DGK Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 817

Urs Fabian Bock

Dynamic Parking Maps from Vehicular Crowdsensing

München 2018

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5232-1

Diese Arbeit ist gleichzeitig veröffentlicht in:
Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Universität Hannover
ISSN 0174-1454, Nr. 342, Hannover 2018

DGK Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 817

Dynamic Parking Maps from Vehicular Crowdsensing

Von der Fakultät für Bauingenieurwesen und Geodäsie
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades
Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation

Vorgelegt von

Dipl.-Phys. Urs Fabian Bock

Geboren am 02.03.1987 in Lebach

München 2018

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5232-1

Diese Arbeit ist gleichzeitig veröffentlicht in:
Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Universität Hannover
ISSN 0174-1454, Nr. 342, Hannover 2018

Adresse der DGK:

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)
Alfons-Goppel-Straße 11 ● D – 80 539 München
Telefon +49 – 331 – 288 1685 ● Telefax +49 – 331 – 288 1759
E-Mail post@dgk.badw.de ● http://www.dgk.badw.de

Prüfungskommission:

Vorsitzender:	Prof. Dr.-Ing. Ingo Neumann
Referent:	Prof. Dr.-Ing. Monika Sester
Korreferenten:	Prof. Dr.-Ing. Christian Heipke Prof. Dr.-Ing. Bernhard Friedrich

Tag der mündlichen Prüfung: 13.04.2018

© 2018 Bayerische Akademie der Wissenschaften, München

Alle Rechte vorbehalten. Ohne Genehmigung der Herausgeber ist es auch nicht gestattet,
die Veröffentlichung oder Teile daraus auf photomechanischem Wege (Photokopie, Mikrokopie) zu vervielfältigen

ISSN 0065-5325
ISBN 978-3-7696-5232-1

Abstract

The search for parking is a highly important problem in many cities. Drivers circle around the blocks on a more or less random search path when looking for on-street parking, wasting time and resources. Thus, parking information has great potential to reduce the search time by guiding the drivers to road segments with a higher chance of finding parking. The goal of this thesis is the automatic generation of *dynamic parking maps*, which contain both the areas with parking permission and the estimated parking availability. As input, the data collection by *vehicular crowdsensing* is investigated, where sensors of many vehicles continuously scan the availability of on-street parking spaces during their regular trips. This goal is addressed holistically, starting with the detection of parked vehicles in parking lanes, over the aggregation of parking availability information, to the quality evaluation of the dynamic parking maps.

The main body of this thesis is split into two parts regarding the underlying data: 1) self-recorded street environment data from a LiDAR mobile mapping vehicle and 2) long-term datasets from static parking sensors and taxi trajectories in San Francisco, US. First, a processing pipeline is presented to extract the locations of parked vehicles from the 3D point clouds of the LiDAR mobile mapping vehicle. A key challenge was the distinction between moving and parked vehicles. Results showed that this pipeline was able to provide accurate results with a recall of 93.7 % at a precision of 97.4 %. Based on detections of parked vehicles at different time instants, a learning approach was developed, in order to generate a parking legality map automatically, containing the areas where parking was allowed. Two classifiers using several features were compared with an approach from the literature. The results showed that the proposed approach using a random forest classifier achieved the best results. Most interestingly, a clustering-based classification achieved a similar quality, without the need for expensive training data.

Since crowdsensing needs many vehicles and thus records of one vehicle are not sufficient to study crowdsensing of parking availability comprehensively, the observations from more than 5,000 static parking sensors were used to simulate crowdsensing. Crowdsensing was simulated by downsampling of the regular sensor observations, based first on an observation rate model and then on trajectories from taxis as potential probe vehicles. The parking data was analyzed in a detailed spatio-temporal evaluation. The main results were that parking had a strong daily periodicity and that the temporal correlations were much more relevant than spatial similarities. With crowdsensing, parking availability is not observed at regular intervals, and therefore needs to be estimated at unobserved time instants. Three methods for parking availability estimation were investigated. Spatial interpolation achieved only modest results. However, a simple persistence of the last observation value on the same road segment achieved remarkable results, unsurpassed by a binary classification. However, the binary classification has the benefits that 1) a better weighting between precision and recall is possible, 2) an accurate probability estimate is provided, and 3) the method can also be used for an availability prediction. In a comparison between taxi crowdsensing and continuous measurements by static sensors, only a small quality decay was observed for crowdsensing. In a final evaluation of the different parking information for an exemplary search scenario, the parking legality map with capacity information led to a strong search reduction. Further improvements were observed by including parking availability information, particularly when the parking capacity was similar in the road segments and thus the parking legality map helped less.

In conclusion, parking crowdsensing with probe vehicles has high potential for providing detailed dynamic parking maps without the need for static sensors in the streets. Moreover, dynamic parking maps significantly support drivers on the parking search.

Key words: Smart parking search, machine learning, collaborative sensing

Zusammenfassung

In vielen Städten ist die Suche nach einem freien Parkplatz ein großes Problem. Autofahrer suchen auf einer mehr oder weniger zufälligen Route nach Parkplätzen am Straßenrand und verschwenden dabei Zeit und Sprit. Parkplatzinformationen haben daher ein großes Potential diese Suche zu erleichtern, etwa durch eine Routenführung zu den Straßenabschnitten mit der höchsten Wahrscheinlichkeit, einen Parkplatz zu finden. Das Ziel dieser Arbeit ist die automatische Erstellung dynamischer Parkplatzkarten, die sowohl die Flächen, auf denen man parken darf, als auch eine Schätzung der aktuellen Parkplatzverfügbarkeit beinhalten. Dabei wird das Sammeln der benötigten Daten mittels Messungen einer Vielzahl an Fahrzeugen (Crowdsensing) untersucht, die die aktuelle Parkplatzverfügbarkeit auf ihren regulären Fahrten über eingebaute Sensoren erfassen. Dieses Ziel wird dabei ganzheitlich verfolgt, von der Detektion parkender Fahrzeuge in Parkbuchten, über die Aggregation von Informationen zur Parkplatzverfügbarkeit, hin zu einer Qualitätsbewertung der dynamischen Parkplatzkarten.

Der Kern der Arbeit teilt sich hinsichtlich der verwendeten Daten in zwei Hauptteile: 1) selbsterfasste Daten des Straßenumfeldes mit Hilfe eines LiDAR Mobile Mapping Fahrzeuges und 2) Langzeitdaten von statischen Parkplatzsensoren und Taxitrajektorien in San Francisco (USA). Zuerst wird ein Vorgehen zur Erkennung von parkenden Autos aus den 3D-Punktwolken des LiDAR Mobile Mapping Fahrzeugs beschrieben. Eine Hauptherausforderung ist dabei die Unterscheidung zwischen fahrenden und parkenden Fahrzeugen. Die Ergebnisse zeigen, dass mit diesem Vorgehen sehr genaue Ergebnisse mit einer Sensitivität von 93,7 % bei einer Genauigkeit von 97,4 % erreicht werden. Basierend auf den Erkennungen von geparkten Fahrzeugen zu unterschiedlichen Zeitpunkten wird ein Lernansatz vorgestellt, um automatisch eine Parkraumkarte zu erzeugen, die die erlaubten Parkbereiche enthält. Zwei Klassifikationsverfahren, die verschiedene raumzeitliche Merkmale nutzen, werden mit einem Ansatz aus der Literatur verglichen. Die Ergebnisse zeigen, dass der Random Forest Klassifikator die besten Ergebnisse erzielt. Interessanterweise erzielt eine clustering-basierte Klassifikation eine ähnliche Qualität ohne die Nutzung aufwendiger Trainingsdaten.

