These points are not filtered before the map matching and also not by the

TrackMatching algorithm. This results in stationary points that are matched to one road, causing the road to contain GPS points which are not actually on that road.

One disadvantage of the service is that only motorised road networks are taken into consideration for the map matching. This will cause the service to match bicycle tracks that are on a cycleway to the motorised road as can be seen in Figure 4.1a. The flags represent the GPS points that are on the cycleway which is the blue dotted line. The points are matched to the blue thick line, which is the motorised road that is closest to the points and in the similar direction of the heading of the points. In the future it might be possible to match the GPS points of bicycles to the cycleways, but this depends on the server capacity of TrackMatching. Presently, cycleways are not included in the topological network to which the GPS points are matched and therefore it is not yet possible to map match bicycle points to them.

The analysis also showed that there were a few GPS points that were matched to road up to 10 kilometers away. Investigation of the tracks that contain these points turned out that the problem was not directly from TrackMatching, but indirectly from OSM. The problem can be seen in Figure 4.1b. The exit of the highway that is located at the top of the Figure has the attribute "access=no", which means that that part of the road is not accessible. The TrackMatching algorithm takes this into account and expects that the car is not able to take that route. Therefore all the following GPS points are matched as if the vehicle is still on the highway, instead of the road which is going south. This will cause the algorithm to match all the following points wrongly and increasing the distance between the GPS points and the assigned road.

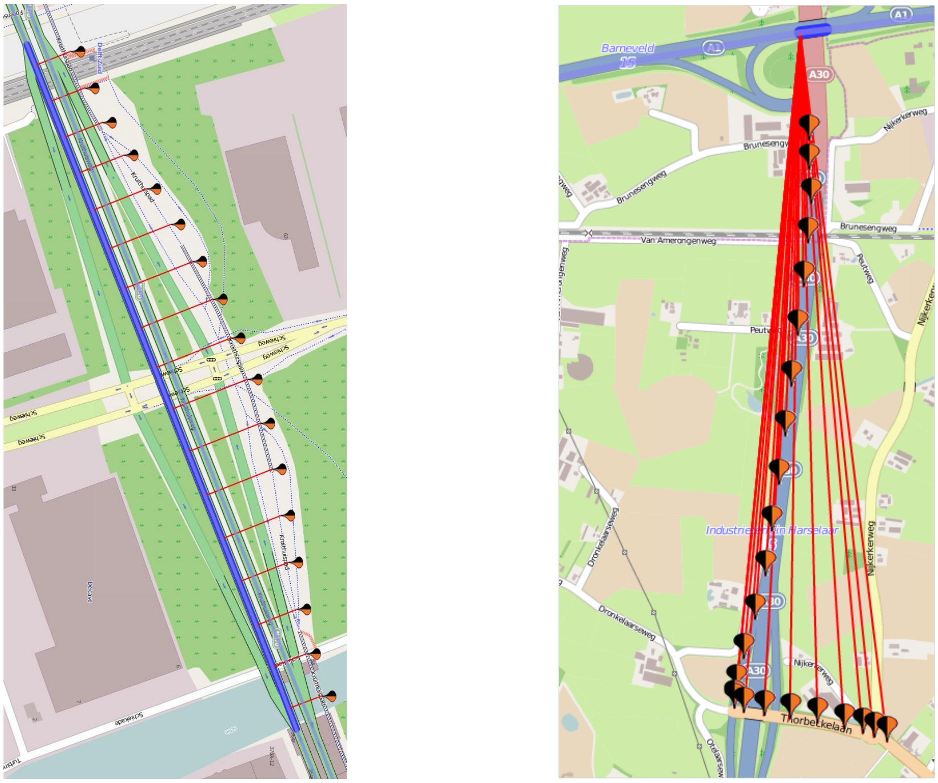
4.2 GPS and OpenStreetMap Data

In previous Sections it is already mentioned that GPS data contain different errors and that also the behavior of people will influence the resulting GPS track. All these influences also affect the map matching, therefore some preprocessing should be done to the map matched GPS and OpenStreetMap data to work with more reliable data. Sometimes people forget to take out the GPS device out of their car, resulting in many points on one location. These points are all matched to the same road, thus it appears that the road is used a lot while in fact it was only one car standing still. Not only these points, but also other wrongly matched points should be filtered out to acquire a dataset which is useful for the research.

The points that should be filtered out are stationary points and points that are most-likely not on the assigned road. Speed, heading, distance to the road and the geometry of the road can be used to filter out these points.

Stationary points can be easily filtered out by setting a threshold for the speed. The threshold chosen for this research is 5 km/h to filter out the stationary points and still keep the significant points that could be useful for the research. This resulted in a decrease of GPS points made by car from 3.658.750 to 2.949.623 points, this means a decrease of 19,4% stationary points.

Points that are most-likely not on the assigned road are points that have a large distance to the assigned road and have a different heading than the direction of the road. For example, a GPS points which is 100 meters away from the assigned road or a point which heading is perpendicular to the direction of the road are most-likely wrongly matched. The plots in Figure 4.2 were made to find a threshold in meters for the distance and in degrees for the heading. The plots describe the number of GPS points that fall inside the buffer on the y-axis and the according buffer on the



(a)

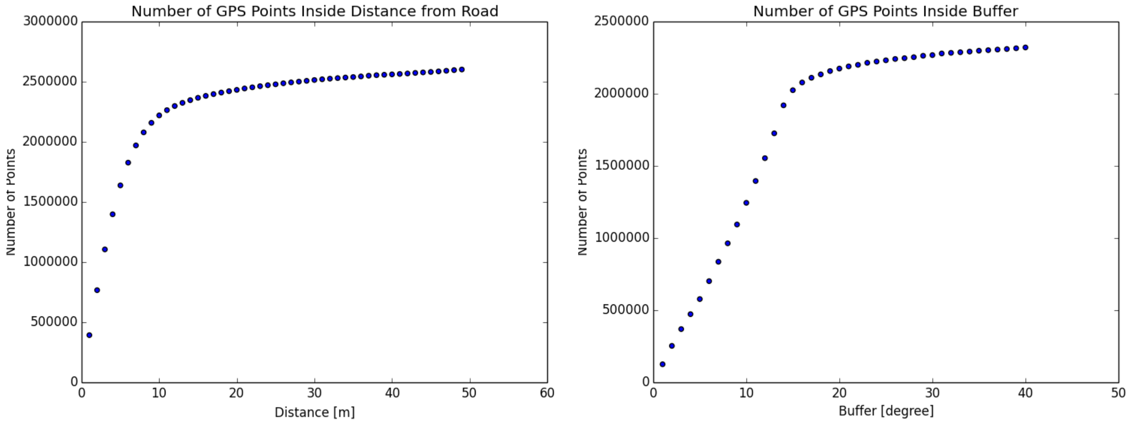
(b)

Figure 4.1: Problems with bicycle points(a) and access(b) with TrackMatching

x-axis. The Python script behind the plots counted all the selected points from the database at a certain buffer and increased the buffer in an iterative way.

The point where the gradient of the plot decreases and approaches near 0 can be considered as the turning point. From this turning point only small increases in the number of points occur per added unit of a buffer. These small increases consists mostly of outliers and therefore the turning point is most-likely the ideal buffer. This means for the distance the buffer should be around 15-20 meters and for the heading it should be around 20 degrees. However, in the case of the distance to road it is important to take into account the width of a road. There are large highways which consist out of 5 or 6 lanes, given the average width of a road is 3,5 meters (WegenWiki, 2014) these roads could have a width of 20 meters. A buffer of 15 meters would then be relatively small comparing to a road that consist out of one lane. To also cover these large highways, the chosen buffer is 30 meters. These two buffers, 30 meters and 20 degrees, are also considered the most suitable buffers by Zhang et al. in their paper discussed in Section 2.2.

The buffer for the distance is implemented using the ST_Distance_Sphere function in PostgreSQL, which calculates the distance between two geometries in meters. A column with the distance is added to the table, so a query containing distance<30 select all the points which have a distance of less than 30 meters to the road.



(a)

(b)

Figure 4.2: Plot of the number of points versus the distance to the assigned road(a) and difference in heading between the point and road(b)

The buffer for the heading is more complicated. First the direction of the road should be known. Roads are not drawn as straight lines in OSM, thus the smaller segments of the linestrings are used to calculate the direction of the road. Then, a column called "relativedirection" is added to determine whether a point heads in the similar direction of the drawing direction of the road or in the opposite direction. The tags 'similar' and 'opposite' are only given when the heading of the points is within 20 degrees of the direction of the road. All other points are given a NULL value and will not be used for further processing.

Finally, the distance buffer filtered 132.022 GPS points made by car and the heading buffer filtered 248.673 points, resulting in a dataset containing a total of 2.568.928 GPS points made by car.

4.3 Training Dataset

This Section describes how the training dataset is created. The training set is typically used for training the classifiers or rules. In the development phase, the training set can also be used to test and adjust the algorithms. The remaining data can be just for validation of the algorithm.

With the final dataset it is possible to derive a training set which can be used for training and testing the attribute extraction algorithms before applying the algorithm on the rest of the data. To derive this training set, the most used roads are selected. The most used roads are the roads that have the most GPS points assigned to them. The top 10 of this selection can be used for training, however the training data should be a representation of the total dataset. The training set would consist out of many roads of the same type if the top 10 most used roads would be used. Therefore, the most used roads of different highway types are picked. The resulting top 10 roads used for training can be seen in Table 4.1. The roads with the same names or references represent different segments and/or directions of that road.

Road ID	**# of GPS Points**	**Highway Type**	**Road Name**	**Road Reference**
"w.120708542"	18728	Secondary	Knardijk	N707
"w.6973056"	16551	Trunk	Gooiseweg	N305
"w.6967520"	13323	Secondary	Zeewolderdijk	N707
"w.89633056"	11166	Motorway		A1
"w.6971654"	10545	Trunk	Gooiseweg	N305
"w.189277435"	9670	Motorway		A1
"w.6971506"	9353	Secondary	Spiekweg	N705
"w.6977206"	8755	Motorway		A28
"w.6971882"	7037	Primary	Berencamperweg	N301
"w.177925552"	5946	Motorway		A12

Table 4.1: Top 10 roads based on the available GPS data and the diversity of roads

CHAPTER 5

Deriving Attributes

The first part of this Chapter discusses the derivation of attributes that are already standard in OSM, however still many of these attributes are not informed. The attributes that are already standard in OSM are whether the road is a one or two way road ("oneway"), speed limit ("maxspeed"), number of lanes ("lanes") and the access of different vehicles on the road ("access"). The methods to derive the attributes are explained per attribute.

The second part of this Chapter will focus on new attributes that are not yet implemented in OSM. These attributes are: the average speed ("averagespeed"), the hours in which a road is congested ("congestion"), the importance of a road ("importance") and the geometry error of a road ("geometryerror"). The extraction of these attributes are explained in the following Sections in this Chapter.

The attributes are classified using code lists which contain the values of the attributes. Different levels of the code lists can be implemented per attribute in a hierarchical structure, called a hierarchical code list. The hierarchy describes the level of detail and the granularity of the value of the attribute. Due to the complexity of some of the attributes, the hierarchies can provide different perspectives on the error of the attributes. The classification of the speed limits might be too complex and could lead to inaccurate results, however it might be possible to accurately classify groups of these speed limits which would be more acceptable. Depending on the attribute two or more levels will be added to some attributes.

5.1 One or Two Way Road

The "oneway" tag in OSM typically describes whether a road is accessible in only one direction or in both directions. This attribute is essential for navigational purposes, knowing if you are allowed in the street or not. If the road is a one way road, the tag will return "yes". When the road is a two way road, the tag will return "no" in its basic form. However, OSM also provides more detailed information. Therefore, different levels in the hierarchy will be introduced for this tag. At the first level (L0), the tag will return "yes" or "no". At the second level (L1), the tag can return "yes", "no", "-1" or "reversible". The hierarchical code list is depicted in Figure 5.1. A road gets the value "-1" when the road is a one way road, but in the opposite direction of the drawing direction. This is important for the rendering of the map to determine in which way the arrow should point. A road gets the value "reversible" when the road is a one way road of which the direction can be changed, e.g. during rush hours.

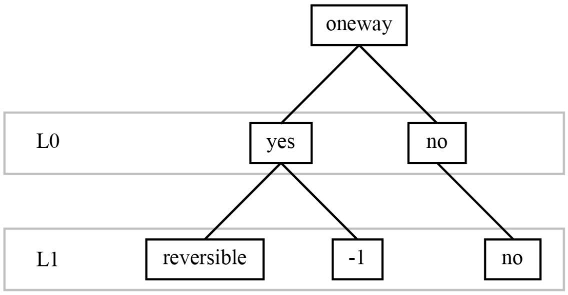


Figure 5.1: Different levels in the hierarchical code list for attribute "oneway"

The "relativedirection" column in the database will be used for deriving the "oneway" tag. This column returns "similar" or "opposite" referring to the heading of the GPS point compared to the drawing direction of the road. The algorithm selects and counts all the relative directions of a certain road and then calculates whether the road is a one or two way road. The calculation is as follows, the total number of "similar" (or "opposite") is divided by the sum of "similar" and "opposite" and if the result is bigger or smaller than a certain threshold the road is considered a one way road. Else, the road is a two way road. The alogrithm is trained using different thresholds and the threshold with the smallest error on the test data was bigger than 0.9 or smaller than 0.1. This means that if a specific relative direction is represented less than 10% the road is considered one way taking possible outliers into consideration. This threshold is reasonable since all GPS points during the day are taken into account and most people drive the same way from and to their destination resulting in a 50%-50% balance.

This method can be used to derive L0 for this attribute, however L1 would require some adjustments. Both "-1" and "reversible" are variations of a one way street and therefore fall in L0 under "yes". If the total number of "opposite" divided by the sum is bigger than 0.9, it means that the street is a one way street in the opposite direction thus a "-1" can be given to the tag. The algorithm for this level in the hierarchical code list can be seen in Algorithm 1.

For the reversible case this is more complicated. This could be detected by checking the time of the GPS points for both directions, the points of a reversible road are in two directions but cannot occur at the same moment in time. Roads which contain points in two directions that never occur at the same time, but that alternate during some intervals in time, are most likely to be reversible roads. The GPS data of this research makes it more difficult to detect reversible roads. The data is acquired by more than 800 people living in 3 different cities in the Netherlands. These people were tracked for a week while going to work, doing groceries, etc. The fact that these people are from only 3 different cities makes it possible that most of them take the same road to and from work at similar times during the day, which would influence the GPS data in a negative manner to detect the reversibility of a road. Therefore, and because of the lack of time in this research, deriving "reversible" is not taken into consideration for this research.

Deriving the one way attribute can also be used for bicycles, the same algorithm can be applied for the bicycle points. There is already a tag "oneway:bicycle" available in OSM for this attribute.

Algorithm 1 Attribute Extraction of "oneway"

```
function ONEWAY(roadid)
    select relative direction for roadid
    if $r \geq 10$ then                                  ▷ r is the number of results
        $similar = count$ "similar"
        $opposite = count$ "opposite"
        $sum = similar + opposite$
        if $similar/sum > 0.9$ then
            $oneway =$ "yes"
        else if $opposite/sum > 0.9$ then
            $oneway =$ "-1"
        else
            $oneway =$ "no"
        $update$ "oneway"
```

However, first the roads which allow bicycles have to be known to implement this algorithm to these roads.

5.2 Speed Limit

The "maxspeed" attribute provides the speed limit for every road. This attribute makes it possible to have the speed limits publicly accessible and in combination with the average speed, roads with a lot of speed violations can be detected. Deriving the speed limit depends on different factors. First of all there is the behavior of the drivers, all people act differently on the road which cause GPS traces to be different either. Then there is the temporal aspect, where speed limits can differ over time and congestions may cause the GPS traces to be different too. All these aspects, and many more, influence the data and makes it difficult to automatically derive speed limits. This Section will describe the steps that are taken to create an algorithm that extracts the speed limit out of GPS points.

The "maxspeed" attribute can be divided into several levels in the hierarchy as can be seen in Figure 5.2. First of all it should be stated that the speed limits that are used for this research are the most common speed limits in the Netherlands, these speed limits are:

- 30 km/h
- 50 km/h
- 60 km/h
- 70 km/h
- 80 km/h
- 100 km/h
- 120 km/h
- 130 km/h

L0 groups these speed limits into an ordinal scale, in this case low, medium and high. Low covers the speed limits of 30 and 50, medium covers 60, 70 and 80 and high covers 100, 120 and 130. The speed limits are divided in this way to make a distinction between the location of the roads. If the attribute returns low, it means that the roads are probably in an urban area where the maximum speed limit is 50. If the speed limit is higher than that, e.g. it returns medium, than the roads are in a rural area. If the attribute returns high, it means that the roads are motorways.

L1 represents the value of the speed limit of a certain road. L2 contains the same code list, however this level in the hierarchy makes a distinction between day (06:00 hours - 19:00 hours) and night (19:00 hours - 06:00 hours) (Rijkswaterstaat, 2014).

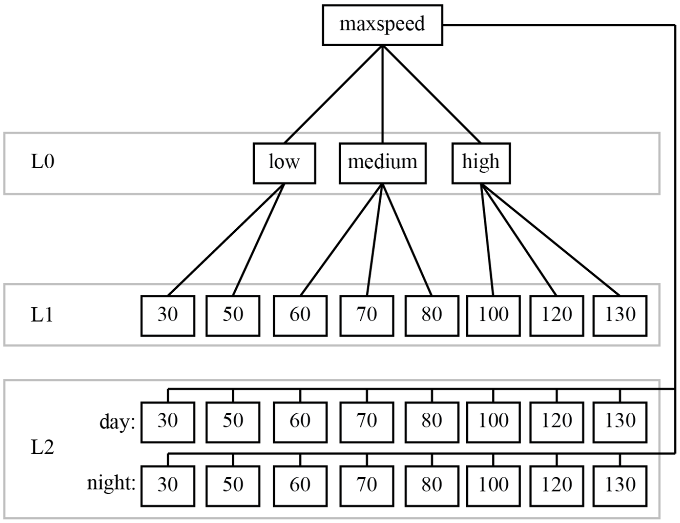


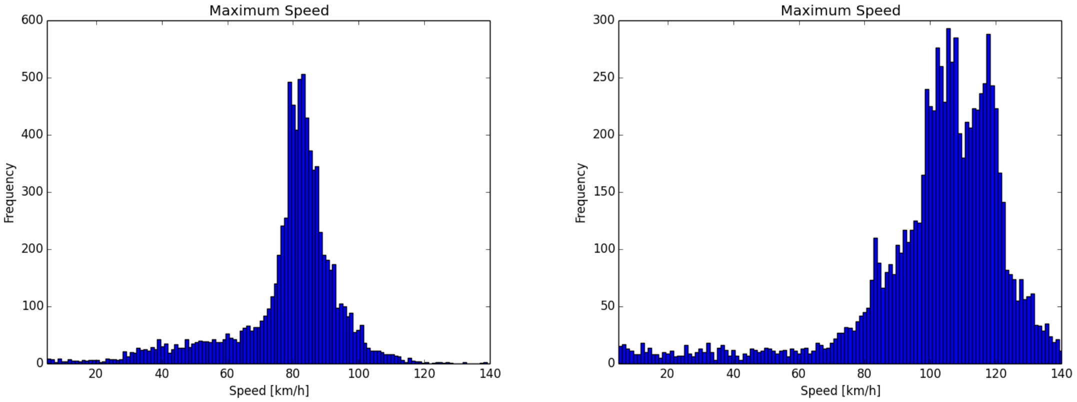
Figure 5.2: Different levels in the hierarchical code list for attribute "maxspeed"

When visualizing the speeds of GPS points of a certain road in histograms, it gives more insight into the driving behavior of people on different types of roads. Two histograms of different types of roads can be seen in Figure 5.3. The first histogram is from road "w.6971506" which is a secondary road. The second histogram is from road "w.6977206" which is a motorway. There is a clear difference in driving behavior between the two histograms. The first histogram representing the secondary road has a clear peak around 80 km/h, where the second histogram has two peaks around 100 km/h and 120 km/h. Also, the width of the histogram is bigger at the second histogram meaning that different people drive different speeds at that road. This is due to the differences in driving behavior between people. There are people that drive 100 km/h on a 120 km/h road, but there are also people that drive 130 km/h or 140 km/h on that same road.