Da Crowdsensing eine Vielzahl an Fahrzeugen benötigt und somit die Messungen eines einzelnen Fahrzeugs nicht ausreichen, um ein Crowdsensing der Parkplatzverfügbarkeit umfassend zu untersuchen, werden Messdaten von mehr als 5.000 statischen Parksensoren in San Francisco verwendet. Crowdsensing wird durch eine Ausdünnung der Sensorbeobachtungen simuliert, zunächst basierend auf einem Modell mit konstanter Beobachtungsrate und anschließend basierend auf Trajektorien von Taxis als potentielle Messfahrzeuge. Darüber hinaus werden die Parkplatzdaten in einer detaillierten raumzeitlichen Auswertung analysiert. Die Hauptergebnisse dieser Auswertung zeigen, dass die Parkplatzbelegung eine starke tägliche Periodizität aufweist und dass die zeitliche Korrelation deutlich relevanter ist als die räumlichen Ähnlichkeiten. Beim Crowdsensing wird die Parkverfügbarkeit nicht in regelmäßigen Abständen beobachtet und muss daher für die unbeobachteten Zeitpunkte geschätzt werden. Drei Methoden zur Schätzung der Parkplatzverfügbarkeit werden untersucht. Die räumliche Interpolation erzielt nur mäßige Ergebnisse. Aber bereits ein einfaches Fortführen des letzten Beobachtungswertes auf demselben Straßensegment (Persistenzmethode) erzielt bemerkenswerte Ergebnisse, die von einer binären Klassifikation nicht übertroffen werden. Die binäre Klassifizierung hat jedoch die Vorteile, dass 1) eine bessere Gewichtung zwischen Genauigkeit und Sensitivität möglich ist, 2) eine genaue Wahrscheinlichkeitsschätzung bereitgestellt wird und 3) auch eine Prädiktion der Parkplatzverfügbarkeit möglich ist. In einem Vergleich zwischen Crowdsensing mittels Taxis und kontinuierlichen Messungen durch statische Sensoren wird nur ein geringer Qualitätsabfall für Crowdsensing beobachtet. Abschließend werden die verschiedenen Informationen der dynamischen Parkplatzkarte in einem beispielhaften Suchszenario ausgewertet. Bereits die Verwendung der Parkraumkarte mit der Parkplatzkapazität führt zu einer starken Reduktion der Suche. Weitere Verbesserungen werden bei Nutzung der Informationen zur Parkplatzverfügbarkeit beobachtet, insbesondere wenn die Parkkapazität in den Straßenabschnitten ähnlich ist und daher die Parkraumkarte weniger Unterstützung bietet.

Zusammenfassend kann festgestellt werden, dass das Parkplatz-Crowdsensing mit Messfahrzeugen ein hohes Potenzial für die Bereitstellung dynamischer Parkplatzkarten bietet, ohne dass statische Sensoren in den Straßen benötigt werden. Darüber hinaus helfen dynamische Parkplatzkarten den Autofahrern erheblich bei der Parkplatzsuche.

Schlüsselwörter: intelligente Parkplatzsuche, maschinelles Lernen, kollaborative Erfassung

Contents

1 Introduction 13
1.1 Background and motivation 13
1.2 Goals of this thesis 15
1.3 Outline 17

2 Background on digital maps and data mining 19
2.1 Digital maps 19
2.1.1 Navigation maps and map dynamics 19
2.1.2 OpenStreetMap 21
2.1.3 Navigation Data Standard (NDS) 21
2.2 Data mining 23
2.2.1 Knowledge Discovery in Databases (KDD) process 23
2.2.2 Taxonomy of data mining methods 24
2.2.3 Classification 25
2.2.4 Clustering 30
2.2.5 Time series analysis 34

3 Related work about mobile crowdsensing of on-street parking spaces 37
3.1 On-street parking 37
3.1.1 Parking occupancy detection 37
3.1.2 Parking availability estimation and prediction 39
3.1.3 Parking search and guidance 40
3.2 Mobile crowdsensing 41
3.2.1 Mobile crowdsensing in transportation 41
3.2.2 Mobile crowdsensing for parking 42
3.3 Research gaps addressed in this thesis 42

4 LiDAR-based parking availability data acquisition 45
4.1 Data recording 45
4.1.1 Sensor equipment 45
4.1.2 Measurement campaign 45
4.2 Methodology 46
4.2.1 Preprocessing 47
4.2.2 Segmentation 48
4.2.3 Classification 49
4.2.4 Repetition of segmentation and classification 50

4.2.5 Matching to road network 50
4.3 Results 50
4.3.1 Object segmentation 52
4.3.2 Classification 52
4.3.3 End-to-end evaluation of complete approach 54
4.3.4 Parking occupancy statistics over the day 55
4.4 Concluding remarks 55

5 Learning parking legality maps from parking observations 57
5.1 Methodology 57
5.1.1 Location of parked vehicles as method input 57
5.1.2 Data preprocessing 59
5.1.3 Definition of feature sets 59
5.1.4 Learning the parking legality of road subsegments 60
5.2 Evaluation 61
5.2.1 Evaluation approach 61
5.2.2 Results 64
5.3 Concluding remarks 68

6 Spatio-temporal analysis of large scale parking availability data and simulation of crowdsensing 71
6.1 Description and processing of parking dataset from SFpark 71
6.2 Time series analysis of parking availability data 74
6.3 Clustering of parking occupancy daily pattern 74
6.4 Spatial relations in parking availability 76
6.5 Modelling of crowdsensing based on downsampling for probe vehicles and mobile apps 80
6.5.1 Scenario based on probe vehicles 81
6.5.2 Scenario based on mobile apps 82
6.6 Modelling of probe-vehicle-based crowdsensing from taxi GPS trajectories 82
6.6.1 Processing overview and description of taxi trajectory dataset 83
6.6.2 Taxi GPS trajectory processing 83
6.6.3 Characteristics and aggregation of taxi coverage 85
6.6.4 Comparison of parking and taxi daily pattern 88
6.6.5 Simulation of parking availability observations 89
6.7 Concluding remarks 91

7 Parking availability estimation and prediction from crowdsensed data 93
7.1 Spatial interpolation of parking availability 93
7.2 Parking availability estimation with persistence method 95
7.3 Estimation and prediction of parking availability based on binary classification . . 98
7.3.1 Binary classification approach 98
7.3.2 Results of binary classification estimation and prediction 99

7.4 Concluding remarks 102

8 Benefits of crowdsensed parking availability information 107
8.1 Types of information for on-street parking 107
8.2 Experimental setup 108
8.2.1 Routing strategies 109
8.2.2 Data sources 109
8.3 Evaluation of the impact of different parking information 110
8.3.1 Results for all decisions in the dataset 110
8.3.2 Results for relevant decisions 111
8.3.3 Similarity of capacity 113
8.4 Concluding remarks 113

9 Conclusion and outlook 117
9.1 Research questions addressed and overall conclusion 117
9.2 Applicability of dynamic map approaches to further dynamic phenomena 119
9.3 Future research directions 121

List of figures 125

List of tables 127

References 129

Acknowledgements 137

Curriculum vitae 138

1 Introduction

1.1 Background and motivation

The search for an empty parking space is a highly important problem for many drivers. When drivers cannot find an empty parking space at their destination, they start to roam around the blocks looking for the next empty spot. This phenomenon, called *parking search traffic*, occurs particularly in cities where many people live in dense areas. The additional driving leads to stress, as well as wasted time and money for the drivers. In addition, other traffic participants suffer from this problem: the parking search traffic increases the total traffic volume (Shoup, 2006), drivers go at a lower speed when they are trying to spot a gap (Hampshire et al., 2016), and thus, all other drivers are slowed down just because of missing parking spaces. The number of traffic accidents is also increased due to the search for parking (Bush and Chavis, 2017). Finally, the environment suffers from needless pollution because of additional fuel consumption.

Many studies have quantified the problem of parking search. Shoup (2006) surveyed several of them from all around the globe. He concluded that, on average, 30 % of total traffic is just parking search traffic in the study areas. The average search time is more than eight minutes[1]. However, as these studies were conducted in problematic districts at problematic hours of the day, these numbers represent rather an upper bound of the problem. In another study, Shoup (2007) estimated that, in a small business district in Los Angeles called Westwood Village with 470 metered parking spaces, parking search cruising adds up to 950,000 miles per year, with pollution of 730 tons of CO_2. Van Ommeren et al. (2012) conducted a nationwide survey to find out the extent of parking search in the Netherlands. When excluding employer-provided and residential parking, 30 % of the trips ended with some parking search. Averaged over all trips, the search time was about 36 seconds. They also found that parking search was more common in large cities. In San Francisco, Millard-Ball et al. (2014) found that parking search traffic is spatially heterogeneous. A small number of blocks is already responsible for half of the parking cruising in the area investigated.