There are also a lot of low speeds at both histograms which could influence the results of the extraction of the speed limit. These low speeds are probably caused by traffic lights, congestions or other forms of traffic disruption. Therefore it is important to take out the lower speeds which do not represent the maximum driving speed of the road. There are two methods to do this: acceleration or velocity change rate (VCR).

"Acceleration is the rate at which the velocity of an object changes over time" (Crew, 2008). Thus acceleration divides the change in velocity by the time taken: $a = \frac{\Delta v}{\Delta t}$. Usually, acceleration is expressed in m/s^2. If a speed decreases, it is called deceleration. A disadvantage of acceleration could be that it does not take into account the magnitude of the velocity.

VCR is a method that does take into account the magnitude of the velocity, i.e. a change of 20



(a) (b)

Figure 5.3: Histograms containing the speeds of the GPS points of road "w.6971506" (a) and road "w.6977206" (b).

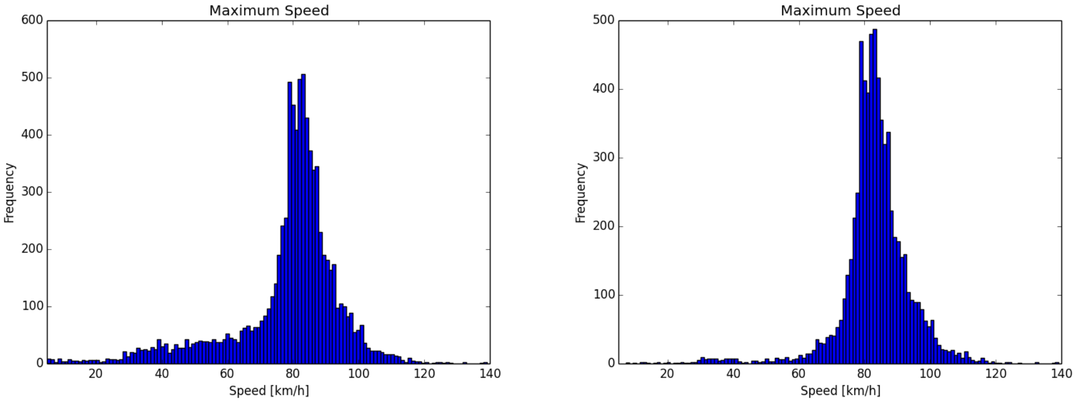
km/h for a speed of 100 km/h is different than the same change for a speed of 30 km/h. However small changes at low maximum speeds are more significant than small changes at higher speeds. VCR is calculated by dividing the change in velocity by the velocity of the first point (Zheng et al., 2010):

$$v_\Delta = \frac{v_{i+1} - v_i}{v_i} \tag{5.1}$$

The disadvantage of this method is that it does not take into account the time between two consecutive points. Therefore it is important that all the points have the same amount of time in between. Analysis of the data shows that 3.560.246 GPS points have a difference of 5 seconds, where only 91.808 GPS points have a different amount of time in between. Therefore it can be stated that it is better to use the VCR to detect outliers. An extra column in the database is created to calculate the vcr and store it in the database. If there would be more variety in time between points, a normalized version of the VCR could be developed. The difference between the two speeds could then be divided by the time between points to create a VCR per second.

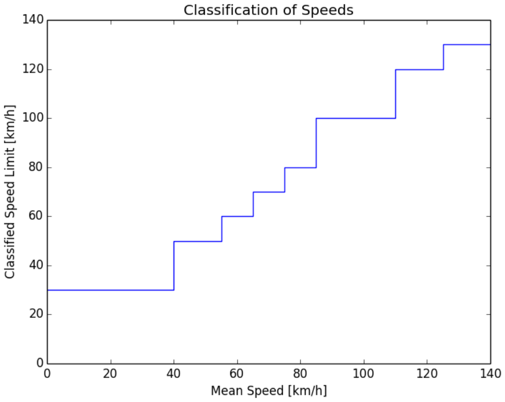
Next, a threshold need to be set for considering whether a point is an outlier or not. Research showed that 0.15 was the largest threshold which gave the best results. Therefore GPS points with a VCR higher than 0.15 or smaller than -0.15 are taken into account for deriving the speed limit. The result of these thresholds can be seen in Figure 5.4 where two histograms of road "w.6971506" show the difference before and after the taking into account the VCR. It is clear that a significant amount of lower speeds are removed from the dataset, which will cause the extraction of the speed limit to be more precise.

By looking at the previous histograms, it seems possible to derive the speed limit using the mean of the speeds. The mean of the speeds can be derived in Python by using the NumPy library. To assign a speed limit to the roads, a certain classification of speed limits need to be created based on the mean of the speeds. Research based on the training data shows that the classification according to Figure 5.5 works best. This research is performed using trial and error on the training data, the classification with the lowest classification error on the training data is considered best.



(a) (b)

Figure 5.4: Histograms containing the speeds of the GPS points of road "w.6971506" before (a) and after (b) taking into account VCR.



Mean **[km/h]**	**Speed Limit** **[km/h]**
< 40	30
> 40 and < 55	50
> 55 and < 65	60
> 65 and < 75	70
> 75 and < 85	80
> 85 and < 110	100
> 110 and < 125	120
> 125	130

Figure 5.5: Classification of speeds according to the mean showing the graph (a) and table (b)

However, for the higher speeds it proved more difficult to classify the correct speed limits. The training data tended to assign lower speed limits when the mean speed was higher than 100 km/h, e.g. 100 km/h speed limit for a road that actually has a 120 km/h speed limit. To solve this problem, only a percentage of the highest speeds is taken into account when the mean speed is higher than 85 km/h. A Python script calculated the errors in km/h for every percentage from 1 to 40 and plotted the results in a graph (see Figure 5.6). This result shows that 18%, 19% and 20% are the three best percentages to choose, further investigation showed that 20% was the best percentage to use for deriving the speed limits above 85 km/h.

Further investigation shows that only using the mean speed for speed limits lower than 85 km/h and only using the mean of the percentage highest speeds for speed limits higher than 85 km/h is

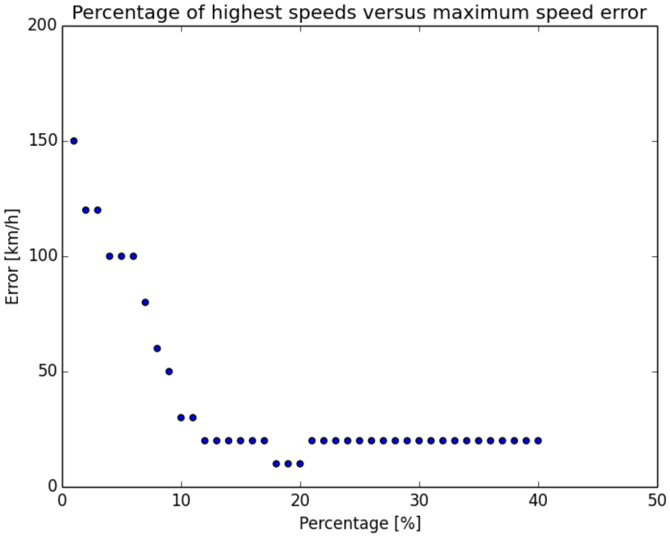


Figure 5.6: Graph showing the percentage used for the highest speeds versus the error accompanied with it

not sufficient. The algorithm tended to assign lower speed limits to the roads. Therefore a mix of the two means is created. For speed limits till 80 km/h, the average of the mean speed (v) and the mean of the percentage of highest speeds (v_{high}) is used: $\frac{v+v_{high}}{2}$. For speed limits higher than 80 km/h, the mean of the percentage of highest speeds is weighted twice as much as the mean speed: $\frac{v+2v_{high}}{3}$. This is because the higher speeds are more representative than the lower speeds for high speed limits. These values are trained and tested on the training data to determine the best weight for this classification.

Since the extraction of the speed limit is a typical classification problem, it is a possibility that the speed limit can be derived using pattern recognition methods. Therefore different classifiers are used to classify the speed limits. Multiple classifiers have been trained on different feature sets. First, the classifiers need to be trained. Thus training data is needed for every speed limit. For every speed limit the five most used roads are selected. The classification will be performed on the Cumulative Distribution Function (CDF) of the speeds of the road. CDFs are more stable than Probability Density Functions (PDF). The values in a PDF vary much from each other from lower noise values to higher noise values. Comparing this to other PDFs can give very different results. CDFs typically are more stable and compensate noise and small changes along the way, which is better for comparison and classification. An example of the CDF of a road can be seen in Figure 5.7.

The training of the different classifiers will be performed on different feature sets. Classifiers react differently on the size of the feature set, therefore they are trained on four different feature sets. The first set consists of the CDF with a bin size of 10 for the speeds between 0 and 140, resulting in 14 dimensional feature set. The second set uses a bin size of 5, resulting in a 28 dimensional

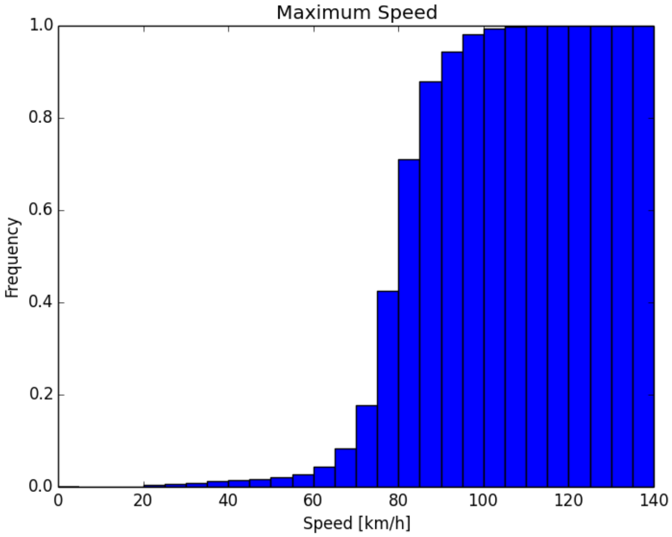


Figure 5.7: Example of a CDF of road "w.120708542"

feature set. The third set uses a bin size of 2 and the final set a bin size of 1. Resulting in a 70 and 140 dimensional feature set respectively. Different classifiers, both linear and non-linear, are trained and tested on these feature sets. The machine learning Python (mlpy) libary is used to be able to use the classifiers in Python.

Investigation of suitable classifiers on a small test set showed that the Support Vector Machine (SVM) and Classification Tree were the classifiers with the lowest error. SVMs typically looks for the separating hyperplane with the largest margin in the case of a linear separable classification problem (Burges, 1998). The goal of a Classification Tree is to find the best way in which the test data can be classified into the according classes. All possible solutions will be computed and the solution with the highest probability is chosen (Theodoridis and Koutroumbas, 2008). The feature set that suited these classifiers the best was the 28 dimensional feature set with the according bin size of 5 km/h. Different Support Vector Classifiers (SVC) were tested of which the nu-SVC was the classifier with the lowest error in combination with a linear kernel. This classifier also outperformed the Classification Tree and therefore was used to test on a bigger set. However, the SVC could not outperform the rule created earlier in this Section and is therefore not implemented in the final system. The final algorithm can be seen in Algorithm 2.

The different speed limits between day and night of L2 are not taken into account for this research. The classification of L1 was already time consuming, therefore not enough time for L2 was left. Most of the algorithm would remain the same, using the speeds between 06:00 hours and 19:00 hours for the speed limit during the day and 19:00 hours till 06:00 hours for the speed limits during nights. However, this level in the hierarchy would also require extra investigation on the driving behavior of people during the night. People tend to drive differently during the night which would require a different method for classification.

Algorithm 2 Attribute Extraction of "maxspeed"

```
function MAXSPEED(roadid)
    select speeds for roadid
    if $r \geq 50$ then                                      ▷ r is the number of results
        $derive$ mean $from$ speeds
        SPEED LIMIT(mean,speeds)
        $update$ "maxspeed"

function SPEED LIMIT(mean, speeds)
    if $((v + h)/2) \leq 40$ then                    ▷ v = normal mean, h = high speeds mean
        $maxspeed = 30$
    else if $((v + h)/2) > 40$ and $((v + h)/2) \leq 55$ then
        $maxspeed = 50$
        etc...
    else if $((v + h)/2) > 85$ then
        if $((v + 2h)/3) > 85$ and $((v + 2h)/3) \leq 110$ then
            $maxspeed = 100$
            etc...
    return $maxspeed$
```

5.3 Number of Usable Lanes on the Road

This Section will discuss the possibility of lane detection in the case of this research. This attribute can be useful for traffic analysts and civil engineers to acquire the capacity of the road or simply for more detailed map making. The problem with GPS data for lane extraction is the inaccuracy, noise and placement of the GPS device with respect to the lanes. As can be seen in Figure 5.8, no clear distinction can be made between different lanes just by looking at the tracks projected on the map. By using a histogram with the assumption that the lanes are normally distributed, a script could extract the number of lanes out of the Figure, e.g. in a similar way as Chen and Krumm did in their paper. Before the actual lane detection can be performed, some preprocessing on the data have to be performed. These preprocessing steps will be discussed first, after which the lane detection is explained.

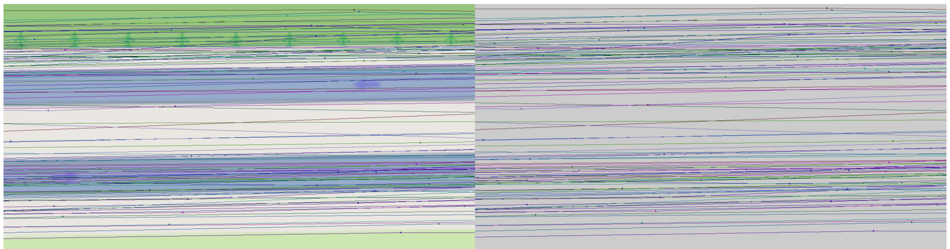


Figure 5.8: Multiple GPS tracks with and without map background

5.3.1 Preprocessing

The distribution of the GPS points is essential for lane detection. To reveal this distribution, a histogram can be created to visualize it. For this, distances from the GPS points to the centerline of the road are needed, but also on which side of the road that point lies.

The distance can easily be calculated using the ST_Distance_Sphere function in PostGIS for the geometry of both the road and the GPS point. This function returns the distance in meters, where ST_Distance returns the distance in degrees instead of meters. An extra column in the table can be added to store the distances between points and roads.

Next, now that the distance is known, it is important to determine on which side of the road a points lies. This can be determined using the cross product. If a line goes from A to B and point is P, then the cross product of this can be written as:

$$v = (Ay - By) * Px + (Bx - Ax) * Py + (Ax * By - Bx * Ay) \tag{5.2}$$

where for v>0 the point is left of the line, v<0 the point is right of the line and for v=0 the point is on the line or in front/behind the line.

The cross product formula (Equation 5.2) was integrated in a SQL query, adding a column 'sideofroad' containing either 'R' for Right, 'L' for Left or 'NULL' for points exactly on the line. After running the query on the dataset, the results were not satisfying. Many of the roads contained points that were mostly right of the road or mostly left of the road. Analysis turned out that the geometry of OSM roads are linestrings and not straight lines. The cross product formula only uses the starting and the end point of a line, therefore working under the assumption that the line is a straight line. The result of this can be seen in Figure 5.9, most GPS points in this Figure will be classified as right side of the road.

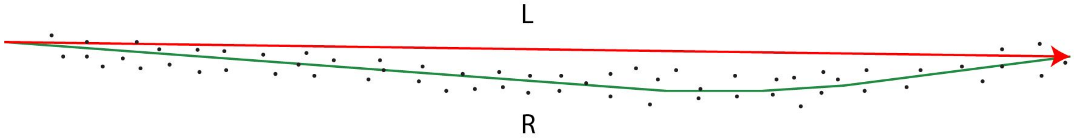


Figure 5.9: Cross product line (red) classifies GPS points (dots) mostly to the right side of OSM road (green)

To solve this problem, linestrings are cut into smaller straight line segments. Every point is then matched to the closest segment, after which the cross product query can be performed. This query results in a classification of the side of road for all 3.660.561 GPS points. Using the side of road, a 'relative' distance column can be created. The relative distance is the distance used to create a histogram of GPS points on a road. The difference between the normal distance is that it uses the side of road to add a negative value to the distance of points that are left of a road. In this way, one can determine immediately if a point is left or right of a road by looking at the relative distance.

5.3.2 Lane Detection

After the preprocessing phase, the data can be visualized for analysis. Like in the paper of Chen and Krumm, histograms are used to visualize the GPS points with respect to the centerline of the OSM road. Figure 5.10 shows the histogram of road "w.89633056".

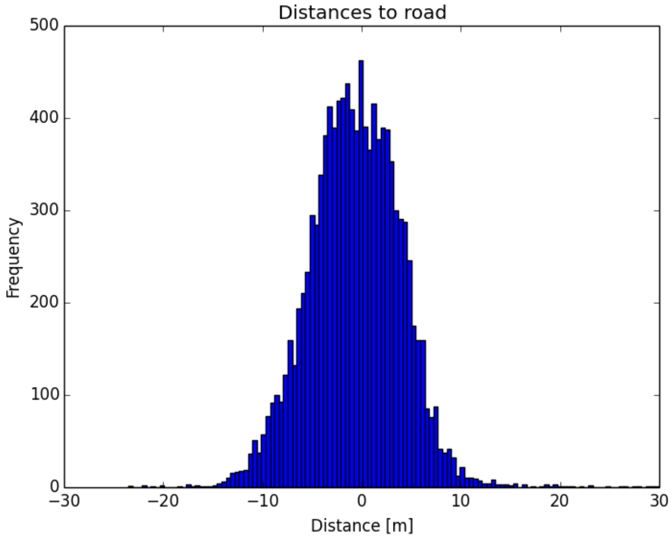


Figure 5.10: Histogram showing the distances from GPS points to the centerline of the road

This histogram shows the distances from the GPS points to the centerline of the road on the x-axis and the frequency of GPS points that have the according distance on the y-axis. Most points seem to be centered around $x = 3$. This can mean that the GPS points have a slight error and/or the OSM road has an error. Even by eye, it is difficult to detect the number of lanes out of these histograms. The standard lane width in the Netherlands varies between 2.5 meters and 4.5 meters depending on the road type (WegenWiki, 2014). By using the lane widths and the standard deviation of the histogram, one could think that the number of lanes can be derived by using the entire road width. To do this, a Gaussian distribution is drawn in the histogram (Figure 5.11 and the mean and standard deviation are calculated.

As can be seen in Figure 5.11, the mean distance from the GPS points to the centerline of the road is 0,813 meters and the standard deviation is 4.587 meters. The actual number of lanes for this road is 2, which is plausible considering a lane width of 4.5 meters and the standard deviation of 4.587 meters. However, Table 5.1 shows a table containing all roads from the top 10, their actual number of lanes and the calculated standard deviations. If a road has two standard deviations, it means that it is a two way road with a lane(s) in one direction and a lane(s) in the opposite direction. The number of lanes column then counts for one direction, so one standard deviation.

There is no clear correlation between the number of lanes and the standard deviations of the data. The standard deviation of roads with only one lane vary from 2.406 to 5.702 with an average of 3.571, where the standard deviation for roads with two lanes vary from 3.176 to 4.587 with an average of 4.027. Roads with two lanes have a slightly higher standard deviation, but unfortunately there is not enough distinction between the standard deviation of one and two lane roads to make a good classification. The method of Zhang et al., as explained in Section 2.2, can also not be aplied here. By applying their threshold of 5.5, almost all roads will be assigned at least 2 lanes except for road "w.6971882". Also for other thresholds it is impossible to correctly detect the number of

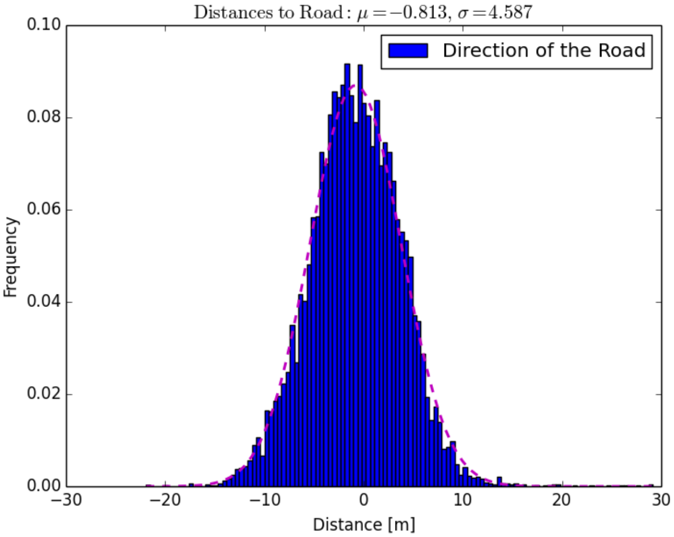


Figure 5.11: Histogram including the Gaussian distribution

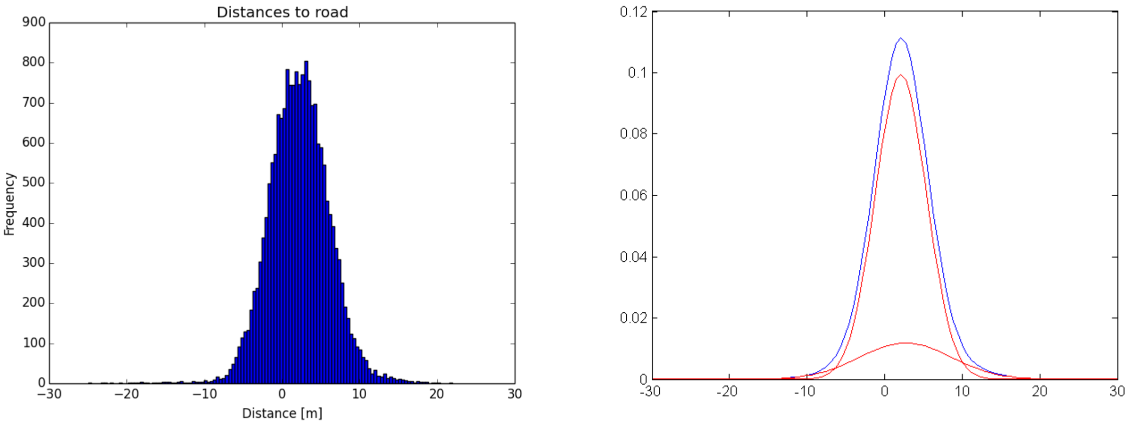
Road ID	Number of Lanes	Standard Deviation
"w.120708542"	1	3.453 and 3.799
"w.6973056"	1	3.233 and 2.901
"w.6967520"	1	2.802 and 3.090
"w.89633056"	2	4.587
"w.6971654"	1	3.850 and 3.476
"w.189277435"	2	3.929
"w.6971506"	1	5.655 and 5.702
"w.6977206"	2	3.176
"w.6971882"	1	2.406 and 2.488
"w.177925552"	2	4.415

Table 5.1: Top 10 roads with the number of lanes and the standard deviation of the histogram of distances

lanes for all roads.

According to Chen and Krumm, Gaussian Mixture Models (GMM) can detect lanes from GPS data. "A GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities" (Reynolds, 2009). Implementing GMM on the data of this thesis, did not result in lane detection. For example, the Gaussian components for road "w.120708542" when applying a GMM with two components have a mean of 2.6944 and 2.1026. This means that the mean of the two components are basically the same, resulting in no clear detection between lanes. Figure 5.12.

If the GMM would work correctly, the different lanes would be detected and the Gaussian com-



(a)

(b)

Figure 5.12: Histogram(a) and according GMM(b) with two components for road "w.120708542"

ponents would follow the Gaussians as depicted in Figure 5.13. However, the GMM in this research was not able to reproduce the method by Chen and Krumm.

When comparing the GMM and histogram of Chen and Krumm of Figure 2.4 in Section 2.2 to the GMM and histogram derived for this research, there are many differences. At first, the histograms are different from each other. The histogram derived for this research is more dense and follows the distribution of a Gaussian, where the histogram of Chen and Krumm contains gaps and a distinction between the two lanes can already be made by eye. Therefore their GMM is able to detect the different lanes. The exact reasons for this difference in data cannot be given, however there are some causes that might influence the data in a way that it is not possible to detect the number of lanes using GMMs.

There are multiple problems that influence the data, these problems are related to GPS, OSM and the behavior of people. The GPS related problems are inaccuracy and placement of the GPS devices. Different errors resulting in GPS inaccuracy have already been explained in Subsection 3.1.1. The placement of the GPS devices is also important for an accurate result. The placement of the GPS device is dependent to the behavior of people. There is no standard position of the small GPS device in a vehicle, thus the relative position to the road within the same lane may considerably deviate. Some people might place the device on the right seat of the car, where other people might place it on the left seat of the car. This will influence the result already by 1 or 1.5 meters approximately.

The behavior of people when driving also influences the data. People that often change lanes will create a completely different GPS track than people who will stay on one lane the entire road. If the latter is the case for all drivers on a road, than the frequency of usage of lanes also influences the data. The data will then be more centered towards the crowded lane of the road and the other lane would have a smaller peak in the histogram.

Finally, OSM also influences the data. The way of drawing, as already introduced in Section 2.2, can influence the data in different ways. First, it is possible that the representation of the road

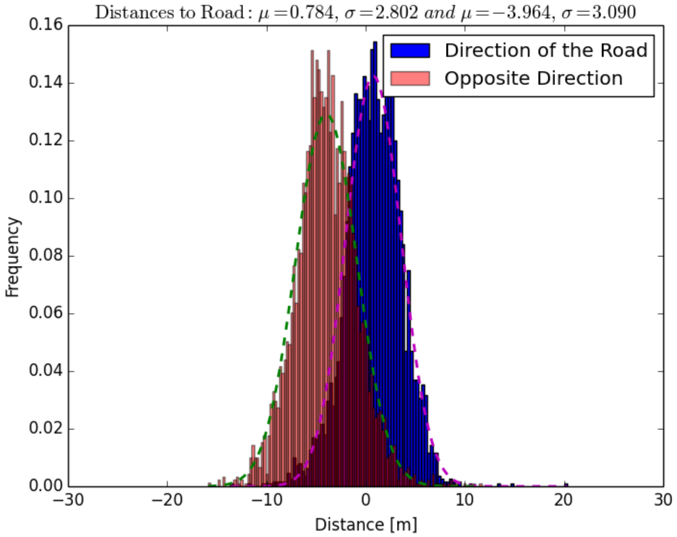


Figure 5.13: Histogram of both lanes of road "w.120708542" including the Gaussians for both lanes

in OSM is not that accurate and represents a road that is not completely parallel to the actual road. For example, if the start of the road differs -2 meters with the OSM data and the end of the road differs +3 meters then the result will be a shifted line on top of the actual road which would influence the relative distances from the GPS points to the centerline of the OSM road. Also, the roads in OSM should be drawn in such a manner that the number of lanes for one road cannot vary.

All these causes can influence the data in a way that it is almost impossible to detect the number of lanes by using GPS tracks. For this research, it was not feasible to reproduce the methods of Zhang et al. and Chen and Krumm in a way that a reliable result is achieved.

5.4 Access of Different Types of Vehicles

Not all types of vehicles are allowed on every road, therefore it can be useful to know which type of vehicles are allowed on a road and which are not. Navigational systems can then determine different routes for different types of vehicles for example. OSM is often used for bicycle navigation applications, however for most roads it is not known whether the bicycle is allowed on a normal road or not.

Most different types of vehicles, besides car, have their own roads or tracks in OSM, e.g. trains, trams and subways fall under the "railway" tag. Even buses do not always drive on the road. Also the relatively small amount of data on bus tracks will not cause a reliable result. However, for bicycles it might be able to derive whether they have access to a road or not.

Unfortunately, the TrackMatching service does not provide matching to cycleways, which makes it harder to derive information about the access of bicycles to a certain road. All bicycle points will now be matched to motorized road networks. If the points could be map matched to cycleways too, it would solve the entire problem and bicycle points would be matched to motorized roads if they would actually drive on that road. Now, only an assumption can be made whether a bicycle point is on a road or not. The idea is to map match the bicycle points to the motorized road network, calculate the distance between the point and the road and find the closest cycleway. With these indicators it should be possible to detect whether a bicycle is allowed on a road or not.

The map matching is done in a similar way as the method explained in Section 4.1, only now for points that are made by bicycle. The result is again stored in a database and similar preprocessing steps are performed.

Next, for every point the nearest cycleway is assigned within a certain threshold. This query looks for all cycleways within a certain area around the GPS point and assigns the cycleway with the smallest distance to the GPS point as the nearest cycleway. After the assigning of the nearest cycleway, the same preprocessing steps to calculate for example distance and relative direction as for motorized roads are calculated and added to the database.

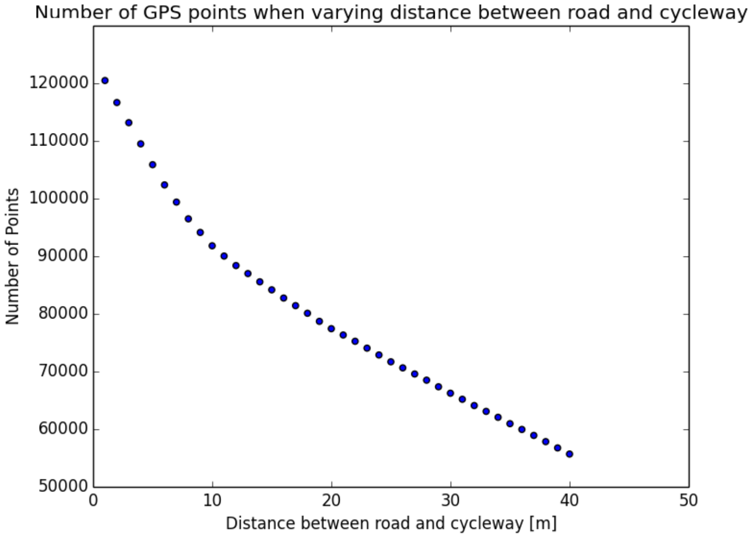


Figure 5.14: Graph showing the number of points that are selected for the different distances between road and cycleway

Now that for all the points a motorized road and a nearest cycleway, provided that that there is a cycleway in the vicinity, is assigned, the algorithm to detect whether a road allows bicycles can be run. This algorithm takes into account a couple of factors. At first the bicycle GPS point have to comply with the same thresholds as the GPS points made by car, i.e. a speed higher than 5 km/h, the distance to the road should be less than 30 meters and the heading of the point should be similar or opposite to the drawing direction of the road. Secondly, the distance to the road should be smaller than the distance to the cycleway. At last, the difference between the distance to the road and the distance to the cycleway should be bigger than 10 meters. The latter should be incorporated into the system due to points that are slightly closer to the road than the cycleway, but actually are on the cycleway. The threshold of 10 meters is chosen with the help of Figure 5.14 which shows a graph of the number of points that are selected for every different distance between the road and the cycleway. The graph shows a nod at 10 meters and after it the graph gradually decreases, 10 meters can be considered a significant distance together with the other thresholds to state that the point is on the road and not on the cycleway. Therefore, roads containing these points can be stated as roads that allow bicycles.

The first algorithm that is used is a SQL command updating all roads that apply to the rule that has been created. The SQL command is written below:

```sql
UPDATE karlmsc.osm_highways_upd
SET access = 'bicycle'
FROM (
    SELECT assigned_roadid
    FROM (
        SELECT assigned_roadid, COUNT(*) AS cnt
        FROM karlmsc.pointsroads_bicycle
        WHERE road_distance < cycleway_distance
        AND (cycleway_distance - road_distance) >= 10
        AND calcspeed > 5 AND road_distance < 30
        AND (relativedirection_road = 'similar' OR relativedirection_road = 'opposite')
        GROUP BY assigned_roadid
        ) AS d
    WHERE d.cnt >= 10
    ) AS c
WHERE karlmsc.osm_highways_upd.id = c.assigned_roadid;
```

The second algorithm is a SQL command updating all roads that have no cycleway in the vicinity. It should be noted that both commands are integrated in Python. The second algorithm is written below:

```sql
UPDATE karlmsc.osm_highways_upd
SET access = 'bicycle'
FROM (
    SELECT assigned_roadid, COUNT(*) AS cnt
    FROM karlmsc.pointsroads_bicycle
    WHERE assigned_cyclewayid IS NULL
    AND calcspeed > 5 AND road_distance < 30
    AND (relativedirection_road = 'similar' OR relativedirection_road = 'opposite')
    GROUP BY assigned_roadid) AS c
WHERE karlmsc.osm_highways_upd.id = c.assigned_roadid
AND cnt >= 10;
```

These two SQL statements are implemented into Algorithm 3. Two functions are used to send the statements to the database and update the "access" column. The Python library that is used for connecting and executing to the database is the Psycopg2 library. First the SQL statement is stored as a variable. Then a connection is made to the database using the host, database name, user and password. Next, the statement is executed and finally the connection with the database is closed.

Algorithm 3 Attribute Extraction of "access"

```
function UPDATE_ACCESS_1
    statement = "UPDATE..."
    connect to database
    execute statement
function UPDATE_ACCESS_2
    statement = "UPDATE..."
    connect to database
    execute statement
```

5.5 Average Speed

The average speed is relatively easy to calculate, but is used much in navigation systems. It can be used for finding the fastest route to a certain destination. Also, Dutch news station Nieuws (2014) writes that Dutch police use average speed data from Dutch navigation system provider TomTom to locate the best roads for speed traps. The average speed can be calculated per road, but it can also be more detailed. L0 of "averagespeed" is the averagespeed of a road in both directions. L1 is the average speed in the similar direction as the drawing direction and in opposite direction, in case of a two way road. This will increase the knowledge of how fast a route is in a specific direction. Finally, L2 will provide the average speed per hour to take into account deviations of speed during the day for more detailed information. All the levels in the hierarchy are depicted in Figure 5.15.

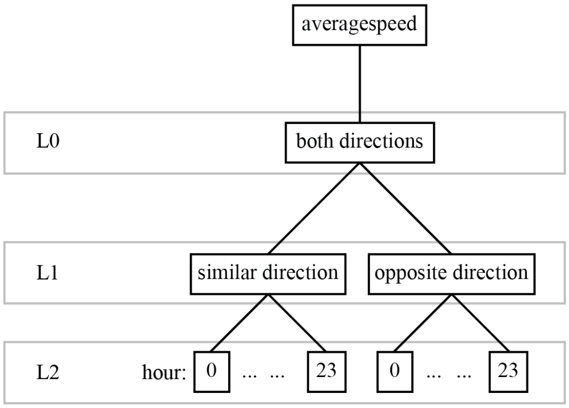


Figure 5.15: Different levels in the hierarchical code list for attribute "averagespeed"

The averagespeed can be calculated by selecting all the speeds of GPS points of a certain road and dividing them by the total number of GPS points. For L0, all the points of a road can be selected. For L1, only points with the relative direction 'similar' or 'opposite' can be selected and for L2 all the points with the relative direction and a specific hour can be selected. The algorithm used for L1 can be seen in Algorithm 4.

For this research, only L0 and L1 will be stored in the database. This research has no direct benefits from also storing the average speed per hour. However, the average speed per hour will be

Algorithm 4 Attribute Extraction of "averagespeed"

```
function AVERAGESPEED(roadid)
    select speed for roadid in "similar" direction
    select speed for roadid in "opposite" direction
    select "oneway" for roadid
    if $oneway$ = "yes" then
        if $s \geq 10$ then                                  ▷ s is the number of "similar" results
            update "averagespeed_similar" $from$ similar speed
    else if $oneway$ = "-1" then
        if $o \geq 10$ then                                  ▷ o is the number of "opposite" results
            update "averagespeed_opposite" $from$ opposite speed
    else if $oneway$ = "no" then
        if $s \geq 10$ then
            update "averagespeed_similar" $from$ similar speed
        if $o \geq 10$ then
            update "averagespeed_opposite" $from$ opposite speed
```

used for the congestion attribute in the next Section.