A promising solution to reduce parking search is the provision of parking information to drivers. For large parking lots and garages, signposts have for decades indicated the directions to parking locations and a count of empty parking spaces (Axhausen et al., 1994). These systems are generally called *Parking Guidance and Information* (PGI) systems. The number of empty parking spaces can easily be obtained from the barriers that count incoming and departing vehicles. For on-street parking spaces, the acquisition of parking availability information is more difficult and expensive. Several types of static sensors have been suggested to monitor the availability in every road segment. However, as such systems are expensive to install and maintain (Xu et al., 2013; Lin et al., 2017), only a few traffic management authorities deploy this technology, mostly in just a few blocks of the city.

An emerging alternative is the use of mobile sensors. Smartphones and modern vehicles are equipped with a variety of sensors that allow the detection of parking-related information. With

---

[1] Note that the ratio of parking search traffic and the average search time do not bear a direct relation. For example, as extreme case, only 1 % of the traffic could search for parking for hours, if there is a large amount of through traffic and there are few parking spaces. Contrary, 50 % of the traffic could search for parking, but they may only spend a few seconds because of a large parking supply in every second road.

a large number of these mobile sensors, mobile crowdsensing might cover a complete city at a satisfactory frequency. Most existing smartphone parking availability applications aim to detect parking and leaving events. These events are detected either directly from GPS and accelerometer signals, or indirectly from the transition of detected travel modes or connection build-up between vehicle and smartphone (e.g. via Bluetooth) (Lin et al., 2017). In addition, there is also the approach to detect parking search by identification of characteristic movement patterns in the GPS traces (Weinberger et al., 2017). Dozens of start-ups (e.g. Parknav[2], Parkbob[3]) with such approaches offer their parking application in the app stores. None of them has reached a wide user acceptance so far. However, a high user rate might be necessary for good data quality (Lendák and Farkas, 2016). Recently, larger companies (e.g. TomTom[4], Inrix[5]) have also started to offer on-street parking data products.

Smart vehicles not only detect their parking and leaving events, but they also have the potential to scan parking availability in the parking lanes during their regular trips. In research studies, ultrasonic sensors or cameras were used to continuously observe parking availability at the side of the vehicle (Mathur et al., 2010; Sivaraman and Trivedi, 2013). With the efforts towards autonomous vehicles, laser scanners are becoming more popular, but there is little research to detect parking availability from these sensors (e.g. Thornton et al., 2014). When the vehicles detect a gap in the parking lane, they could communicate the location and length of the gap to a central server instance (see also Figure 1.1). This instance collects all detected gaps and compares them to a parking map, to distinguish between empty parking spaces and no-parking zones. Information about the parking spaces can then be sent to the navigation systems or smartphone applications of drivers on the search in the form of *dynamic parking maps*. Similar concepts were announced recently from automotive suppliers and car manufacturers (e.g. Robert Bosch GmbH, 2016; Daimler AG, 2017), but such products are not on the market yet.

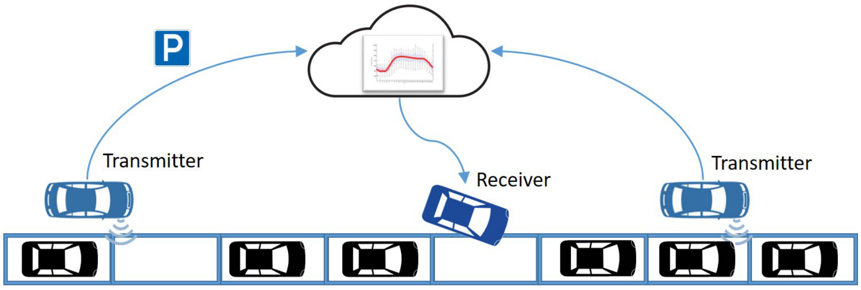


Figure 1.1: Concept of vehicular crowdsensing for parking availability. Vehicles scan for gaps in the parking lane during their regular trips and communicate gap information to a central server instance. The parking gap information is aggregated there and sent to searching vehicles. The illustration is based on the work by Robert Bosch GmbH (2016).

Not many researchers have addressed the exploitation of crowdsensed parking data from smart vehicles so far (Lin et al., 2017). To generate parking information for drivers, knowledge about the location of parking spaces and the current parking availability is essential. Many cities do not have detailed maps of parking lanes, or they are outdated. Thus, data-driven methods can be applied to derive parking maps from crowdsensed parking data (Ge et al., 2013; Coric and Gruteser, 2013). However, machine learning methods have not yet been applied to improve the quality of parking maps. Regarding the parking availability, as sensing vehicles do not scan all parking lanes at all time instants, it is highly relevant to estimate the missing availability information. Research exists

---

[2] http://parknav.com/
[3] http://www.parkbob.com/
[4] https://automotive.tomtom.com/products-services/connected-services/parking/
[5] http://inrix.com/products/parking/

only for the case of parking crowdsensing with smartphones where the data input is just parking and leaving events (Xu et al., 2013), but not scans of a complete road segment. To suggest other destinations or travel modes to drivers, prediction of the parking availability is also highly relevant. It is intuitive that the quality of the crowdsensing approach strongly depends on the number of the sensing vehicles and their spatio-temporal coverage. Thus, it is important to study how the number of sensing vehicles and their movement patterns influence the information quality. This problem was addressed only by Mathur et al. (2010) in a simplified evaluation. Finally, the relevance of this parking information and its quality on the parking search needs further investigation, as only studies based on simulation environments exist (Tasseron et al., 2016). Thus, there is still a great need for research in the field of crowdsensing of parking-related information.

1.2 Goals of this thesis

This thesis aims to address the steps for the generation of dynamic parking maps from crowdsensing, from the detection of parked vehicles in parking lanes to the aggregation of parking availability information and the quality evaluation of the maps. In Figure 1.2, an overview of the parts of this thesis is provided. For the generation of a parking crowdsensing dataset, 3D point clouds from a LiDAR mobile mapping system were recorded and processed. In order to also investigate parking crowdsensing for a large number of sensing vehicles and for a longer period, real-world data from static parking sensors and GPS-equipped taxis were used to simulate a second parking crowdsensing dataset. Based on this parking crowdsensing data, the parking legality was derived and the parking availability was estimated. Finally, both parking information types were combined into a dynamic parking map.

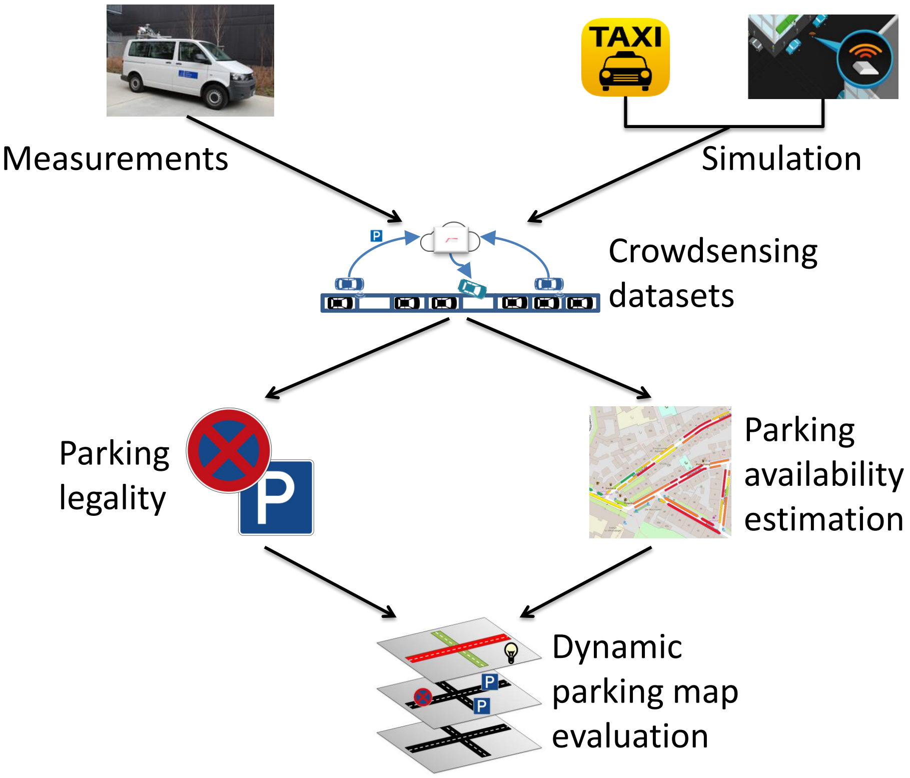


Figure 1.2: Overview of the parts of this thesis, from parking crowdsensing to the generation and evaluation of the dynamic parking map.

The thesis is built on the following main research questions:

- **How can on-street parking occupancy information be obtained with a LiDAR mobile mapping system?**
  A LiDAR mobile mapping system can be used to record precise 3D point clouds of the street environment. The objective is the formulation of a processing pipeline to automatically extract parking occupancy information on the roads from these 3D point clouds.
- **How can parking legality maps, representing the legality of parking on the streets, be derived from crowdsensed parking observations?**
  The locations of parked vehicles are usually a good indicator for the locations of parking spaces. However, if the parking availability is scarce, drivers also tend to park in illegal areas. In contrast, some parking locations are less attractive and thus seldom occupied. The objective of this research question is the derivation of a methodology to learn the parking legality from crowdsensed parking observations at different times.
- **How can parking availability be estimated based on crowdsensed parking availability information?**
  Even if many sensors observe the parking availability on the streets, parking is unobserved in some places at some time instants. Thus, methods are needed to estimate the parking availability in these cases.
- **What is the potential quality of parking crowdsensing with taxis as probe vehicles?**
  Taxis drive in cities around the clock, using all the roads on the road network. Thus, they might be suitable as probe vehicles for crowdsensing. The objective of this research question is the spatio-temporal assessment of the quality of parking availability information from taxis as crowdsensing sources.

In addition to answering these research questions, the following contributions are made, which are directly related to the research questions:

- **Assessment of spatial and temporal characteristics in parking behavior based on an extensive dataset from static parking sensors**
  Parking data from more than 5,000 parking spaces over six weeks were analyzed to extract basic statistics, as well as spatial and temporal characteristics. After cleaning from irregular sensor data, the capacity of parking lanes and the prevalence of different parking occupancy levels and their periodicity was evaluated. Moreover, the existence of spatial and temporal correlations in parking occupancy behavior was investigated. Clustering of parking lanes was performed to identify areas with similar temporal parking characteristics.
- **Examination of the spatio-temporal movement patterns of a large fleet of taxis**
  A taxi trajectory dataset from more than 500 taxis over nearly one month was preprocessed and matched to the road network. Their spatio-temporal coverage was evaluated to examine their suitability for parking crowdsensing.
- **Simulation of parking crowdsensing by taxis based on real-world parking and taxi trajectory datasets**
  Assuming that taxis were equipped with sensors to detect parking availability while driving, a parking crowdsensing dataset was simulated from the parking dataset based on typical taxi activity behavior. In particular, the number of lanes in the road, one-way roads, and the spatio-temporal movement characteristics were considered. As a comparison, simple models for taxi and smartphone crowdsensing were developed based on a constant observation rate, without consideration of spatio-temporal characteristics.

- **Evaluation of the required fleet size of probe vehicles for a sound parking crowdsensing**
  When evaluating the quality of parking crowdsensing with taxis as probe vehicles, different fleet sizes of probe vehicles were considered. That way, the required number of probe vehicles was determined depending on the quality requirements.
- **Relevance of different on-street parking information for the reduction of parking search**
  Knowledge about the parking capacity of roads and about the parking availability from different sources helps searching drivers. Different types of parking information were evaluated in an exemplary scenario for parking search to identify their relevance on the search.

With these contributions, this thesis presents a holistic approach to generate dynamic parking maps from crowdsensing. In addition, thorough analyses of the datasets and the obtained results are presented.

1.3 Outline

This thesis is structured as follows: as parking is a highly dynamic phenomenon that can be represented in maps, basic knowledge about dynamic maps is described, with a focus on navigation maps, their updates, and the Navigation Data Standard (NDS) in Chapter 2. For the processing of large datasets of parking information, methods from the field of data mining are used. Thus, Chapter 2 also contains fundamentals about data mining and the relevant methods, including a description of the evaluation measures used in this thesis. Chapter 3 outlines the related work in the context of on-street parking and mobile crowdsensing. Based on this state of the art, the research gaps that are addressed in this thesis are described.

The main body of the thesis is divided into two parts: in the first part, a self-recorded dataset from a LiDAR mobile mapping system was used to detect parked vehicles on the street (Chapter 4) and to learn the parking legality from vehicle detections recorded at multiple time instants over a complete day (Chapter 5). For the second part, long-term measurements from parking sensors and GPS sensors in the city of San Francisco (US) were exploited to simulate a much larger crowdsensing dataset. The parking data stems from the SFpark project, which provided availability information from more than 5,000 static parking sensors (San Francisco Municipal Transportation Agency, 2014). The taxi GPS trajectories were recorded in a socio-economic project called *Cabspotting* (Piorkowski et al., 2009). The processing and analysis of these raw datasets, as well as the simulation of the crowdsensing datasets are described in Chapter 6. Based on the simulated crowdsensing data, multiple approaches to estimate parking availability are investigated in Chapter 7. Finally, the benefits of different parking information are evaluated based on a model scenario in Chapter 8.

In the last chapter, the answers to the research questions are given and an overall conclusion is drawn. Furthermore, the relationship to other dynamic phenomena in a street environment and the applicability of the presented methods are described. Finally, future research directions based on the results presented are outlined.

2 Background on digital maps and data mining

In this chapter, the basic concepts and techniques relevant for this thesis are presented. First, in Section 2.1, the main aspects of digital maps related to navigation are addressed. In particular, navigation maps and their dynamics, the open-source map platform OpenStreetMap, and the upcoming Navigation Data Standard (NDS) are described. In Section 2.2, the basic principles and terminology of data mining are addressed. Relevant data mining algorithms are explained in detail.

2.1 Digital maps

Digital maps are the foundation for several applications in the field of vehicle navigation and traffic management. Routing algorithms use navigation maps (see Section 2.1.1) to provide route suggestions to drivers. Additional attributes such as Points of Interest (POI) or parking information can be used to inform the driver or can also be included in the routing calculation. An open source map including road information is OpenStreetMap, which is generated voluntarily by many users, as described in Section 2.1.2. This source is used in this thesis to extract road information for the investigation sites. Finally, the Navigation Data Standard (NDS) is a map standard currently under development by a large consortium of car manufacturers and automotive suppliers. This standard is described in Section 2.1.3 with a focus on the potential integration of dynamic parking information.

The term *dynamic map* is not well-defined in the literature and different aspects are related to this name. Within this thesis, dynamic maps are digital maps that are continuously updated with new and near-real-time information. Similarly, dynamic maps are also used in robotics for simultaneous localization and mapping tasks in dynamic environments (e.g. Wolf and Sukhatme, 2005). The term is also used for maps that illustrate locations of dynamic behavior and its temporal variations, such as the dynamic pedestrian map of Schlichting and Brenner (2016). In addition, the term is used in the domain of map visualizations, where the map illustration continuously changes (e.g. Hu et al., 2012).