5.6 Hours in which Congestion Occurs

Taylor et al. (2000) provides multiple definitions for congestion, but the definition that is subsequently proposed for use in traffic studies is:

> Traffic congestion is the phenomenon of increased disruption of traffic movement on an element of the transport system, observed in terms of delays and queuing, that is generated by the interactions amongst the flow units in traffic stream or in intersecting traffic streams. The phenomenon is most visible when the level of demand for movement approaches or exceeds the present capacity of the element and the best indicator of the occurrence of congestion is the presence of queues [...] it recognizes that the capacity of an element in a traffic systems may vary over time, e.g. when traffic incidents occur or for minor stream traffic movements where capacity may depend on the traffic volume in the major stream. (Taylor et al., 2000)

This means that a disruption in normal traffic situations, e.g. traffics jams, can be classified as congestion. In the case of this research, normal traffic situations can be described as the average speed and disruptions are basically the deviations from this average speed. This Sections describes the approach to derive the hours in which congestion occurs to acquire knowledge about which roads to avoid at certain hours. Also, this data can be interesting for traffic analysts and civil engineers.

The different levels in the hierarchical code list for this attribute are at L0 the difference between congestion during the week and congestion during the weekend. Congestion tend to happen more often during weekdays, because of the working hours of inhabitants. L1 separates the congestion per day of the week. The levels in the hierarchical code list are depicted in Figure 5.16.

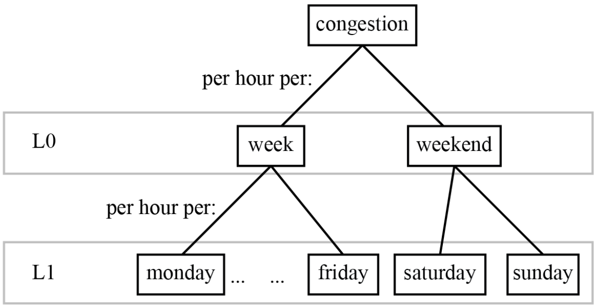


Figure 5.16: Different levels in the hierarchical code list for attribute "congestion"

There are two ways to derive the hours in which congestion occurs. One method is to compare the average speed of a specific hour to the average speed of the road and the other method is to use the VCR to detect a significantly higher amount of stopping or slowing down of the car compared to the normal behavior.

For this research the average speed will be used as an indicator of congestion, because this method achieves the goals of congestion more directly. As already explained, congestion is the disruption of the traffic. The congestion attribute is created to detect disruptions of traffic so they can be avoided, the main reason to avoid congestions is to take a faster route. Therefore using the average speed will directly apply to the main reason, where using VCR would indirectly relate to the speed issue. VCR detects the slowing down of the car, but this could have multiple reasons. Traffic lights, differences in speed limits and more could cause the VCR to fluctuate making it not as reliable for detecting congestions as the average speed.

First of all, it should be noted that the congestions are calculated per driving direction. This is important because it is possible that in one direction there is no congestion, but in the opposite direction there is a lot of congestion during a certain moment in time. This means that the for the opposite direction a different route might be more beneficial, but not for the other direction. Thus for every road in both directions the average speed will be derived and the average speed per hour will be calculated. Next, a percentage of the average speed will be derived as a congestion indicator. In this case, 10% seemed a reasonable percentage to detect a significant difference between the average speed of the road and the average speed per hour. The average speed per hour will then be subtracted from the average speed and compared with the 10% of the average speed of the road. If the difference between the average speed and the average speed per hour is higher than the percentage, that hour of the road will be considered congested. The Python script will calculate this for every hour of the day in an iterative way and will add the congested hours to a string which can be updated to the database once the script finishes all hours of the day. The final algorithm can be seen in Algorithm 5.

Algorithm 5 Attribute Extraction of "congestion"

```
function CONGESTION(roadid)
    select averagespeed_similar from roadid                                        ▷ is called $s$
    select averagespeed_opposite from roadid                                       ▷ is called $o$
    for every hour of the day do
        select speed for roadid during $hour$ in week in similar direction          ▷ is called $hs$
        select speed for roadid during $hour$ in week in opposite direction         ▷ is called $ho$
        if $hs \geq 10$ and $s$ != None then
            $ahs$ = average $hs$
            if $s - ahs \geq 0.1s$ then
                add $hour$ to $hour\_similar$
        if $ho \geq 10$ and $o$ != None then
            $aho$ = average $ho$
            if $o - aho \geq 0.1o$ then
                add $hour$ to $hour\_opposite$
    update congestion_hour_similar
    update congestion_hour_opposite
```

5.7 Importance of a Road

Rankings of roads can be based on the type of roads, but this does not take into account the actual usage of the roads. However, the distribution of GPS points can give an indication of the usage of roads. Therefore, importance in this research can be described as the usage of roads compared to the total usage of a country, state or province. This can be used by urban developers and traffic analysts to easily detect important roads and to use it to their advantage or, e.g. increasing the number of lanes at important roads which are also congested to improve the flow on that road.

There are multiple ways to derive the "importance" attribute, the basic idea is to derive a percentage per road compared to the total usage of the GPS points. The following units can be used to derive the importance:

- Number of GPS points on a road
- Time spend on a road by GPS points
- Number of passes on a road

The first unit is perhaps the most simple one, the more people drive on a road the more GPS points that road contains. However, this unit also has a disadvantage: the speed. The speed influences the number of GPS points that are left on a certain road. For example, if someone drives a 2 kilometer road with a speed of 100 km/h and the GPS device acquires points every 5 seconds, it takes 72 seconds to drive the road which means approximately 14 GPS points on that road. If the speed on the same road was 50 km/h, it would take twice as long to drive the road which means that the GPS device would acquire around 28 points in 144 seconds. This would mean that in the case of using the number of GPS points on a road, the road with lower speeds would be classified relatively higher compared to the actual usage of the road.

By taking the time spend on the road, the result will also be biased. Take, for example, motorways which are congested often. These roads will then have more time spend on them, while they are not actually used more often. So also the time unit does not represent the usage of roads well.

The number of passes could give a better, unbiased result compared to the previous two units. A pass is counted when a vehicle is on a certain road. If it leaves the road and returns after a while, a new pass is counted. With this method it is possible to actually calculate the usage of a road and compare it to other roads in the country, state or province.

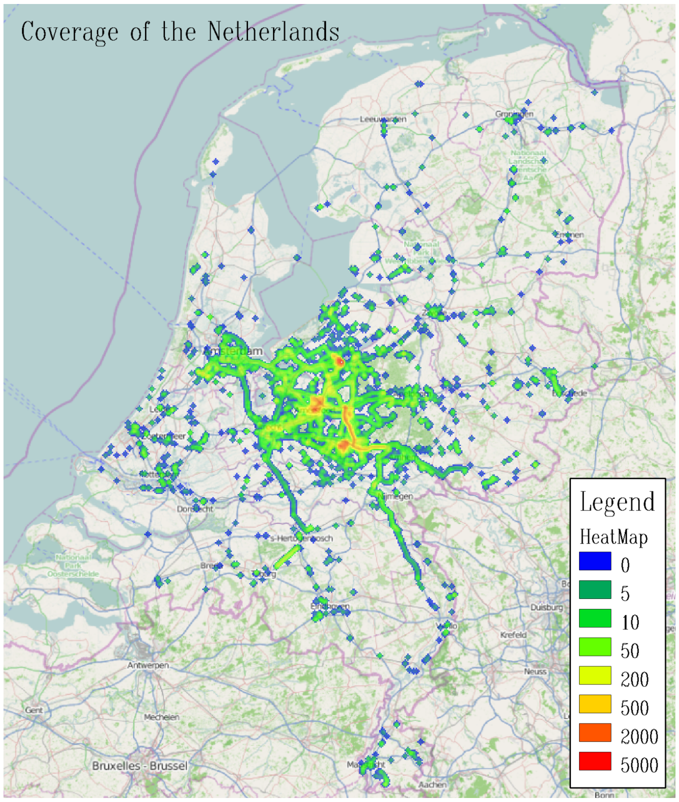


Figure 5.17: Heat map from sample of GPS points from the dataset showing the Netherlands is not fully covered

Finally, if the number of passes of all roads are calculated a classification can be performed. This classification depends on the data and the amount of passes that are made. A classification in this case is subjective and can be performed in different kind of ways. The classification depends on the total number of passes available and is therefore variable, a constant classification cannot be created. This means that the classification is variable and should be obtained based on the total number passes or better, based on the road with the maximum number of passes. In total, five different categories will be derived with 1 as the highest importance and 5 as the lowest importance. Then if, for example, the road with the maximum number of passes has 5000 passes every class will add 1000 passes. Thus, class 5 will contain all the roads with 0 to 1000 passes and class 4 will contain all the roads with 1000 to 2000 passes and so on. In this way a hierarchy between the different roads will be created. The complete algorithm can be seen in Algorithm 6

Algorithm 6 Attribute Extraction of "importance"

```
function CATEGORIES
    select MAX(roadpasses)
    c = rounded max roadpasses / 5                              ▷ c is category steps
    $c5 = c$, $c4 = 2c$, $c3 = 3c$, $c2 = 4c$, $c1 = 5c$
function IMPORTANCE(roadid)
    select roadpasses for roadid                                ▷ is called r
    if $r \leq c5$ then
        update category 5 for roadid
    else if $r \leq c4$ then
        update category 4 for roadid etc...
```

This method for deriving the importance of a road has one major disadvantage, the coverage of the GPS data. The GPS should be evenly distributed in a way that they represent the actual situation. The GPS data used for this research contains tracks derived from people living in Amersfoort, Veenendaal and Zeewolde. This causes the data to be centered to these cities and is therefore not evenly distributed in the Netherlands. Figure 5.17 shows a heat map of a sample of the GPS data used for this research. It clearly shows that the data is centered in the middle of the Netherlands and therefore shows that the data is not suitable for deriving this attribute for the Netherlands. However, it is possible to create a bounding box around one of the cities and apply the algorithm on that part of the Netherlands to test it.

When zooming on the three cities as can be seen in Figure 5.18, one can clearly see that the red parts in the heat map are centered around Zeewolde, Amersfoort and Veenendaal. Also, the structure of the motorways can be seen from the data.

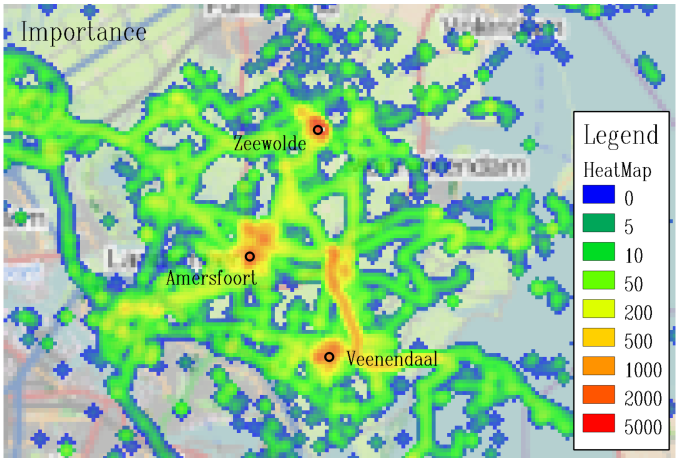


Figure 5.18: Heat map from sample of GPS points from the dataset showing most GPS points center around the three cities and their motorways

5.8 Error of the Road Geometry

The error of the geometry of road networks are usually assessed compared to official datasets. However, it can also be done with a different method using only GPS points and the OSM road network. This attribute can therefore be seen as a quality check and can be used to improve the quality of the map.

In the past, multiple comparative studies have been performed to assess the quality of OSM geometry. Already referred to in previous Sections, Haklay (2010) compared the OSM road network to United Kingdom's Ordnance Survey to assess the quality of the road network. Bhattacharya did her MSc Thesis in Geomatics on "Quality assessment and object matching of OSM in combination with the Dutch topographic map TOP10NL" (Bhattacharya, 2012). Ciepłuch et al. used Google Maps and Bing Maps to compare to OSM data and Fan et al. (2014) used the German Authority Topographic-Cartographic Information System (ATKIS) to detect errors. Zielstra and Zipf assessed the quality of OSM using the commercial TeleAtlas dataset. All these research projects use official datasets to detect errors in the OSM road network, this thesis will present an alternative to these methods.

The idea is to use the GPS points as a geometrical indicator of the road. There are multiple research projects available where road networks are extracted or improved from GPS points, however not to use the GPS points as a quality indicator. For example, the research mentioned in Section 2.2 by Zhang et al. improves existing road data from GPS traces. Also Bruntrup et al. uses GPS traces to generate maps and infer the road geometry. The advantage of GPS traces is that if the dataset is big enough, eventually a line derived from the GPS points will follow the exact curve of the road.

This research tries to detect quality errors and makes it possible for the OSM contributors to be alarmed for this errors and correct them. The attribute can have two levels in the hierarchical code list, L0 describes whether the road exceeds a certain geometry error and L1 provides the value of the error (see Figure 5.19). L0 will return "yes" when the geometrical error is bigger than 5 meters. L1 will provide the distance in meters of the offset of the road compared to the GPS data.

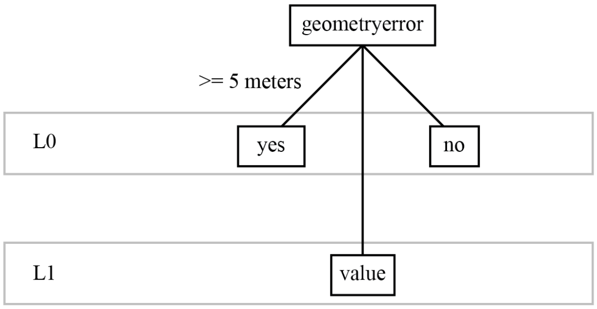


Figure 5.19: Different levels in the hierarchical code list for attribute "geometryerror"

Again, the relative distance can be used to calculate these errors. The mean of all the relative distances of a road is the average offset of the GPS data compared to the centerline of that road. Figure 5.20 shows an example of this for road "w.6973056". The dashed red line can be considered

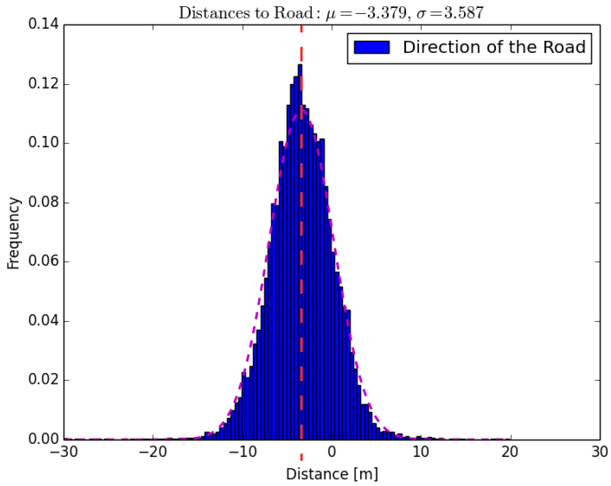


Figure 5.20: Histogram of road "w.6973056" showing the difference between the OSM centerline at 0 meters and the data's centerline at -3.379 meters (dashed red line)

as the data's centerline where the 0 meters value is the OSM centerline.

This mean, however, cannot be directly used as a quality indicator due to the fact that in the case of multiple lanes the distribution of the GPS points over these lanes influences the result. For example, if for a two lane road 90% of all the people on that road drive on the right side and only 10% of the people drive on the left side, the mean of the relative distances will not represent the centerline of that road. Unfortunately, it was not possible to derive the number of lanes out of this research that could be used for shifting the data to take into account the distribution of the GPS points on the lanes. Therefore, this research will only implement L0 with a relatively high threshold of 5 meters to detect large errors without taking into account the distribution of the GPS points. The final algorithm can be seen in Algorithm 7.

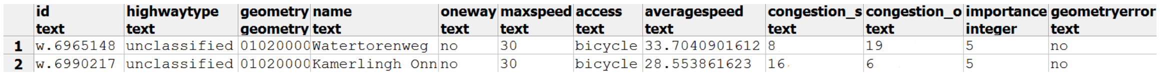
Algorithm 7 Attribute Extraction of "geometryerror"

```
function GEOMETRYERROR(roadid)
    select relative distance for roadid
    if $r \leq c5$ then                                ▷ r is the number of results
        $derive$ mean $from$ relative distance
        if mean<0 then
            $geometryerror = -1mean$
        else
            $geometryerror = mean$
        if $geometryerror \geq 5$ then
            $update$ geometryerror = "yes"
        else
            $update$ geometryerror = "no"
```

5.9 Updating Attributes

Now that all algorithms are developed, the OSM data should be updated. It is not possible to update these attributes directly on the OSM server, however it is possible to update the database table. Therefore a copy of the "osm_highways" table is made and the columns for the new attributes are added. Each algorithm can then update the results into the table. An updating function is created and implemented in the Python attribute extraction script using the Psycopg2 library. One single SQL command is send to the database using the column name of the attribute, the value of it to and the road ID to update the attribute for that particular road.

This is done for every attribute and for every road. The result can be seen in Table 5.21 which shows a sample of the updated OSM table.



	id	highwaytype	geometry	name	oneway	maxspeed	access	averagespeed	congestion_s	congestion_o	importance	geometryerror
	text	text	geometry	text	text	text	text	text	text	text	integer	text
1	w.6965148	unclassified	01020000	Watertorenweg	no	30	bicycle	33.7040901612	8	19	5	no
2	w.6990217	unclassified	01020000	Kamerlingh Onn	no	30	bicycle	28.553861623	16	6	5	no

Figure 5.21: Sample from the updated OpenStreetMap database table

The current situation for updating the attributes is depicted on the left in Figure 5.22. First the data is extracted from the OSM database and together with the GPS data stored on the local database. While the data is on the local database the preprocessing, the attribute extraction and the updating is performed on the local database. After that, the OSM database can be updated. Now, this has to be done manually. This means that for every road the attribute has to be edited manually in OSM. Currently, there is no awareness of a possibility to transfer the local database to the OSM database automatically.

The ideal situation would be storing only the GPS data in a local database and preprocess this data with the OSM data directly from the OSM database. No extraction of OSM data is then needed. Also, the attribute extraction and updating can then be performed on the data in the local database and the OSM database can be directly updated. The schema for this situation is depicted on the right of Figure 5.22.

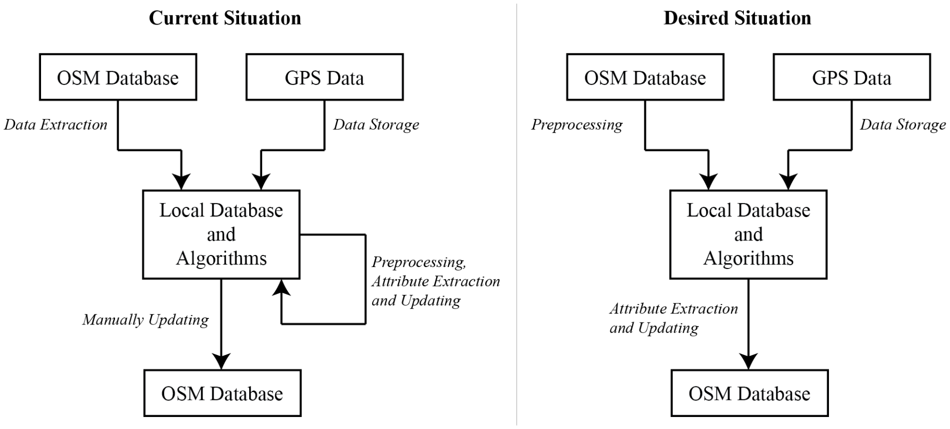


Figure 5.22: Current schema for extracting and updating attributes (l) and the desired schema (r)

CHAPTER 6

Validation

As already mentioned in previous Chapters and Sections, there are many factors that influence the results of the attribute extraction algorithms. This Chapter will discuss these influences, the minimum acceptable number of points and the validation and analysis of the attributes. A validation is needed to check the results of the final algorithms with ground truth. It is not possible for every attribute to automatically check whether the result is correct. Therefore the method of validation differs per attribute. This Chapter will explain the method of validation for every attribute and the results will be provided.

Validation sounds simple. However, there are many errors that should be thought of. Firstly, it is difficult to know what is ground truth. The available data for validation is also error sensitive, e.g. out of date satellite imagery or Google Street View. This is also related to the second error, the temporal aspect of the data. GPS data is collected at a certain point in time and extracts information about that moment in time. Therefore it is possible that data is derived during temporal changes in the road networks or other temporary conditions (e.g. roadwork). This can cause that the direction of a road is in a certain way till one point in time and from that point on it is changed in the opposite direction. These influences are already explained in Section 6.1. The third error consists of noise and inaccuracies, both from the GPS data and OSM data. All these influences can lead to Type I and Type II errors. A Type I error is also called a false positive, meaning that a classification has been made while in fact that classification is not correct or does not exist. A Type II error is called a false negative and means that a classification has been made and validated as correct while in fact the ground truth is wrong, as is the classification.

6.1 Influences on Attribute Extraction

This Section will discuss some external factors that will influence the attribute extraction algorithm and its results. There is for example the inaccuracy of the GPS data and the behavior of the people that can influence the results. But there are more factors which have not been discussed yet.

One of the major factors is time. There are many temporal aspects that influence the algorithms. One can think of permanent temporal aspects, but also temporary aspects.

Permanent temporal aspects are changes over time. The world keeps changing and so does the infrastructure. The GPS data used for this research is acquired over a few weeks time which is not long. However, when the algorithms will be used for multiple datasets the timespan in which the datasets are gathered is most likely larger. It can then occur that there are permanent changes made in the infrastructure, e.g. a change of speed limit or a change of driving direction for a one way road. This research does not investigate the possibility of detecting these changes over time, but this could be implemented in further research.

Besides permanent changes, there are also temporary changes like redirections during roadworks and speed limit changes during roadworks. It is difficult to detect these changes since the timespan of roadworks, for example, differs. Also the choice has to be made whether the temporary change should be updated or whether the original values should be kept.

Not only the time has influence on the result of the algorithms, but also the differences in location. This counts for example for differences between countries or differences between urban and rural. Not all the countries have the same infrastructure and laws for traffic. This research takes the speed limits of the Netherlands into account, but these speed limits could differ in other countries. Besides speed limits, there is also the width of the road. The width of the roads are not in all countries the same, there are even differences inside a single country. This makes it harder to detect the number of lanes for example.

While evaluating some attributes the former mentioned drawing problems in Subsection 3.1.2 came to notice. For example, a case with the "oneway" attribute that is depicted in Figure 6.1. In this case the algorithm classified the road as a two way road, while OSM considered it as a one way road. In real life, the road is indeed a one way road. However, the contributor that drew the road decided to draw the two roads that come together as one linestring instead of two separate lines. This causes the algorithm to detect GPS points in two directions for the road, assigning the "oneway" tag a "no". It is difficult to automatically account for these kind of drawing problems.

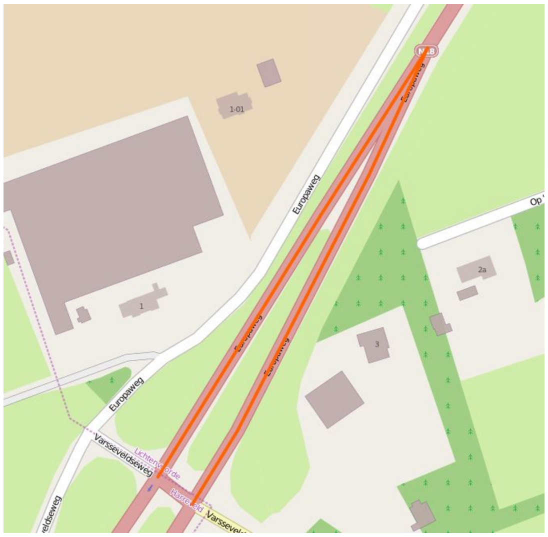


Figure 6.1: OpenStreetMap drawing problem for the "oneway" attribute

6.2 Minimum Acceptable Number of Points

The increase of the size of the data increases the accuracy of the results. This Section tries to determine this increase of quality and determines the minimal size of the data. To improve the reliability of the attribute extraction algorithms it is important to state a minimum acceptable number of GPS points on a road. This can be done in general for all attributes, but since the attributes have different conditions it is better to specify these minimum acceptable number of points for every attribute separately. However, a road should at least have 10 GPS points assigned to it to start the attribute extraction to remove roads containing outliers. The minimum acceptable number of points is acquired by calculating the error for a defined number of samples which are randomly selected out of the data. The smallest sample with an acceptable error is the minimum acceptable number of points. The errors are calculated using the training data and the samples that are used are:

- 10
- 20
- 50
- 100
- 200
- 500
- 1000

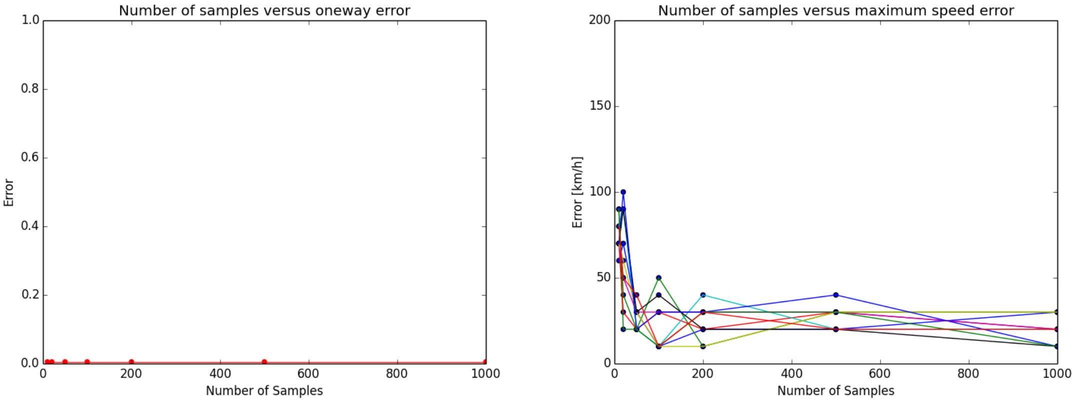
It should be noted that the errors can only be calculated if the actual situation of the training data is known.

This method was first applied to the "oneway" attribute. The result is depicted in Figure 6.2a. It can be seen that for the training set of 10 roads, no errors were made during the classification for all the samples. Thus it can be stated that it is reliable to retain the minimum number of 10 GPS points per road.

The "maxspeed" attribute is a more complicated attribute and this can also be seen from the results plotted in Figure 6.2b. For this attribute, the script ran 10 times to make sure the results are reliable. For 10 and 20 samples the error is still high, but at 50 samples the error seems to remain relatively stable. The average error at 50 samples is 26 km/h. This differs not much from the average speeds of higher samples that have average errors of respectively 23 km/h, 24 km/h, 27 km/h and 20 km/h. Therefore, for this attribute a minimum of 50 GPS points per road is chosen.

The minimum number of points for the other attributes cannot be calculated in a similar way as the previous attributes, mostly due to the lack of ground truth. Therefore a considered decision has to be made for these attributes. For the "access" attribute it is decided to have a minimum of 10 GPS points made by bicycle on a certain road. It can be stated that the minimum of 10 assigned bicycle points in the same direction and close to the road is reliable. The "averagespeed" attribute does not make a classification, but relies on the facts of the GPS points (i.e. the speed). Therefore it will have the smallest minimum of 10 GPS points to acquire a reliable average speed. The "congestion" attribute is based on the average speed and therefore also requires 10 GPS points per hour to determine a reliable average speed for every hour. Finally, the "geometryerror" attribute also uses the minimum amount of 10 GPS points per road.

The minimum acceptable number of GPS points for all attributes can be found in Table 6.1.



(a)

(b)

Figure 6.2: Graphs showing the error per sample for "oneway" (a) and "maxspeed" (b)

It should be noted that that the "importance" attribute is not taken into account, because the number of GPS points are not used in this case. The number of passes are used as an indicator of the attribute. Therefore the exact number of passes is important. Thus this attribute does not need a minimum number of points or passes.

Attribute	**Minimum number of GPS points**
"oneway"	10
"maxspeed"	50
"access"	10
"averagespeed"	10
"congestion"	10 (per hour)
"importance"	10
"geometryerror"	10

Table 6.1: Minimum acceptable number of GPS points per road

In total, the database contains 22.619 roads with 10 or more GPS points assigned to them against 38.639 roads that have less than 10 GPS points. There are 7.241 roads that have 50 points or more assigned to them. Only the points are taken into account that have a speed higher than 5, a distance to the road smaller than 30 and a similar or opposite heading as the road. The VCR is not taken into account.

6.3 One or Two Way Road

The "oneway" attribute consists of two levels in the hierarchical code list. L0 only returns "yes" and "no" to determine whether the road is one or two way. L1 also takes into account the direction of the one way road, whether the driving direction is in the same or in opposite direction of the drawing direction.

The first level in the hierarchy, L0, updated the attribute for 22.619 roads of which 11.577 roads where not informed yet. Thus the resulting 11.042 roads can be compared to the original OSM data. It can be stated that using the OSM data as ground truth for 11.042 roads is representable for the entire attribute.

The comparison showed that 72 out of 11.042 roads are differently classified, this is an error of 0,6%. For this level, "-1" falls under "yes". Out of the 72 errors, 23 errors are caused by not being able to classify the "reversible" roads. Also, some errors are caused by drawing errors that are similar to the one in Figure 6.1.

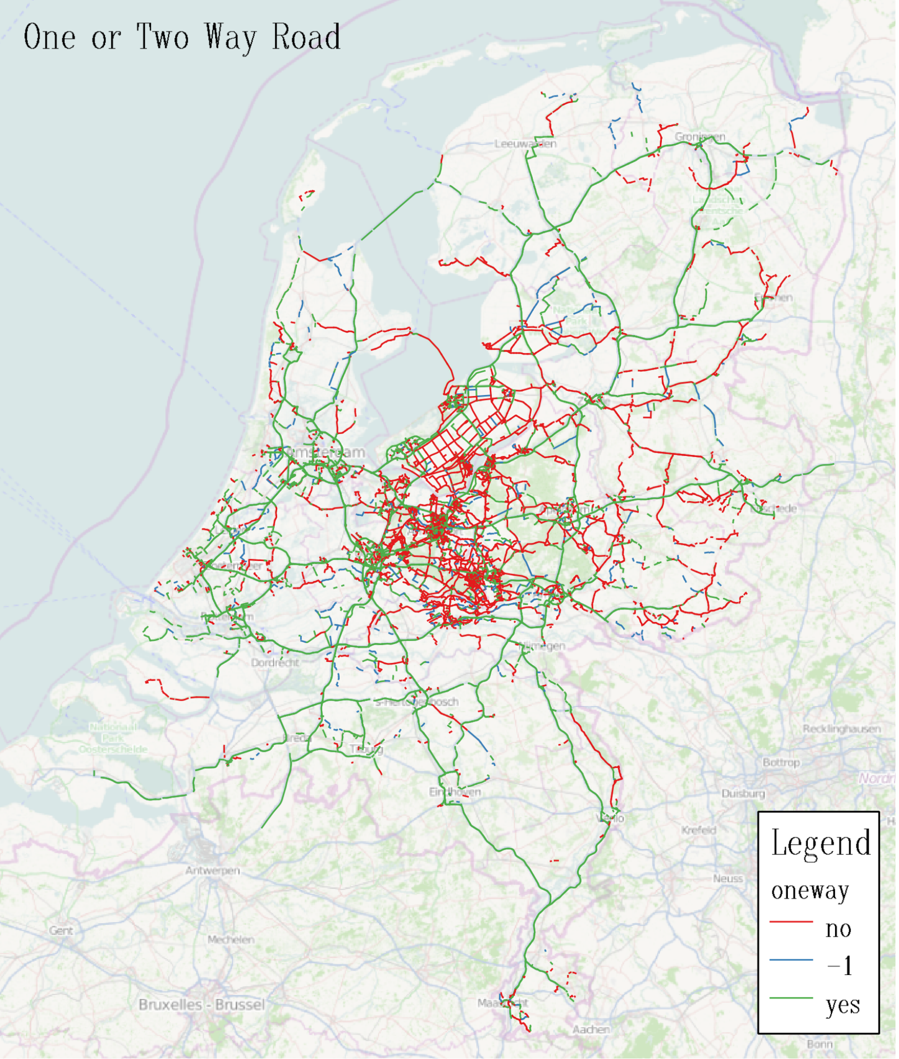


Figure 6.3: Visualization of the "oneway" attribute

L1 is only slightly different by also classifying the "-1" roads. The algorithm updated the same amount of roads with exactly the same error as L0. This means that all "-1" roads are correctly classified by the algorithm resulting in a correct classification of 99,4%.

The resulting classification is shown in Figure 6.3. This Figure shows the OSM map of the Netherlands with the classified roads.

For the one way attribute for bicycles, there is no ground truth available. However, since the same method is used for the cars it can be stated that the attribute for bicycles more or less have the same result and same classification error. It is also important that the classification of roads that allow bicycles is performed well.

6.4 Speed Limit

The "maxspeed" attribute consists of multiple levels in the hierarchical code list. L0 groups the speed limits into low, medium and high. L1 classifies the speed limits and L2 makes a distinction between speed limits during the day and during the night.

L0 updated a total of 5.137 roads of which 1.151 roads where not informed yet. This means that 3.986 roads are already informed in OSM and can therefore be used as ground truth for the validation of this attribute. The results show that 414 rows are differently classified for this level which results in an error of 10,4%, i.e. a correct classification for 89,6%.

L1 updated the same amount of roads as L0, but with a classification error of 34,0%. Further investigation showed that a part of the error is caused by the "links" of some types of highway, e.g. an exit of a motorway is called a "motorway_link". Drivers usually have to slow down for these links, causing the algorithm to differently classify these type of roads. Without taking into account these links the classification error decreases to 30,8%. This is still high, however many classifications are only one speed limit away from the actual speed limit, e.g. a classification of 30 km/h instead of 50 km/h or a classification of 100 km/h instead of 120 km/h. When taking into account that classification that are one speed limit away are allowed, the classification error decreases to 5%. Most of the errors are caused by higher speed limits, from 1.946 roads that are differently classified 1.192 roads are from roads with speed limits higher than 100 km/h. This is a percentage of 61,3% out of all errors. It is highly likely that these errors are caused by the driving behavior of people on motorways. GPS points on motorways have a different behavior than other types of roads. Where most people on 60 km/h roads drive around 60 km/h, people on a 120 km/h road tend to drive different from each other. Some people might drive 100 km/h, some people drive 140 km/h. People in trucks or people with caravans or trailers are mostly allowed 80 km/h. All these different behaviors cause classification of higher speed limits to be more inaccurate.

The results of L1 are depicted in Figure 6.4. A clear distinction between the high speed motorways and the lower speed city centers can be made in the Figure.

L2 is not yet implemented and therefore no validation over this can be made.

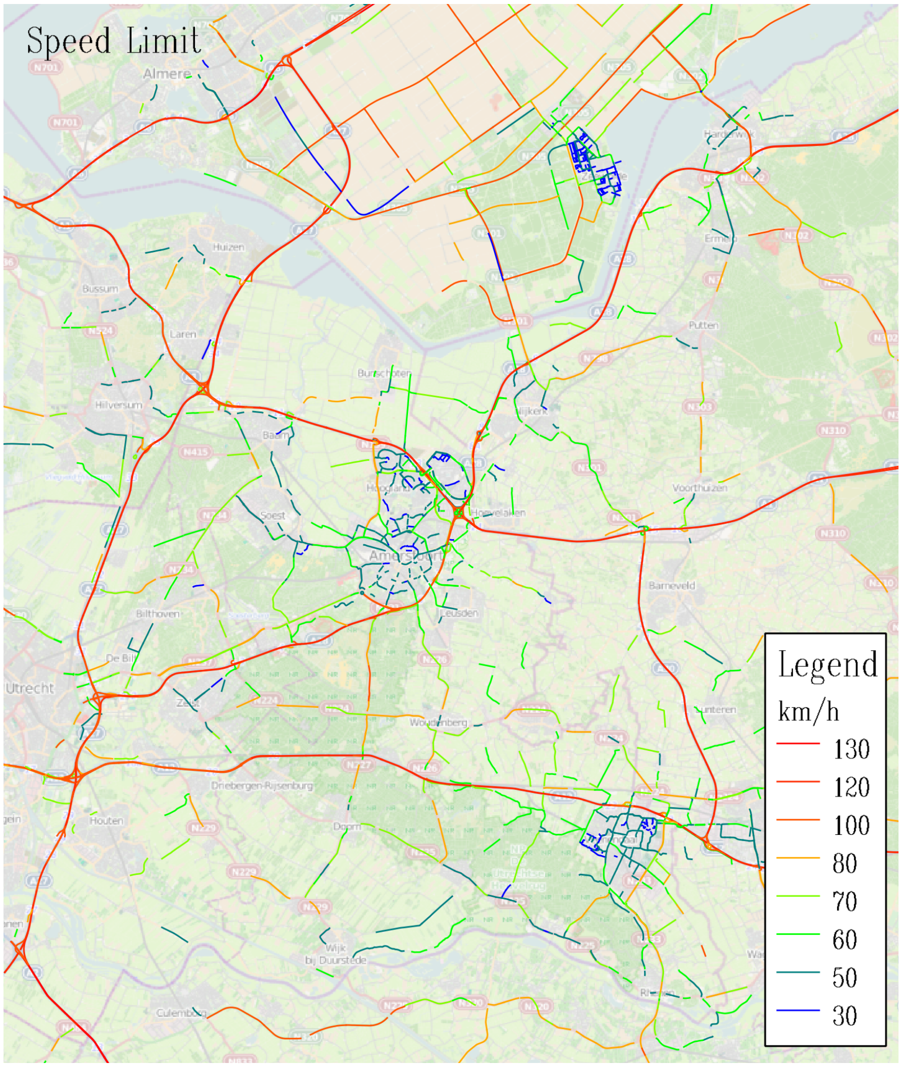
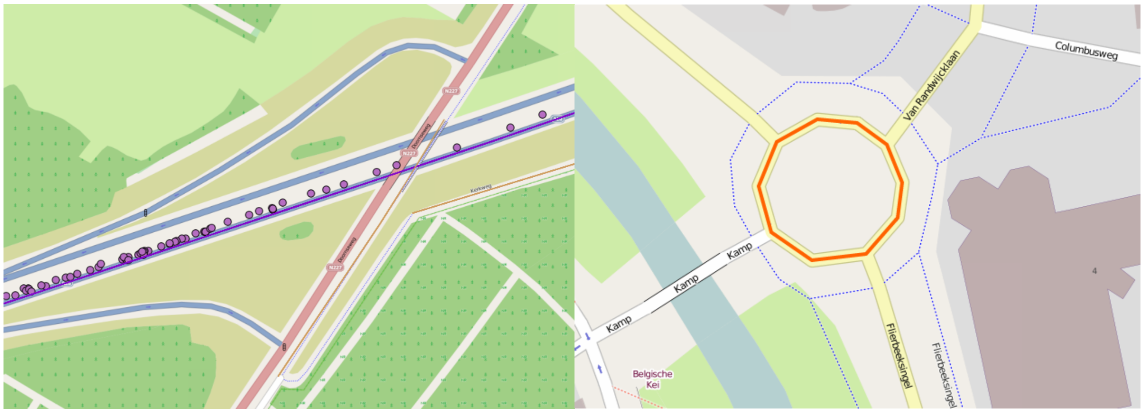


Figure 6.4: Visualization of the "maxspeed" attribute

6.5 Access of Different Types of Vehicles

The "access" attribute is only implemented for bicycles in this research. A total of 3.481 roads were given the "bicycle" tag by the algorithm. There is no ground truth available from the OSM data, therefore a manual validation was performed. In total, a random sample of 100 roads were validated using Google Maps and Google Street View.