2.1.1 Navigation maps and map dynamics

Navigation maps are a digital representation of the real road network in the form of a routable graph and additional information related to the nodes and edges of the graph. Typically, an edge represents a road segment between two intersections, which are represented by nodes (Kleine-Besten et al., 2015). Additional information includes road geometries, street names, road classes, zip codes, speed limits, turn restrictions, and further types of background information (Bill, 2016). To compute a route from start to destination, a shortest path is searched for in the graph. Edge costs of the graph may be based on length, travel time, and/or road classes (Kleine-Besten et al., 2015). The main commercial providers of navigation maps are TomTom[1] and HERE[2]. However,

---

[1] https://www.tomtom.com/

[2] https://here.com/

more and more navigation solutions are based on crowdsourced map data from OpenStreetMap (e.g. Skobbler[3]).

Modern navigation maps are not only used for route guidance and driver information, but also for driving assistance systems. The electronic horizon provides information such as road curvature or hazard spots on the road ahead, which can be integrated into warning systems (Kleine-Besten et al., 2015). Map information can also support the sensor perception of the environment and enhance vehicle localization (Matthaei et al., 2015).

According to the book chapter by Kleine-Besten et al. (2015), changes in the environment lead to errors in navigation maps. Thus, they must be updated by the providers at regular intervals. For example, TomTom has provided a map update once per quarter year since 2008 (TomTom, 2017). Traditionally, these updates are sold to customers, who transfer the updates to the navigation device with a storage medium. In recent years, updates are also available via an online connection. To avoid large data traffic for updating maps that cover a country or even a continent, regular local updates in the area of residence or of regular trips are proposed. Consequently, the navigation maps for the vehicle might become more personalized. In the future, a personalized map might learn the preferences of drivers and contain user-group-specific map content (e.g. support for elderly people).

Near-real-time information such as traffic information is broadcast via non-audible very high frequency (VHF) radio signals (Radio Data System - Traffic Message Channel (RDS-TMC), Schmidt et al., 2009; Kleine-Besten et al., 2015) and mobile communication. A TMC message consists of a predefined event code, location code, and an optional expiration time. The ‘Event Code List’ contains more than 1400 event types about traffic-related information such as traffic jams, construction sites, or accidents. The ‘Location Code List’ encodes relevant locations such as highway sections, highway ramps, intersections, and parking lots in a 16 bit code (up to 65,536 entries possible). For Germany, there are about 40,000 locations listed. This standard was tailored for communication with low data transfer rates. An enhancement is the TPEG (Transport Protocol Experts Group) coding standard to transfer more detailed information about traffic, travel, weather, public transportation, parking, and more (Schmidt et al., 2009). TPEG is more flexible and scalable than TMC. It is independent of the method of communication and the type of navigation map because of standardized location references such as AGORA-C (Navigation Data Standard (NDS) e.V., 2016). A traffic message expires if either the expiration time is reached or a delete message is broadcast.

While offline route calculations were once the standard in navigation systems, modern communication technologies allow online or hybrid navigation (Kleine-Besten et al., 2015). In these cases, parts of or all routing calculations are performed on a backend server, and the result is transferred to the client in the car. That way, more complex algorithms can be used and less information needs to be processed and stored on the client side. However, these solutions are dependent on a continuous network connection. In addition, vehicle sensor data can be transferred to the backend server, where information is collected and aggregated from many vehicles (e.g. extraction of traffic information from floating car data) (Klanner and Ruhhammer, 2015).

---

[3] https://www.skobbler.com/

2.1.2 OpenStreetMap

OpenStreetMap[4] is a project that provides a free world map, currently at a very detailed level in many places worldwide. It helps people in regions where map data is proprietary, restricted, or not available in such detailed granularity. More than four million users contributed about four billion nodes to the map up to the end of 2017 (OpenStreetMap, 2017b). The integrated data stems from GPS devices, aerial photography, and manual annotations, as well as data imports from public institutions. While the data quality is high in areas with many contributors, it is still very heterogeneous, depending on the region.

In the context of this thesis, mainly the road network from OpenStreetMap is used. Several evaluations about the coverage of taxis and parking sensing are based on the road class as defined by OpenStreetMap. This definition is rather fuzzy and may differ from the road class definitions of other providers. Moreover, contributors might not follow the official description. The most relevant road class keys are, in descending priority (OpenStreetMap, 2017a):

- **Motorway:** Highest road class with two or more lanes in both directions and on- and off-ramps.
- **Trunk:** Most important roads that are not motorways, often with on- and off-ramps.
- **Primary:** Main (often multi-lane) roads in cities / links between larger towns.
- **Secondary:** Medium-large roads in cities / roads that link towns.
- **Tertiary:** Small roads in cities with low to moderate traffic / roads that link smaller towns and villages.
- **Unclassified:** Least important road class that serves a purpose other than access to properties. Note that ‘unclassified’ is a defined class and does not stand for a missing classification.
- **Residential:** Serve access to housing, not meant for through traffic.
- **Service:** Access roads to or within private area (e.g. car park or industrial estate).

2.1.3 Navigation Data Standard (NDS)

The Navigation Data Standard (NDS) is a registered association of industry partners that aims to develop a standard for navigation maps. Numerous car manufacturers, car suppliers, map data providers, and telematics service providers are members of this association. The following descriptions of NDS are mainly based on the book chapter by Behrens et al. (2015).

The goals of the NDS association are the worldwide use of their map standard, the separation of navigation software and data, and the definition of procedures for update and extension of navigation data. The standardization is an ongoing process. The first products based on NDS have been available on the market since 2013.

The hierarchy for navigation data in the NDS map format is the following: the navigation data splits into regions (e.g. different countries). For each region, there are several *building blocks* for different purposes. They are the main organization units that can be exchanged and adapted. Each building block finally divides into small spatial parts called *tiles*. Across all building blocks, data is stored in a database based on SQLite, which allows efficient incremental updates and search across all building blocks.

---

[4] https://www.openstreetmap.org/

There are 14 building blocks that are specific for each region:

- **Overall Building Block:** contains meta data and region-specific information that are accessible for all other building blocks.
- **Routing Building Block:** consists of the road network plus attributes such as speed limit, road class, curvature, and road width.
- **Traffic Information Building Block:** allows the use of traffic information from different standards. It contains the reference tables for TMC and TPEG.
- **POI Building Blocks:** contains location information (Points of Interest) plus additional attributes such as phone number or opening hours. They can be point, line, or area information. There are two building blocks for integrated (referenced to an edge or node) and not-integrated POIs.
- **Naming Building Block:** contains the names of elements in the Routing and Basic Map Display Building Blocks.
- **Phonetic/Speech Building Block:** contains the phonetic transcriptions and recorded speech patterns for voice announcements.
- **Index Structure Building Blocks:** consists of the SQLite Index Building Block for a fast search for destination input and the Free Text Search Building Block, which contains an index for a free text search across all contents.
- **Basic Map Display Building Block:** responsible for the main data needed for map display: points (e.g. city center), lines (e.g. street geometries), areas (e.g. land use), icons, and 2.5D city models.
- **Advanced Visualization Building Blocks:** consists of the Digital Terrain Model Building Block, the Orthoimages Building Block, the 3D Objects Building Block, and the Junction View Building Block.

Dynamics in the environment are handled in three ways in the NDS standard: map updates, daily patterns, and real-time information messages. Infrastructure changes, such as the opening of a new road or changes in the turn restrictions, are recorded by map providers and released in regular map updates (see also Subsection 2.1.1). Phenomena with typical and periodic changes such as travel speeds can also be stored in the map. In the case of travel speeds, the typical daily pattern (speed profiles) are stored relative to the free flow speeds. To reduce data storage, speed profiles are often not stored for each road segment individually, but some representative speed profiles are stored and all road segments are referenced to them (Navigation Data Standard (NDS) e.V., 2016). For real-time and temporary information, the NDS standard supports the TMC and TPEG protocols (described in Subsection 2.1.1) in the Traffic Information Building Block, but other NDS-compliant protocols can also be integrated as proprietary extensions (Navigation Data Standard (NDS) e.V., 2016).