The validation resulted in 26 wrongfully classified roads which is equal to an error of 26%. These errors had different causes. The "bicycle" tag was given 84 times to a motorway or a trunk. Investigation of these points showed that the GPS points classified as bicycle are on the motorway and therefore it is highly likely that the classification and segmentation algorithm in the preprocessing phase returned the wrong transportation mode. An example of this is shown in Figure 6.5a.



(a) (b)

Figure 6.5: Problems with the access attribute: wrongfully classified bicycle points (a) and the problem with roundabouts (b)

A different problem involves roundabouts. Most roundabouts have a cycleway around the roundabout, this can be seen in Figure 6.5b where the blue dotted line is the cycleway. However, the algorithm takes into account the heading of the GPS point and the direction of the road. Apparently, these do not match for the cycleway but do match for the road causing the algorithm to assign the bicycles to the road instead of the cycleway. During the validation of 100 roads, there were 3 of these cases.

The most common problem is the error caused by map matching. There are multiple roads which are classified that they allow bicycles, where the GPS points actually belong to the road next to the assigned road. Therefore the algorithm thinks that the bicycles are on the road, while actually they are next to the road on a different road. These map matching errors cause a wrong classification of the access attribute.

Finally, there are also still some normal classification errors of which the cause is not directly clear. During the validation it came to notice that roads are regularly modified, thus it might be that this is also one of the problems. It is then possible that the cycleways next to the roads are not yet drawn in the data that is used for this research, but they are drawn before the validation is performed.

The algorithm also showed that is capable of detecting new paths for bicycles. There were an amount of roads that were not classified as a cycleway, but turned out to be cycleways. In combination with car points new cycleways could be detected with this method, however this is something for future work. Also roads which are now correctly given the tag "bicycle" can be used for bicycle navigation software.

6.6 Average Speed

The average speed is updated for two levels in the hierarchical code list. First there is the normal average speed, L0, which counts for the entire road, i.e. in both directions. Second there is the average speed per direction which is L1. L2, the average speed per hour is not implemented in this research but is used for the congestion attribute.

The results of the average speed cannot really be validated, since the average speed is directly derived out of the data. However, an indication of the results will be given in this Section.

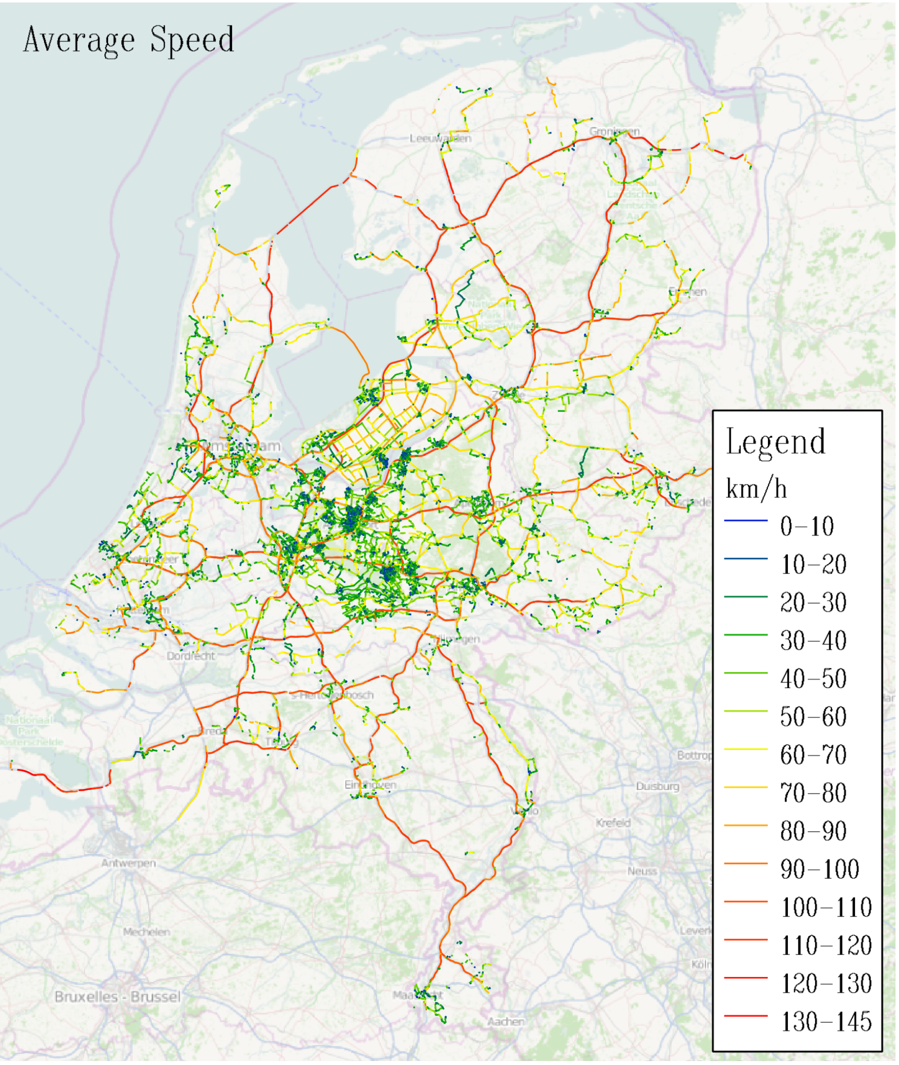


Figure 6.6: Visualization of the "averagespeed" attribute

The L0 algorithm updated the average speed for a total of 25.325 roads. The average of all these roads is 54,65 km/h with a minimum of 5,15 km/h and a maximum of 145,32 km/h. L1 affected a total of 19.132 roads, the difference between L0 and L1 results out of the fact that a minimum of 10 GPS points is needed per direction. Some two way roads have only 10 GPS points making it not possible to derive the average speed in both directions. The average of all the roads in similar direction is 60,26 km/h and in opposite direction is 43,71 km/h. The average in similar direction is higher due to the fact that all motorways are one way roads in the similar direction as the drawing direction. The minimum in similar direction is 6,36 km/h and in opposite direction 6,45 km/h. The maximum in similar direction is 139,31 km/h and in the opposite direction 108,80 km/h. There are 1.042 roads where there is a difference of 5 km/h between one of the directions and the average speed of both directions. And there are 184 roads where this difference is higher than 10 km/h. This difference between directions can be caused by congestions that happen more often or more heavily in one direction than the other.

The map that is made for the L0 average speed in the Netherlands can be seen in Figure 6.6, a zoomed map on the three cities can be seen in Figure6.7. Seeing the high average speeds of the motorway network next to the lower average speeds in city centers may be a good indicator of the accuracy of this attribute.

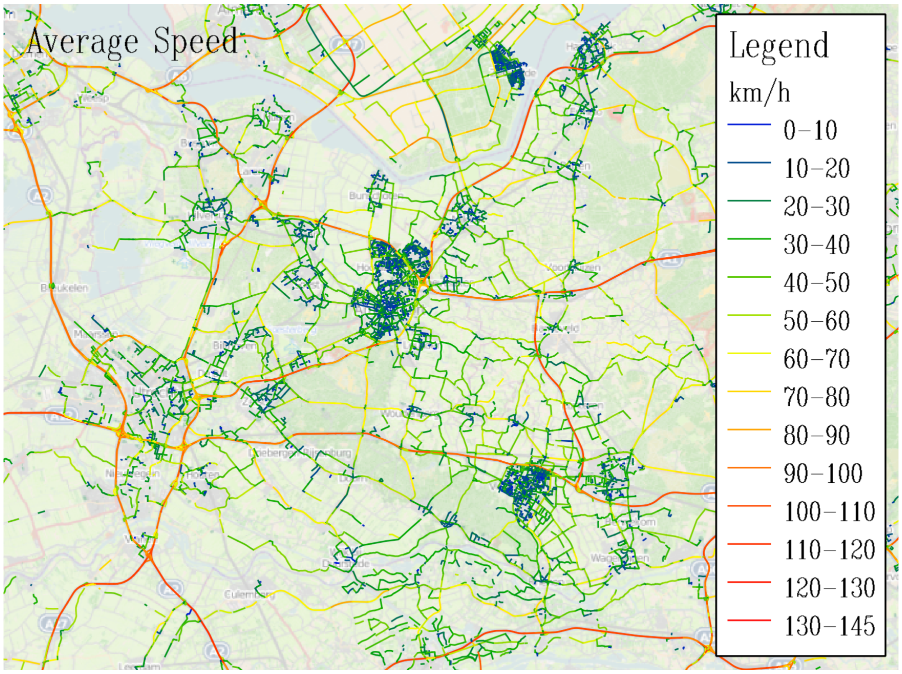
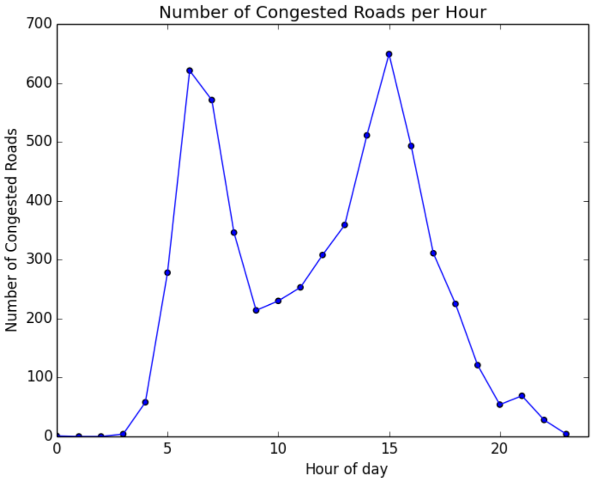


Figure 6.7: Visualization of the "averagespeed" attribute zoomed on the three cities



Hour	Roads		Hour	Roads
0	1		12	308
1	0		13	359
2	0		14	511
3	4		15	649
4	58		16	493
5	278		17	311
6	621		18	225
7	571		19	121
8	346		20	54
9	214		21	69
10	230		22	28
11	253		23	4

Figure 6.8: Number of congested roads per hour

6.7 Hours in which Congestion Occurs

The "congestion" attribute is only implemented for L0 and only for the weekdays. This is done as a result of time issues and because this level is most important and interesting due to work traffic. The validation of this attribute is difficult since there is no ground truth available. Again, an indication of the results will be given in this Section.

The algorithm updated 3.729 roads of which 390 roads where at least one hour congested in both ways. When these roads are grouped by type of highway, a remarkable discovery is done. The most congested roads are motorways with a total of 1.245 roads, then primary roads (618 roads), secondary roads (580 roads) and finally tertiary roads with a total of 417 roads. This shows that the amount of congestion follows the structure of the type of roads.

When looking at the distribution of the congestion per hour, the expected hours are the most congested. Figure 6.8 shows the resulting graph and its table. The most congested hours are 6 (i.e. from 06:00 hours to 07:00 hours) and 15 (i.e. from 15:00 hours to 16:00 hours). This can be related to the community, where most people have working hours from early in the morning till the afternoon. This can also be seen in the Figure where the peaks of the graph center around these parts of the day.

6.8 Importance of a Road

The "importance" attribute is also difficult to validate, therefore an indication of the results will be given again. A total of 67.968 roads were updated with a category of importance between 1 and 5, with 1 the most used roads and 5 the least used roads. Logically, category 5 is the category with the most roads for it contains also the roads that are not used. The number of roads in category 5 is 67.456. Category 4 has 415 roads, category 3 66 roads, category 2 15 roads and category 1 16 roads.

The roads colored by importance can be seen in Figure 6.9. The most used roads are situated near Zeewolde and are probably the most important roads to and from Zeewolde. These are mostly secondary roads, 10 roads out of 16 to be precise. Even the roads in category 2 are mostly secondary, 13 out of 15 roads are secondary roads. Category 3 and 4 are mostly motorways.

The results shows what is already expected in Section 5.7. The GPS data is biased because it is derived from people out of only three cities. Therefore Zeewolde now has important roads, where normally this is probably not the case. However, when data covering and representing the Netherlands could be derived this algorithm could be a solution to classify the roads by importance.

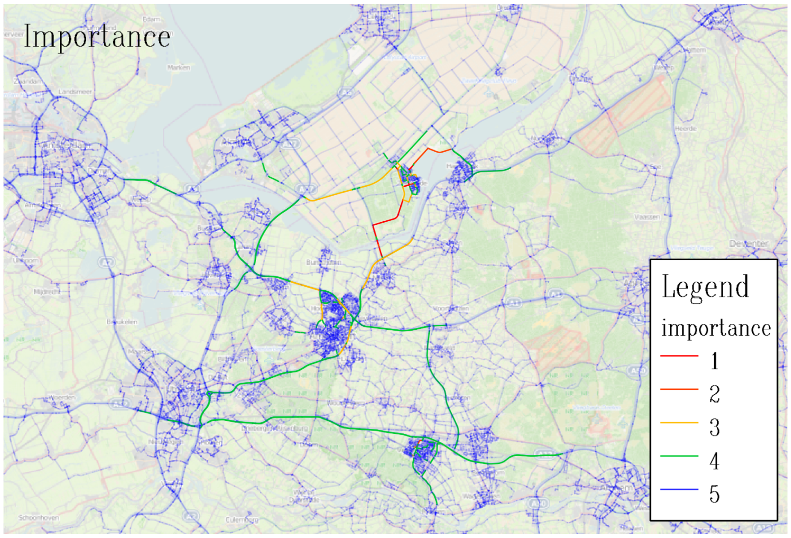


Figure 6.9: Visualization of the "importance" attribute

Therefore the algorithm is also applied on the three cities separately, only using road passes that are made by people from one city. For example, the road passes made by users from Amersfoort are used to define the importance in Amersfoort. This is done for all three cities separately and the results can be seen in Figure 6.10. The legend of the results is similar to Figure 6.9.

What should be noticed is that in most cases the most important roads are the roads that are used for entering and exiting the city. The people enter and exit the city at a few roads around the city and then split up in different directions. A table is created to compare the results of the total importance with the results per city, this can be seen in Table 6.2. Zeewolde has the most updated roads and the most amount of important roads, where Veenendaal has the least updated roads and no classification of the second category.

Especially the results in the Figures show that this attribute has its limitations. The more realistic the coverage of the data is compared to the real world, the better the results represent the world.

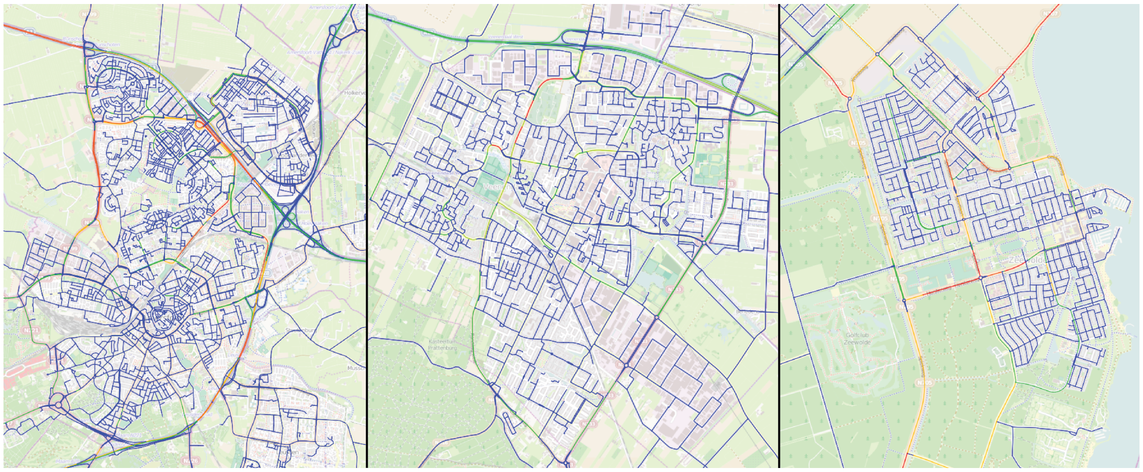


Figure 6.10: Visualization of the "importance" attribute for the three cities: Amersfoort, Veenendaal and Zeewolde

Results of	**Roads Updated**	**Max # of Road Passes**	**Cat. 5**	**Cat. 4**	**Cat. 3**	**Cat. 2**	**Cat. 1**
Amersfoort	32.271	291	31.929	256	57	28	1
Veenendaal	25.842	419	25.842	120	17	0	6
Zeewolde	37.305	520	37.150	99	29	7	20
Total	67.968	537	67.571	322	49	13	13

Table 6.2: Results of the total importance of the Netherlands compared with the results per city

6.9 Error of the Road Geometry

The "geometryerror" attribute returns "yes" if the algorithm suspects an error in the geometry of that road. An error in this case is if the GPS data differs more than 5 meters with the centerline of the OSM road. This Section compares the geometry error attribute with Google Satellite data.

The roads with a geometry error are plotted on Google Satellite data to investigate the errors of these roads. In total, there are 2.597 roads classified with a geometrical error. All these roads are depicted in red in Figure 6.11. There are many roads which are just slightly shifted from the actual centerline of the road. However, there is also an amount of roads that have different kind of errors.