Parking information can be included in several building blocks. Large parking lots and garages are contained in the POI Building Block with the POI standard category ‘Parking’. On-street parking, also called *lane-level parking*, is included in the Routing Building Block via a flexible attribute ‘Parking’. This building block also defines the boundaries of a larger parking area and the entrance and exit to parking lots and garages. Similarly to the speed profiles, the parking occupancy daily pattern can be stored offline to represent the typical parking occupancy level (Richter et al., 2014). Finally, the TMC and TPEG protocols can be used to transmit real-time parking information to navigation systems and smartphone applications using the Traffic Information Building Block.

2.2 Data mining

In this section, the role of data mining in the Knowledge Discovery in Databases (KDD) process is described and the main aspects of data mining are discussed. In addition, data mining approaches used in this thesis are explained in detail.

2.2.1 Knowledge Discovery in Databases (KDD) process

Measurement data from sensors can be processed to extract meaningful knowledge, such as the location of parking spaces or the prediction of parking availability addressed within this thesis. This extraction of knowledge refers to the domain of *Knowledge Discovery in Databases* (KDD). “KDD is the organized process of identifying valid, novel, useful, and understandable patterns from large and complex data sets” (Maimon and Rokach, 2005). It consists of several steps, out of which *data mining* is the core element. The steps are performed in consecutive order. However, it is a somewhat iterative process, with jumps back to one of the previous steps after gaining some additional knowledge.

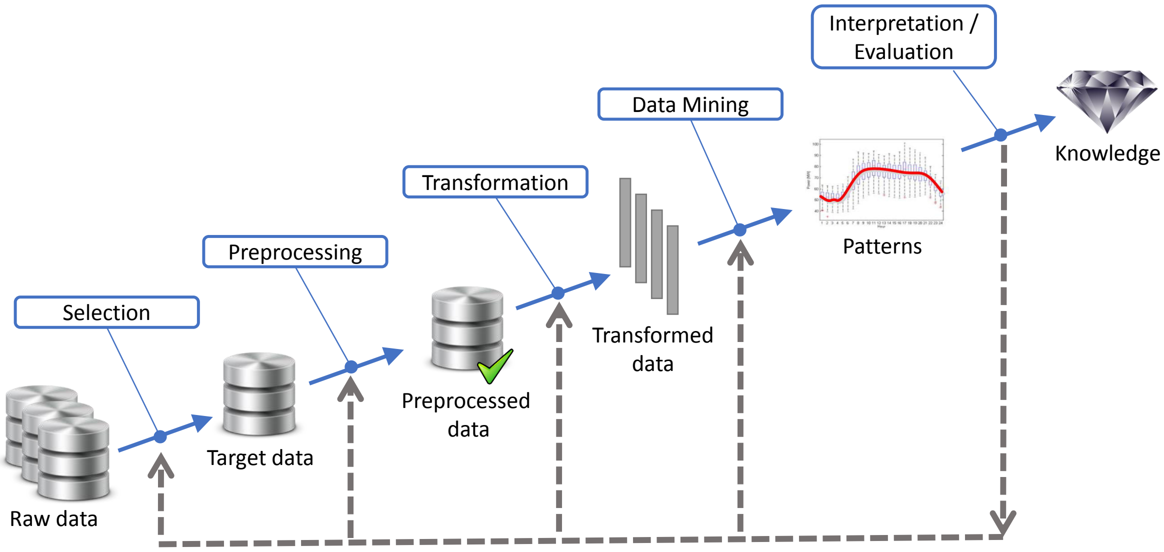


Figure 2.1: Steps of the Knowledge Discovery in Databases (KDD) process. The illustration is based on Fayyad et al. (1996).

The KDD process consists of the following steps (see also Figure 2.1) (Maimon and Rokach, 2005):

0. **Domain understanding and Goal definition:** Before processing the data, it is necessary to gain a profound understanding of the application domain. Without this understanding, it is difficult to make the right decisions about the data capture, processing and algorithms. Consequently, misinterpretations are a great risk. In addition, the goals of the discovery process need to be defined. Of course, re-definition of the goals might become necessary after some of the following steps.

1. **Selection:** Within the large amount of data often involved in the KDD process, it is crucial to select the relevant information. Certainly, more information has the potential to produce

better results, but this also comes with additional efforts and computational costs. In addition, further attributes and other data might be identified, which still need to be collected before the next steps.

2. **Data preprocessing and Cleaning:** Datasets are often heterogeneously formatted and contain irregular values. In this step, they are homogenized and cleaned of implausible, duplicate, or missing values. The data cleaning (also called *cleansing*) can be divided into three phases (Maletic and Marcus, 2005): 1) definition and determination of error types, 2) search and identification of error instances, and 3) correction of the errors identified. To identify the errors, statistical, clustering, pattern-based, and association rule methods can be applied. Several approaches such as interpolation or imputing to replace missing values also exist.

3. **Transformation:** The transformation step is needed to compute the input of the data mining algorithms. On the one hand, additional attributes can be calculated by aggregation of raw data or combination of attributes. On the other hand, the dimension of the dataset can be reduced (e.g. by a Principal Component Analysis (PCA)) to decrease the complexity of the desired model. Moreover, the dataset can be transformed to another coordinate system (e.g. polar coordinates) to facilitate the computations.

4. **Data mining:** Data mining describes algorithms that explore the data, develop the model, and discover unknown patterns. This step starts with the identification of the problem type. There are several classes of problems and related data mining tasks, such as clustering and classification. For every data mining task, there is a bunch of algorithms, each with its advantages and disadvantages. There is no superior algorithm in general, but the quality of the result is also dependent on the type of dataset (see e.g. Caruana and Niculescu-Mizil, 2006). In addition, other aspects, such as run time or interpretability of the model can be relevant criteria for the algorithm choice. Finally, the chosen algorithm is applied to the data. It is often repeated multiple times to test for different parameter settings or for different initial random values.

5. **Evaluation and Interpretation:** In the last step, the knowledge, which was extracted from the data with some data mining algorithms, needs to be evaluated and interpreted. In particular, it is compared with the goals set in the beginning and set into context with the decisions in the previous steps.

A highly related field to KDD and data mining is *machine learning*. In machine learning, the focus is on computer programs that perform certain tasks and whose performance improves with experience (Mitchell, 1997). However, there are no clear lines between those domains, since the terms are not well-defined regarding methods and type of problem (Maimon and Rokach, 2005).

2.2.2 Taxonomy of data mining methods

According to Maimon and Rokach (2005), data mining methods can be divided into verification and discovery methods (see also Figure 2.2). Verification relates to the evaluation of predefined hypotheses which are tested with statistical methods. With discovery methods, new models and patterns are identified from data. Among these, prediction methods learn a model from some training data and then apply this model to some new data. They belong to the class of *supervised learning*. In particular, classification (Subsection 2.2.3) and regression are prominent examples for this type of methods. In addition to prediction methods, there is also the class of description methods. Those methods interpret data to reveal the underlying patterns. Methods that group samples without labeled training data are also called *unsupervised*. Clustering (Subsection 2.2.4)

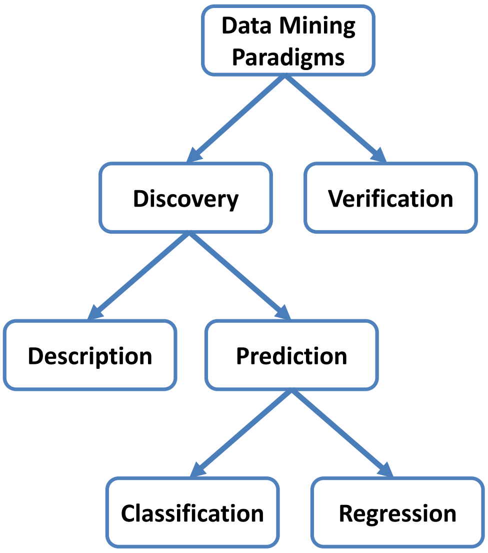


Figure 2.2: Taxonomy of data mining methods. The illustration is based on (Maimon and Rokach, 2005).

and outlier detection methods are two prominent examples of unsupervised learning approaches. In addition, there are some further types of description methods such as visualization or summarization methods. The field of time series analysis (Subsection 2.2.5) is a special case of data mining working with sequences of data points related to specific times.

In supervised learning, the choice of the right model complexity is a main challenge, referred to as the *bias-variance tradeoff* (Geurts, 2005). Bias relates to the error between a chosen model and the underlying problem (Bishop, 2006):

$$\text{bias}(\boldsymbol{x}) = E_D\left(y(\boldsymbol{x},D) - h(\boldsymbol{x})\right) \tag{2.1}$$

where $h(x)$ is the optimal prediction for an input vector x, E_D is the expectation value over all possible (training) datasets D, and $y(\boldsymbol{x},D)$ is the model output for input value $\boldsymbol{x}$ and dataset D. As the expectation value is computed over all datasets, the bias is independent from the particular training data. The more complex a model, the better the problem is described and the lower the bias. On the contrary, variance is the variability of models for different training datasets (Bishop, 2006):

$$\text{variance}(\boldsymbol{x}) = E_D\left(\left[y(\boldsymbol{x},D) - E_D(y(\boldsymbol{x},D))\right]^2\right) \tag{2.2}$$

It increases with model complexity. Thus, a trade-off needs to be found between both measures. The choice of an overly complex model that learns too many details is called *overfitting* and usually comes with a low bias and a high variance. The error of the model is low in the training data, but high for test data. An overly simple model has a high bias and a low variance. This case is called *underfitting* and comes with high errors for both training and test dataset. In addition to reducing the model complexity, the variance can also be reduced by increasing the size of the training dataset, since additional data points lead to additional constraints for the model, and thus to less variability.

2.2.3 Classification

Classification methods are an essential part of supervised learning. In classification, a classifier takes a set of training samples to learn a classification model. Each sample consists of multiple features (independent variables) and a class label (dependent variable). Features are either numeric, categorical, or ordinal. The class label assigns each sample to one class out of a set of predefined classes. The training of a classifier tries to find a model that describes the relation between features and class label. The learned model is able to predict the class of a new sample based on the features.

There are several prominent types of classifiers such as Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and decision tree algorithms. Artificial Neural Networks are inspired by the human brain and consist of a network of many neurons. These neurons combine signals together with an activation function to a new signal. Support Vector Machines transfer the feature vectors to a higher-dimensional space and search for a hyper-plane that separates the classes with the largest margin from the support vectors. The Finally, decision tree algorithms automatically build trees of relational operations following some optimization criteria.

The combination of many weak classifiers is also popular to reduce variance and the risk of overfitting (see Rokach, 2005, for more details). So-called *ensemble methods* vary the classifiers by *bagging*, *boosting*, and/or *stacking* and combine them into a final result. Bagging, also called bootstrap aggregating, draws random subsets of the training dataset with replacement. A classifier is trained for each of the subsets. The modus of the individual classification results is chosen for the final class. Random forests are bagged decision trees with random feature subset choices in every split. Boosting algorithms iteratively adapt weights of misclassified samples, in order to force the algorithm to also learn those cases. Again, the results of the iterations are combined in the end. Finally, stacking runs multiple different classification algorithms and combines their results with another classification method such as logistic regression.

For the classification tasks in this thesis, after examining different classifiers, random forests were used, because they provide robust and competitive results (Caruana and Niculescu-Mizil, 2006). In addition, solid results can already be achieved with low efforts of parameter tuning, compared with other popular classifiers (Hastie et al., 2009). Decision trees and random forests are explained in more detail in the following.

Decision trees

Following the book chapter by Rokach and Maimon (2005b), a decision tree is a classifier that recursively divides the instance space by decisions. The decisions, based on relational operations, build up a directed tree from a start node (root) over several internal nodes to the leaves, which contain the class label. Each decision at a node represents a split in the tree and a split of the instance space into two or more parts. A simple example is illustrated in Figure 2.3 for the question ‘Do I need to wear a jacket?’. In this case, the instance space has the numeric dimensions (features) temperature and rainfall and the binary dimension ‘windy’. The possible output classes are ‘yes’ and ‘no’. Starting from the root, one or multiple decisions on the features are necessary until a final decision is made. For example, if the temperature is over 20 degrees Celsius and there is no rain and no wind, then the classification result is ‘no’ (I do not need to wear my jacket).

Decision tree learning algorithms build a decision tree from a training dataset. As optimal decision tree algorithms are only feasible for small problems, because of their computational complexity, heuristics are required. There are several top-down, greedy decision tree algorithms such as ID3, C4.5 and CART. The following explanations are focused on the CART algorithm.

The learning of a decision tree consists of a growing and an optional pruning phase. In the growing phase, the locally optimal splitting is determined from the training data at every node. For that, all possible splits for every feature are evaluated based on an impurity measure. The split with the highest impurity reduction is chosen. Several impurity measures were proposed in the literature,

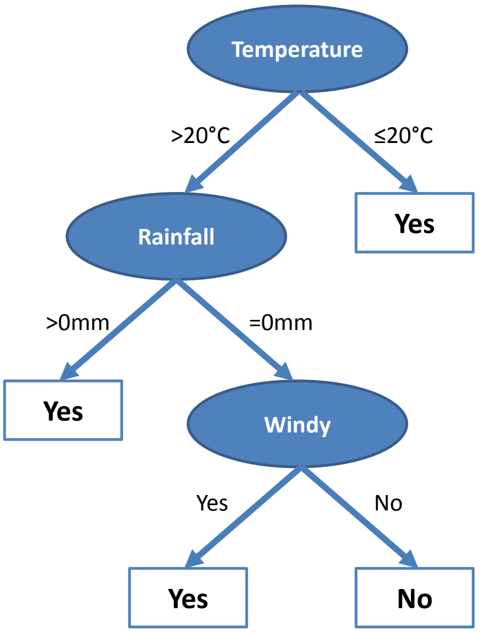


Figure 2.3: Example for a decision tree for the question 'Do I need to wear a jacket?'. The considered features are temperature, rainfall, and wind.

such as misclassification error, Gini index, and cross-entropy (Hastie et al., 2009). For example, the Gini index is defined as (Hastie et al., 2009):

$$\text{Gini} = \sum_{k=1}^{K} p_k(1 - p_k) \tag{2.3}$$

where K is the number of classes and p_k is the proportion of class k in the considered node of the decision tree:

$$p_k = \frac{N_k}{N} \tag{2.4}$$

with N_k being the number of samples that belong to class k and N the total number of samples in the node. The higher the Gini index, the higher is the purity of the class distribution.

This way, the tree is grown until there is no longer any impurity reduction for any split, or a stopping criterion is fulfilled. Stopping criteria can be a maximal tree depth, a minimal number of samples in every node, or all samples in the nodes belonging to the same class. If the tree growing is stopped too early, the decision tree underfits the problem. On the contrary, a large tree is likely to run into overfitting. Instead of searching for the right size of the tree, there are also approaches to build an overfitting decision tree and then apply some *pruning* methods. Pruning means the reduction of a decision tree by removing nodes that do not improve the generalization accuracy.

Random forests

As deeply-grown decision trees have a low bias but a high variance, random forests combine many decision trees to reduce the total variance (Breiman, 2001; Hastie et al., 2009). A majority vote determines the final class from the results of the individual decision trees. Two main techniques are used to diversify the decision trees. Firstly, a subset of samples is randomly chosen for every decision tree by a bagging approach. Secondly, the correlation between the trees is reduced by random selection of features in every step of the tree-growing process. At every split, a subset of all features is randomly chosen as split candidates. Breiman (2001) recommends a subset size of $\sqrt{f}$, with f being the number of features. Since this additional feature subsampling reduces the correlation between the trees, the variance can be further reduced, compared with bagging (Hastie et al., 2009).

According to Hastie et al. (2009), random forests have the advantage that they provide a quality assessment similar to a cross-validation already in the training phase. As not all samples are used for each tree, the unused samples, called *out-of-bag* (OOB) samples, can be evaluated. That way, the training phase can be stopped when the OOB error converges or reaches a threshold value. In addition, OOB samples help to estimate feature importance. To compute the importance of a feature, the values of this feature are randomly permuted over the OOB samples and the classification accuracy is determined for these samples. The decrease in classification accuracy by the permutation of the values is averaged over all trees and indicates the feature importance. If there is a strong accuracy decrease, the feature has a high importance.