At first there are the roads which have a too long extension. These are usually roads that connect to other roads, e.g. motorway links or in other words the entrances and exits of the motorways. These roads tend to attach to the motorway to soon, leaving a part of the entrance of the motorway not covered where the GPS points continue. Therefore the distance to the centerline increases and the geometrical error is noticed. Secondly, there are also roads that are completely wrong. These roads have a complete offset with the actual road. It is difficult to say how these errors originate. It is possible that permanent or temporary changes are causing these errors. Thirdly, there is the problem with curves in the roads. This is a common problem which causes slight errors. These

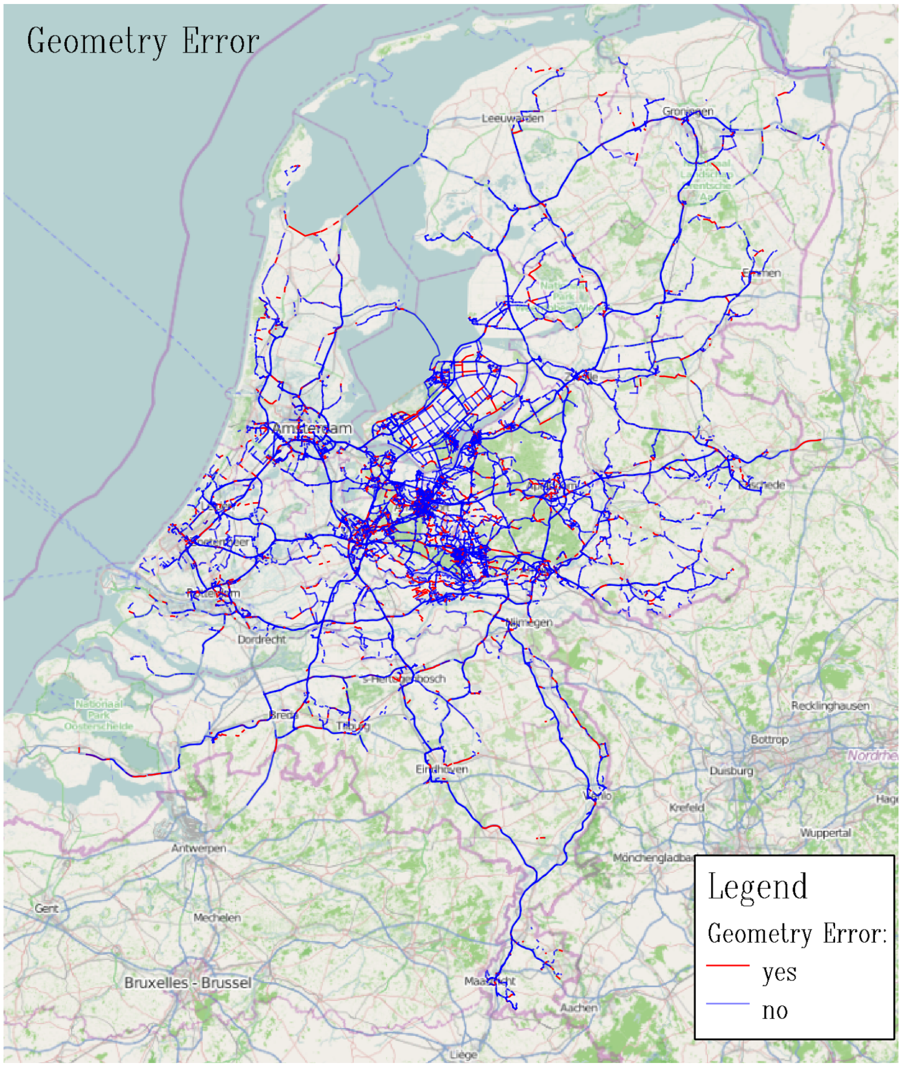
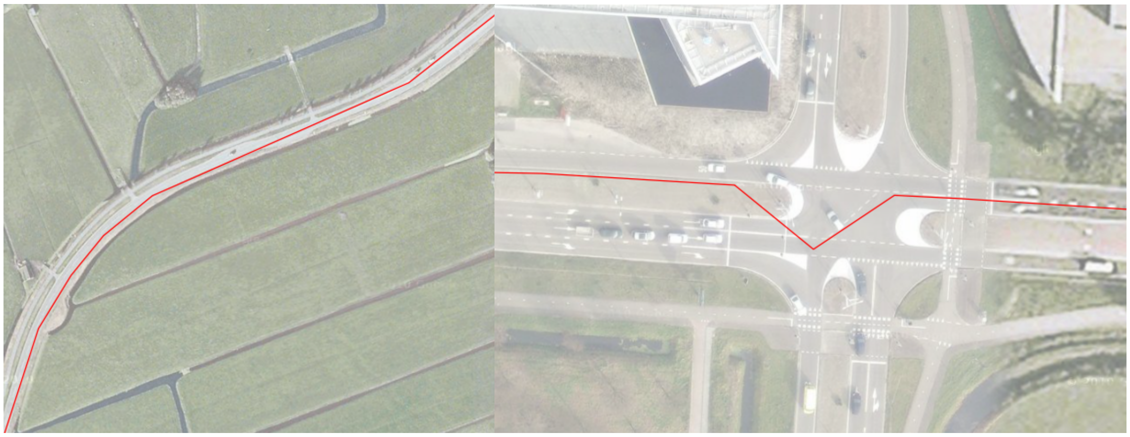


Figure 6.11: Visualization of the "geometryerror" attribute

errors originate from the way of drawing of the contributor. Curves are drawn using points, the more points that are in a curve the better the shape is. However, less points means a more generalized road and therefore can cause errors. An example of this case can be seen in Figure 6.12a. Finally, there are also other problems with the way of drawing of the OSM contributors. Some roads have strange shapes and do not follow the actual shape of the road. This can be seen in Figure 6.12b.



(a) (b)

Figure 6.12: Geometry errors from the data visualized on Google Satellite: curved roads (a) and strange drawing (b)

6.10 Overview and Analysis

This Section provides an overview on the classification of the different attributes, an analysis of the errors and the possible sources of errors. The validation that is performed focuses on the total error of the classification. However, it is not clear whether all of the errors are caused by the algorithms that are developed for this thesis. There are numerous external factors which can also cause errors, e.g. wrong classification of transportation modes or errors caused by the TrackMatching algorithm.

An overview on the current classification accuracies is given in Table 6.3. This Table only takes into account the attributes of which a classification error is available. The error for determining whether a street is a one or two way street is 0,6% for both levels in the hierarchical code list. The error for the speed limit is 10,4% for L0 and 34,0% for L1. The access attribute has a classification error of 26,0%.

Attribute	**Level**	**Classification Accuracy [%]**
"oneway"	0	99,4
	1	99,4
"maxspeed"	0	89,6
	1	66,0
"access"	0	74,0

Table 6.3: Overview of the classification accuracy of the attributes

The different attributes have a different minimum number of required points, but they also have different input values. These values are not always informed for every point, causing the attributes to update a different amount of roads. Table 6.4 is created to put these numbers into perspective.

The Table shows the percentage of the attributes that are informed compared to the total number of motorised roads before and after the attribute extraction algorithm is applied. The percentage of informed attributes before the algorithm only applies to the first three attributes, since these are provided in the data of OSM. Column two and three shows the percentage compared to the total number of roads in the Netherlands, where column four and five only take the roads into account that contain GPS points. The "oneway" tag achieved a total increase of 1,3% and a relative increase of 17,0%, for the speed limit these percentages are respectively 0,1% and 1,7%. These percentages are lower due to the higher minimum number of required points. The access attribute is provided by OSM, however none of the updated roads have this attribute informed. The importance attribute has the highest amount of updated roads, because the roads which have 1 GPS point on them are also classified. In other words, almost all roads are updated in the case of this attribute. L1 of the average speed attribute is updated less than L0, because not all roads with a total minimum number of 10 points have the same minimum number in one single direction (i.e. similar or opposite). The congestion and access attribute are only updated when the road is either congested or allows the access of bicycles, explaining there low percentages.

Attribute	**Tot. Informed Roads Before Extraction [%]**	**Tot. Informed Roads After Extraction [%]**	**Rel. Informed Roads Before Extraction [%]**	**Rel. Informed Roads After Extraction [%]**
"oneway"	13,8	15,1	43,1	60,1
"maxspeed"	26,3	26,4	50,8	52,5
"access"	n/a	0,3	n/a	4,0
"averagespeed" L0	n/a	2,8	n/a	37,3
"averagespeed" L1	n/a	2,1	n/a	28,1
"congestion"	n/a	0,4	n/a	5,5
"importance"	n/a	7,4	n/a	100
"geometryerror"	n/a	2,5	n/a	34,1

Table 6.4: Overview on the total (all roads) and relative (only roads with GPS points) percentages of informed attributes in OSM before and after extraction

The resulting errors can have different causes, which may be internal or external. Internal errors are the errors which are caused by the attribute extraction algorithm developed in this thesis. External errors are errors which cannot be avoided, i.e. caused by external factors. There are numerous external factors which can cause these errors:

- classification of the wrong transportation mode
- GPS points that are map matched wrongfully
- wrong classification due to one of the influences on attribute extraction (temporal aspect, location or way of drawing)
- data used as ground truth is not correct or outdated

The methods used in the preprocessing phase are also not 100% accurate. The classification and segmentation algorithm by Biljecki et al. (2013) has a classification accuracy of 91,6%. This means that 8,4% of the tracks is given the wrong transportation mode. This could influence the attribute extraction, since most attributes assume that the tracks are made by car or in the case of the access attribute by bicycle. The wrong classification of transportation mode causes the attribute extraction algorithm to extract wrong information. In the case of the one way attribute it is possible that other vehicles are allowed in two ways of a certain road, but cars are not allowed in two ways. For the maximum speed attribute it might be the case that the speeds that are derived are actually from different vehicles than a car. For some vehicles this might not be a problem, however if the algorithm assumes that it is a car while in fact it is a bicycle strange results can be updated. Also for the access attribute it is a problem when the assumption of bicycle points is made, while in fact it are car points.

Wrongfully map matched GPS point causes the GPS point to have the wrong road assigned. The TrackMatching method that has been used for map matching has a classification error of around 4,5%, which would mean that 4,5% of the points has a wrongfully assigned road. In this case wrong information is derived for that road. This is a problem for the extraction of all attributes. There is also the possibility of wrong classification due to the influences discussed in Section 6.1, these are temporal influences, location based influences and the influences caused by the way of drawing in OSM. It is also difficult to state what is ground truth. For the validation of the attributes OSM and Google Maps/Satellite are used. However, there is also the possibility that this "ground truth" is wrong. It can be the case that the ground truth is outdated or that the situation has changed since the acquisition of the GPS data and therefore the ground truth is more recent than the data.

The classification accuracies that are calculated in the previous Sections contain errors that are caused by these external factors. Therefore these classification errors not really represent the error of the algorithm. The errors caused by external factors will be filtered out to find the correct representation of the error of the attributes.

The "oneway" attribute has a classification error of 0,6% using automatic validation using OSM as ground truth. Manual validation has been performed on the errors using Google Maps and Quantum GIS to find the cause of the errors. All 72 wrongfully classified roads were manually validated and the results showed that three of the 72 roads were wrongfully classified by OSM and not by the extraction algorithm. Further it should be noted that 10 of the errors are caused by the fact that "reversible" is not taken into account in the algorithm. Also five errors were caused by drawing errors in OSM similar as the one in Figure 6.1. Finally, two errors were detected due to bad data. This data occurred in city centers where the GPS points were scattered, most likely due to the urban canyon problem. All these external factors caused the classification error to be higher than the actual error of the algorithm, without these external errors the classification error of the algorithm is 0,5%. To be precise, a decrease of 0,15% is achieved.

For the "access" attribute a similar research has been performed. However, the access attribute was already manually validated. From the 100 validated roads, 26 were classified incorrectly. From these 26 errors, 11 roads were wrongfully classified due to wrong map matching. An example of this map matching problem is depicted in Figure 6.13. The GPS points that are on the parallel road are matched to the road next to it (orange road), causing the algorithm to state that this road allows bicycles while in fact no bicycles are allowed. One of the reasons for this error could be that the map matching algorithm does not notice that the first GPS points are on the cycleway, simply because these are not taken into account. Therefore, most likely, the algorithm expects the points to be on the roundabout for example and therefore classifies the points to the wrong road.

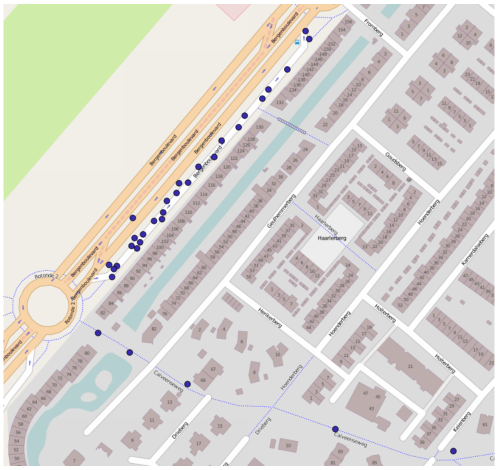


Figure 6.13: Map matching error: points are matched to orange road instead of the white road next to it

Also, six classification errors were made due to classification of the wrong transportation mode. Most likely these classification errors originated from low speed GPS points in traffic jams, resulting the classification algorithm to assign the wrong transportation mode. Finally, one error was also caused by drawing errors in OSM. Without these external factors the classification error of the algorithm decreased 16,24%, from 26% to 9,8%.

It is difficult to perform the same manual validation for the "maxspeed" attribute, since there is no ground truth data freely available. The only external problem that can be mentioned for this algorithm is the "motorway_link" problem. In total, 125 roads of a total of 1.354 wrongfully classified roads are caused by this problem. Without these links the classification error of the algorithm decreased 3,2%, from 34,0% to 30,8%.

These external factors cause errors which cannot be directly solved and which are not caused by the algorithm itself. Therefore a revised overview on the classification accuracy of the attributes is given in Table 6.5. Here are the percentages taken into account which are not caused by external factors, thus these percentages of error represent the classification error of the actual attribute extraction algorithm.

Attribute	**Level**	**Classification Accuracy [%]**
"oneway"	0	99,5
	1	99,5
"maxspeed"	0	89,6
	1	69,2
"access"	0	90,2

Table 6.5: Revised overview of the classification accuracy of the attributes, without the influence of external factors

CHAPTER 7

Conclusions and Future Work

This Chapter will elaborate on the conclusion made based on the research and will provide some ideas for future work.

7.1 Conclusions

Before starting with the conclusions it is important to remember the research questions of this thesis.
The main research questions was:

- To what extent is it possible to derive and update attribute road data by using movement trajectories?

The research sub questions were:

- Which road attributes in OpenStreetMap data can be enriched by GPS tracks?
- Can GPS tracks provide information for new attributes to OpenStreetMap data?
- What is needed to extract the information for the missing attributes?
- How can the method automatically update OpenStreetMap?
- How can the results be validated?

The main research question will be treated last in this Section, first the sub questions will be treated.

This research has proven that not all attributes can be updated. First a selection was made based on the hypothesis that movement trajectories could be used to enrich these attributes. These attributes are "oneway", "maxspeed", "lanes" and "access". These attributes were mostly not informed and sometimes wrongly informed. The percentages of informed roads are for "oneway" 13,9% and for "maxspeed" 26,3%. For the other two attributes these numbers are not known, since they were not available in the OSM dataset. Movement trajectories, or in the case of this research GPS tracks, have also proven to be able to extract information for new attributes. The new attributes that are introduced in this thesis are "averagespeed", "congestion", "importance" and "geometryerror".

Before the attribute extraction can be performed, a lot of preprocessing has to be done. The most important steps in this phase are the classification of the transportation mode (Biljecki et al., 2013) and the map matching of the GPS points (Marchal, 2013). The latter assigns the road ID on which the GPS points are located. This part is essential for deriving the attributes for the roads. Further preprocessing that needed to be done was calculating the distances between GPS points and the roads, determining the direction and difference in direction of the roads and GPS points etc. After this preprocessing phase, the algorithms for extracting the attributes could be developed.

The different attribute extraction algorithms make use of different input and different methods. The "oneway" attribute makes use of the heading of the GPS point, where "maxspeed", "averagespeed" and "congestion" uses the speed of the GPS points. The "lanes" and "geometryerror" attributes used the distribution of the GPS points. All these different inputs led to different methods and different results. This research has proven that it is very difficult or near to impossible to detect the number of lanes out of GPS tracks. Also, the "importance" error needs a decent coverage to acquire representable results.

The results that are acquired could not be directly updated into OSM. The existing and new attributes are added and/or updated in the local database. This database contains a copy of the actual OSM data. It can be stated that at the moment, there is a gap in updating OSM in an automated way. A possible reason for this can be the influence of the contributors in the OSM community which are against automated methods for updating OSM. However, the possibility of automatically updating OSM could be very beneficial for the completeness and the level of detail for OSM in the future. If people are then willing to provide movement trajectories for attribute extraction on a regular basis, e.g. from OSM-based mobile applications, a constant flow of data will then be available for attribute extraction.

Finally the results were validated. It was not possible to acquire a ground truth for all attributes. The "oneway" attribute resulted to be the most accurate with a correct classification of 99,4% and "access" classified 74% correctly. The "maxspeed" attribute resulted in a correct classification of 89,6% for L0 and 69,2% for L1. However, when taking into account speed limits that are one step away from the classified speed limit the classification increases to 95%.

This research not only updates the standard attributes in OSM, but also extends the OSM data with four new attributes (i.e. "averagespeed", "congestion", "importance" and "geometryerror"). When updated, the results can contribute to many different applications. First of all navigation systems can operate better for cars and for bicycles due to the "oneway", "averagespeed" and "access" attributes. Traffic analysts will have access to more information about the roads with the "maxspeed", "importance" and "congestion" attributes. Finally, OSM contributors can also benefit from the "geometryerror" attribute to increase the accuracy of the geometry in OSM.

It should also be noted that the strength of this research is that it is completely automatically. All steps that are performed in this thesis are implemented in Python and thus one single program can be created that automatically derives the attributes starting from the map matching till the updating. By deriving and updating the attributes automatically, the error of the attributes may also be reduced since the contributors are allowed to freely fill in the tags which is highly error sensitive. Also, this research is useful not only for OSM and GPS data but also for other maps (e.g. Google Maps, HERE Maps) and other movement trajectories (e.g. other GNSS, cellular trajectories). The latter could be acquired using crowdsourcing which could cover a great part of the earth and opens opportunities for attribute extraction all over the world and ideas for new applications and attributes.

7.2 Future Work

This research is one of the first in its kind which means that there is still much to do and improve. This Section will provide some recommendations for future work related to this thesis.

First improving the attribute extraction algorithms could be a possibility. The algorithms of these 8 attributes are developed in a strict timespan and could be improved for more reliable and better results. One of the things that could be improved is adding a confidence factor that represents the reliability of the results. Then, a better estimation can be made whether the attribute should be updated or not. Adding constraints might also improve the classification accuracy of the algorithms. For example, the "access" attribute could have the constraint: no bicycles allowed on motorways. It is certain motorways do not allow bicycles and therefore that possibility can be left out. This would already improve the current algorithm and could be applied for more attributes.

The "oneway" attribute could be improved by adding the "reversible" possibility to be more complete. The algorithm might improve further when also using a minimum of distinct user IDs and not only the number of GPS points. In the current case it is possible that on long roads, all the points on that road are made by only one user ID in one trip. This could cause the algorithm to classify this road as a one way road, however more user IDs would increase the reliability and certainty of the classification.

For the "maxspeed" attribute it could be beneficial to do a more in-depth research on classifying using pattern recognition techniques. Now, only the basic techniques are used due to the timespan. A more complex method could improve the attribute extraction algorithm in the future.

Attributes could also be expanded, like the "access" attribute. More types of vehicles than only bicycle could be integrated in this attribute. Cycleways could also be detected by using the roads which allow bicycles, but do not contain GPS points from other types of vehicles. The same can be done with pedestrians. For this it is important that map matching to cycleways and footpaths is possible, making the results more reliable and accurate.

The geometry error could be expanded in a way that not only a warning is given, but the actual geometry is corrected using this attribute. Zhang et al. already implemented this in their paper and this could also be implemented in the case of this thesis.

Also, a complete study could be performed on the derivation of the number of lanes of a road. Both Zhang et al. and Chen and Krumm have already performed research on this topic, however the results could be improved. Improvement of this attribute, and possible other attributes, could be derived from different sources of data. For this research only movement trajectories were taken into account. If more data was available and allowed the extraction of certain attributes might be improved. Examples of possible extra data are cameras on cars that take pictures every few seconds, satellite imagery and other datasets. Satellite imagery and the photos from the cameras might improve the extraction of lanes and perhaps open new opportunities for new attributes.

Ideas for new attributes can also be reason for future work. There are still many attributes of which one can think of, e.g. if it is allowed to pass a vehicle on a two way road or the average waiting time on a road with a lot of traffic lights. And there are more attributes that can be derived and implemented using movement trajectories. When a new attribute has to be extracted, one can follow a few steps to develop the algorithm. These steps are depicted in Figure 7.1. The first step is logically the idea for a new attribute, the second step is then to define which value is needed for deriving the attribute. This can be the speed of the GPS, the direction, etc. After that, one should think of the outliers that should be removed, e.g. for the "maxspeed" attribute it is important to remove the points that exceed the threshold of the VCR. However, for the "averagespeed" attribute it is very important that these points are still present for a good estimation of the average speed. The fourth step is to choose a technique to extract the attribute, this can be a developed rule or a pattern recognition technique for example. Finally, all these steps and choices should come together in developing the algorithm and optimizing it in a way that the result is best.

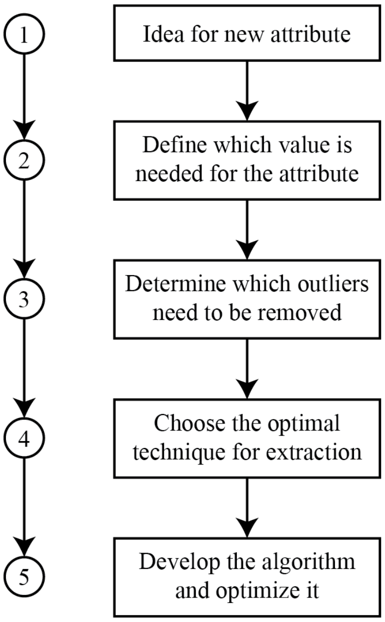

1
Idea for new attribute
2
Define which value is needed for the attribute
3
Determine which outliers need to be removed
4
Choose the optimal technique for extraction
5
Develop the algorithm and optimize it

Figure 7.1: Action plan for new attributes

One of the major ideas for future work is the possibility to derive the attributes real-time with data that is added every moment of the day. Crowdsourcing might be helpful in this case to acquire a vast amount of movement data from people. This could lead to live maps which would change almost real-time with the things that happen on the road. This research would be a few steps further than the research of this thesis, but could lead to new possibilities in maps.

References

Bernstein, D. and Kornhauser, A. An introduction to map matching for personal navigation assistants. 1998.

Bhattacharya, P. Quality assesment and object matching of OpenStreetMap in combination with the Dutch topographic map TOP10NL. *MSc thesis in Geomatics*, 2012.

Biljecki, F., Ledoux, H., and van Oosterom, P. Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science*, 27(2):385–407, 2013.

Brandeis, L. and Dossick, P. *The right to privacy.* Editions Artisan Devereaux, 1890.

Bruntrup, R., Edelkamp, S., Jabbar, S., and Scholz, B. Incremental map generation with GPS traces. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE, pages=574–579, year=2005, organization=IEEE.*

Burges, C. J. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

Chen, Y. and Krumm, J. Probabilistic modeling of traffic lanes from GPS traces. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages=81–88, year=2010, organization=ACM.*

Ciepłuch, B., Jacob, R., Mooney, P., and Winstanley, A. Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps. In *Proceedings of the Ninth International Symposium on Spatial Accuracy Assessment in Natural Resuorces and Enviromental Sciences 20-23rd July 2010, pages=337, year=2010, organization=University of Leicester.*

Clementini, E., Sharma, J., and Egenhofer, M. J. Modelling topological spatial relations: Strategies for query processing. *Computers & graphics*, 18(6):815–822, 1994.

Crew, H. *The Principles of Mechanics.* BiblioBazaar, LLC, 2008.

Fan, H., Zipf, A., Fu, Q., and Neis, P. Quality assessment for building footprints data on OpenStreetMap. *International Journal of Geographical Information Science*, (ahead-of-print):1–20, 2014.

Friedman, J. H. Data Mining and Statistics: What's the connection? *Computing Science and Statistics*, 29(1):3–9, 1998.

Giannotti, G., Giannotti, F., and Pedreschi, D. *Mobility, data mining and privacy: Geographic knowledge discovery.* Springer, 2008.

Girres, J.-F. and Touya, G. Quality assessment of the French OpenStreetMap dataset. *Transactions in GIS*, 14(4):435–459, 2010.

Google. Google Maps, month=may, year=2014, url=http://maps.google.nl.

Google. Google Map Maker, June 2014. URL `http://www.google.nl/mapmaker`.

Hahn, A. Wiki GISpunkt der Hochschule Fur Technik Rapperswil, May 2014. URL `http://giswiki.hsr.ch/OpenStreetMap`.

Haklay, M. How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and planning. B, Planning & design*, 37 (4):682, 2010.

Haklay, M. and Weber, P. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE*, 7(4):12–18, 2008.

Han, J., Kamber, M., and Pei, J. *Data mining: concepts and techniques.* Morgan kaufmann, 2006.

HERE. Here Map Creator, June 2014. URL `http://here.com/mapcreator`.

Herrera, F., Carmona, C. J., González, P., and del Jesus, M. J. An overview on subgroup discovery: foundations and applications. *Knowledge and information systems*, 29(3):495–525, 2011.

Hilbers, H., van de Coevering, P., Hoorn, A., and van Kempen, G. *Openbaar vervoer, ruimtelijke structuur en flankerend beleid: de effecten van beleidsstrategieën.* Planbureau voor de Leefomgeving, 2009.

Jain, A. K., Duin, R. P. W., and Mao, J. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, 2000.

Jawad, A. and Kersting, K. Kernelized Map Matching for Noisy Trajectories. *SIG SPATIAL*, 2010.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, 2002.

Kaplan, E. D. and Hegarty, C. J. *Understanding GPS: principles and applications.* Artech house, 2005.

Kleijer, F., Odijk, D., and Verbree, E. Prediction of GNSS Availability and Accuracy in Urban Environments Case Study Schiphol Airport. In *Location Based Services and TeleCartography II, pages=387–406, year=2009, publisher=Springer.*

Lemmens, M. *Geo-information: Technologies, Applications and the Environment.* Geotechnologies and the Environment. Springer, 2011. ISBN 9789400716674. URL `http://books.google.nl/books?id=n_tUAWYg4UQC`.

Marchal, F., Hackney, J., and Axhausen, K. W. Efficient map matching of large global positioning system data sets: Tests on speed-monitoring experiment in Zürich. *Transportation Research Record: Journal of the Transportation Research Board*, 1935(1):93–100, 2005.

Marchal, F. TrackMatching, December 2013. URL `https://mapmatching.3scale.net/`.

Miller, H. J. *Geographic data mining and knowledge discovery.* Blackwell Publishing, 2008.

Miller, R. J. and Yang, Y. Association rules over interval data. In *ACM SIGMOD Record, volume=26, number=2, pages=452–461, year=1997, organization=ACM.*

Nakatsu, R. and Iacovou, C. An Investigation of User Interface Features of Crowdsourcing Applications. In *HCI in Business: First International Conference, HCIB 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings*, pages 410–418. Springer, 2014.

Newson, P. and Krumm, J. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages=336–343, year=2009, organization=ACM.*

Nieuws, R. TomTom tipt politie over hardrijder, June 2014. URL `http://www.rtlnieuws.nl/economie/tomtom-tipt-politie-over-hardrijder`.

Ochieng, W. Y., Quddus, M., and Noland, R. B. Map-matching in complex urban road networks. *Revista Brasileira de Cartografia*, 2(55), 2009.

OpenStreetMap. OpenStreetMap, December 2013a. URL `http://www.openstreetmap.org/`.

OpenStreetMap. OpenStreetMap Wiki, December 2013b. URL `http://wiki.openstreetmap.org/`.

OpenStreetMap. OpenStreetMap Copyright, March 2014. URL `http://www.openstreetmap.org/copyright`.

Pluijmers, Y. and Weiss, P. Borders in Cyberspace: Conflicting Public Sector Information Policies and their Economic Impacts, 2002.

Quddus, M. A., Noland, R. B., and Ochieng, W. Y. A high accuracy fuzzy logic based map matching algorithm for road transport. *Journal of Intelligent Transportation Systems*, 10(3):103–115, 2006.

Quddus, M. A., Ochieng, W. Y., and Noland, R. B. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312–328, 2007.

Reynolds, D. Gaussian mixture models. *Encyclopedia of Biometrics*, pages 659–663, 2009.

Rijkswaterstaat. Rijkswaterstaat Maximumsnelheid, Ministry of Infrastructure and Environment, April 2014. URL `http://www.rijkswaterstaat.nl/actueel/verhogingmaximumsnelheid/kaart/`.

Taylor, M. A., Woolley, J. E., and Zito, R. Integration of the global positioning system and geographical information systems for traffic congestion studies. *Transportation Research Part C: Emerging Technologies*, 8(1):257–285, 2000.

Theodoridis, S. and Koutroumbas, K. *Pattern Recognition, Fourth Edition.* Academic Press, 4th edition, 2008. ISBN 1597492728, 9781597492720.

van Loenen, B. The impact of access policies on the development of a national GDI. *Geodaten-und Geodienste-Infrastrukturen-von der Forschung zur praktischen Anwendung. Munster*, 2003.

van Loenen, B. and Crompvoets, J. De staat van de geo-informatie infrastructuur is in Nederland zo slecht nog niet.

Velaga, N. R., Quddus, M. A., and Bristow, A. L. Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems. *Transportation Research Part C: Emerging Technologies*, 17(6):672–683, 2009.

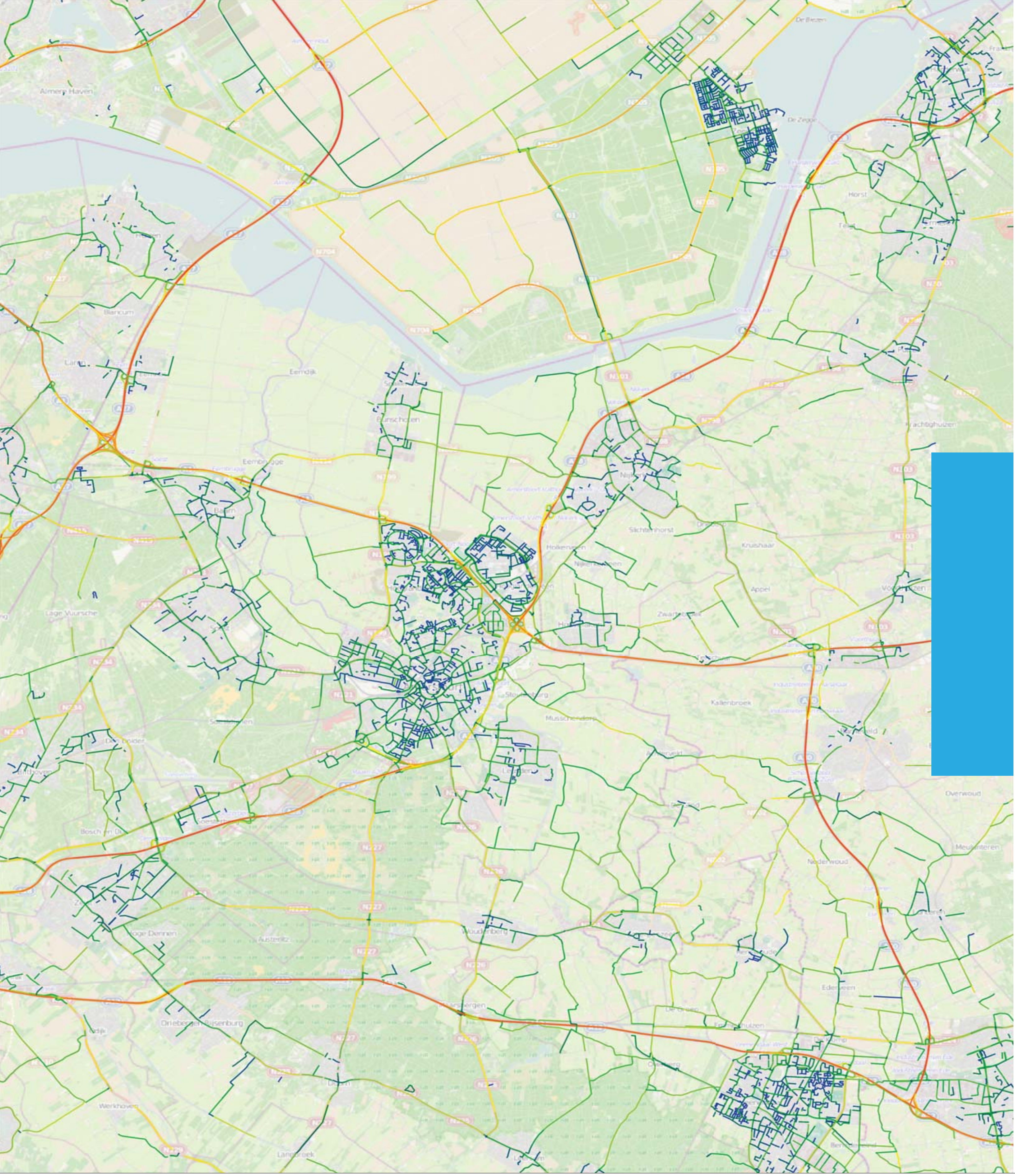
WegenWiki. WegenWiki Rijstrook, March 2014. URL `http://www.wegenwiki.nl/Rijstrook`.

Winter, M. and Taylor, G. A modular neural network approach to improve map-matched GPS positioning. In *Web and Wireless Geographical Information Systems, pages=76–89, year=2006, publisher=Springer*.

Zhang, L., Thiemann, F., and Sester, M. Integration of GPS traces with road map. In *Proceedings of the Second International Workshop on Computational Transportation Science, pages=17–22, year=2010, organization=ACM*.

Zheng, Y., Chen, Y., Li, Q., Xie, X., and Ma, W.-Y. Understanding transportation modes based on GPS data for web applications. *ACM Transactions on the Web (TWEB)*, 4(1):1, 2010.

Zielstra, D. and Zipf, A. A comparative study of proprietary geodata and volunteered geographic information for Germany. In *13th AGILE international conference on geographic information science, year=2010*.


Almere Haven
De Biezen
De Zegge
Horst
Blaricum
Laren
Eemdijk
Bunschoten
Eembrugge
Baarn
Nijkerk
Hoevelaken
Nijkerkerveen
Slichtenhorst
Kruishaar
Appel
Zwartebroek
Voorthuizen
Krachtighuizen
Lage Vuursche
Soest
Stoutenburg
Musschendorp
Kallenbroek
Soesterberg
Den Dolder
Bilthoven
Leusden
Achterveld
Overwoud
Bosch en Duin
Zeist
Hoge Dennen
Austerlitz
Woudenberg
Nederwoud
Meulunteren
Renswoude
Ederveen
Driebergen-Rijsenburg
Werkhoven
Doorn
Langbroek
De Klomp
Bennekom

Reflection

This research was conducted from November 2013 till June 2014, a total of 8 months. The research started with literature reviews, searching and reading all the related works in the field of the topic: "Automatically deriving and updating attribute road data from movement trajectories". At this point the research question was more focused on OSM and GPS tracks in particular, however in a later stage it came to mind that this research is valuable to a bigger picture and can be applied for all kinds of maps and all kinds of movement trajectories.

After the literature review the preparation of the data was important. The GPS data and the OSM data was already available, however the data still lacked the matched roads to the points. Fortunately, the TrackMatching application was found and this was the solution of which would have been a time consuming problem. With the data prepared the development of the attribute extraction algorithms could begin.

The first thing was visualizing and understanding the data, knowing what kind of outliers are present and which elements of the data could be beneficial for attribute extraction. This became more clear throughout the research process and new problems and solutions occurred along the way. Most algorithms were developed on a trial and error basis, different techniques and thresholds were trained and tested to acquire the best results. Slowly, but steady the algorithms were finalized after which the validation could start.

Validation was a difficult part of the research, since there was not many related work available and also ground truth was difficult to find, biased or not available. Nonetheless, some attributes had a high classification accuracy. There were also some classification accuracies of attributes, like the "maxspeed" attribute, which were slightly lower than expected. However, for a first research in this field these results are hopeful for future works.

During the entire research it was managed to stick to the planning. Although some problems occurred during the research, there was no significant delay in the research. This was partially due to good planning, but also due to the facilities and data that were made available by the supervisors. Since all that was needed was provided at the beginning of the thesis, the research could start immediately.

The final products are multiple scripts containing the codes for all steps that are taken in this research, from the map matching and preprocessing to the actual extraction and updating of the attributes. These scripts could be implemented into one script that automatically preprocess the data, extracts and update the attributes.

This research has a strong relation with the field of geomatics. Movement trajectories are used as input and the attributes for the roads are the output. Both the input data and the output data are geometrical data or attributes of geometrical data.
The methodological line of approach in this research can be compared to the approach of the Master

Geomatics. The work flow in the Master can be categorized in five steps:

- Data Capture
- Data Storage
- Analysis
- Communication/Visualization
- Quality Control

The data capture step was already performed in advance of the research by the people carrying the GPS device and the OSM contributors. The data storage step was partially performed in advance, but is expanded by the preprocessing and updating in this research. The analysis and visualization steps in this research were the most important steps. These steps helped to understand the data and extract useful information out of it that results into the updated attributes.

This research contributes to all different kinds of parts in society. It investigates if movement trajectories can be used for automatically deriving and updating attribute road data. Automatically deriving this kind of information saves a vast amount of time and money compared to manual extraction. The results could be used for various applications as navigation systems, traffic analyses, urban development and civil engineering and more.