The risk of overfitting for random forests is often discussed, for example by Hastie et al. (2009). They state that random forests do not overfit for an increasing number of trees, as the random forest estimate converges to its expectation value. However, decision trees that grow too deep might cause overfitting. Controlling the tree depth can lead to small improvements. Nevertheless, a relevant overfitting is found only seldom. Thus, the improvement must be balanced against the efforts of tuning an additional parameter.

Classification evaluation

Classification results are evaluated based on a comparison between the predicted and the actual classes. Four cases are distinguished: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). In the case of a binary classification with the classes A and B, and with A being the ‘positive’ class, a true positive is a sample for which class A is predicted and is the correct class. A false positive is a sample for which class A is predicted, but class B would be correct. A true negative is a sample for which class B is correctly predicted. A false negative is a sample for which class B is predicted, but class A would be correct. The counts of these cases are usually presented in a confusion matrix (see Table 2.1).

Table 2.1: Example for a confusion matrix.

		Actual Class	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
Class	Negative	False Negative	True Negative

Based on these counts, additional measures can be calculated:

- **False positive rate:**

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \tag{2.5}$$

- **True positive rate / Recall:**

$$\mathrm{TPR} = \mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.6}$$

- **Precision:**

$$\mathrm{Pre} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2.7}$$

- **Accuracy:**

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{2.8}$$

- **F-measure** (also called F_1 score):

$$\mathrm{F}_1 = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{2.9}$$

If a binary classifier returns a classification score in addition to the class label, the decision threshold for the positive class can be varied. That way, different combinations of true positive rate and false positive rate are obtained. A curve with all possible combinations of TPR and FPR is called a Receiver Operating Characteristic (ROC) curve (Davis and Goadrich, 2006). Examples for ROC curves are shown in Figure 2.4. If both classes have equal numbers of samples, a random classifier achieves a straight line from (0,0) to (1,1) (dotted red line in the figure). A perfect classifier already reaches TPR=1 at FPR=0 (dashed yellow line). Typical ROC curves are monotonically increasing, with the curve placed between the random and the perfect classifier. The area under the curve (AUC) is a popular quality measure to evaluate the ROC curve. The larger the area under the curve, the better the classification result. It can be interpreted as the probability that a randomly chosen positive sample gets a higher classification score than a randomly chosen negative sample.

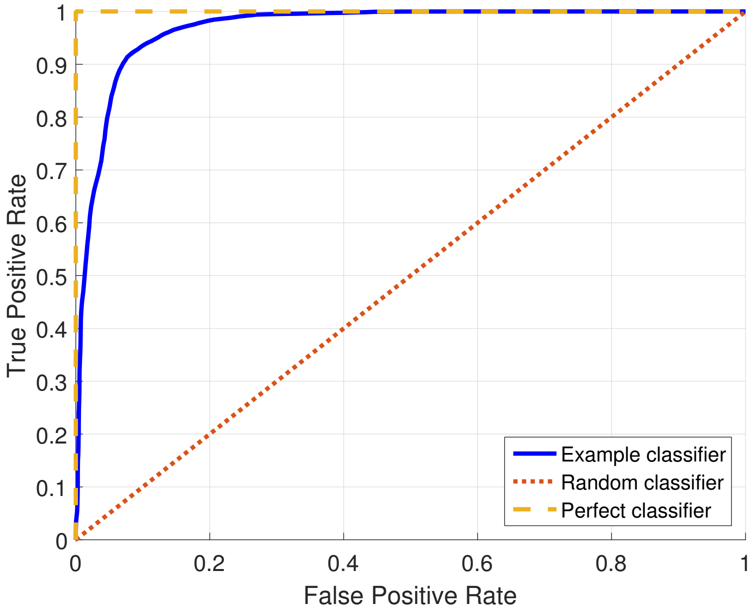


Figure 2.4: ROC curves for an example classifier, a random classifier, and a perfect classifier.

For highly imbalanced datasets, where, for example, the negative class has far more samples than the positive class, quality measures based on TN such as accuracy and FPR do not provide accurate quality information about the classification result (He and Garcia, 2009). It is recommended instead to use precision, recall, and the resulting Precision-Recall (PR) curves. For these curves, the area under the Precision-Recall curve (AUPR) can also be used as quality measure. Each point of a PR curve has a corresponding point on the ROC curve. When comparing two classifiers, the one with the superior ROC curve also has the better PR curve. However, optimizing the area under the ROC curve does not necessarily optimize the area under the PR curve (Davis and Goadrich, 2006).

Following the explanations by Bishop (2006), in order to measure the generalization performance, a dataset is divided into two subsets: training and test data. The training data is used to learn the classifier. Evaluating the classifier on the test data gives the generalization quality of the approach. If the test dataset is evaluated many times to tune the parameters of the classifier, it is recommended to divide the dataset into three subsets (training, validation, and test dataset) to avoid overfitting of the test data. In this case, the classifier is trained with the training dataset for different model parameters and evaluated with the validation dataset. The parameters with the best quality on the validation dataset are then used on the unseen test dataset to determine the generalization quality.

For small datasets, it is inadvisable to use only a part of the dataset for training. The test dataset should also be sufficiently large for a representative result. To solve this dilemma, the dataset can be divided into k subsets. $k-1$ subsets are used for training (or $k-2$ subsets if a validation subset is necessary) and the remaining subset is used for testing. This procedure is repeated k times so that each subset was the test dataset in one iteration. The final result is then aggregated over all iterations. This approach is called *cross-validation.* For very small datasets, each subset could contain just one sample, called *leave-one-out cross-validation.* A drawback of this approach is the strongly increased computation time.

2.2.4 Clustering

Clustering is an unsupervised learning task with the goal of identifying groups (clusters) of similar samples in the dataset (Bishop, 2006). According to the explanations by Rokach and Maimon (2005a), the similarity of sample pairs is described by a similarity measure such as cosine measure or Pearson correlation, or by a distance measure (dissimilarity) such as the Euclidean distance or the more general Minkowski metric. Valid similarity (distance) measures $s(x_i,x_j)$ need to be symmetric and reach the maximum (minimum) value for identical samples $x_i = x_j$. Clustering methods are often based on hierarchical and partitional approaches, but there are also density-based, model-based and grid-based approaches.

In the following, the K-means clustering (partitional clustering) and hierarchical clustering are explained in more detail as they are used in the thesis. In addition, two cluster validation measures are explained, which are used to describe the quality of the clustering result and to determine the suitable number of clusters.

K-means clustering

K-means is a basic and well-known partitional clustering algorithm. The number of clusters K must be specified in advance. All samples are assigned to one of the K clusters with the goal to minimize the cost function (Bishop, 2006):

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| x_n - \mu_k \right\|^2 \tag{2.10}$$

where J is the cost value, N the number of samples x_n in the dataset, μ_k are the prototypes of the K clusters, $\|x_n - \mu_k\|$ is the Euclidean distance between sample x_n and prototype μ_k, and r_{nk} are binary values for the assignment of a sample x_n to a cluster k. That means $r_{nk} = 1$ if sample x_n is assigned to cluster k and $r_{nj} = 0$ for all other values of j $(j \neq k)$. Thus, the cost function is minimized if the prototypes μ_k are chosen such that the distances $\|x_n - \mu_k\|$ are smallest for all assigned samples x_n.

First, initial values are defined for the prototypes μ_k of the K clusters. Then, the minimization of the cost function happens iteratively in two steps (Bishop, 2006) (see Figure 2.5 for an example):

1. **Assignment of samples to cluster prototypes:** Each sample has to be assigned to one of the clusters so that the cost function is minimized for the given prototypes. This is the case if every sample is assigned to the nearest prototype. The assignment values r_{nk} are updated accordingly.
2. **Re-computation of cluster prototypes:** The cluster prototypes are computed based on the new sample assignments. The cost function is minimized if the prototypes are placed in the center of the assigned samples, corresponding to their mean value:

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}} \tag{2.11}$$

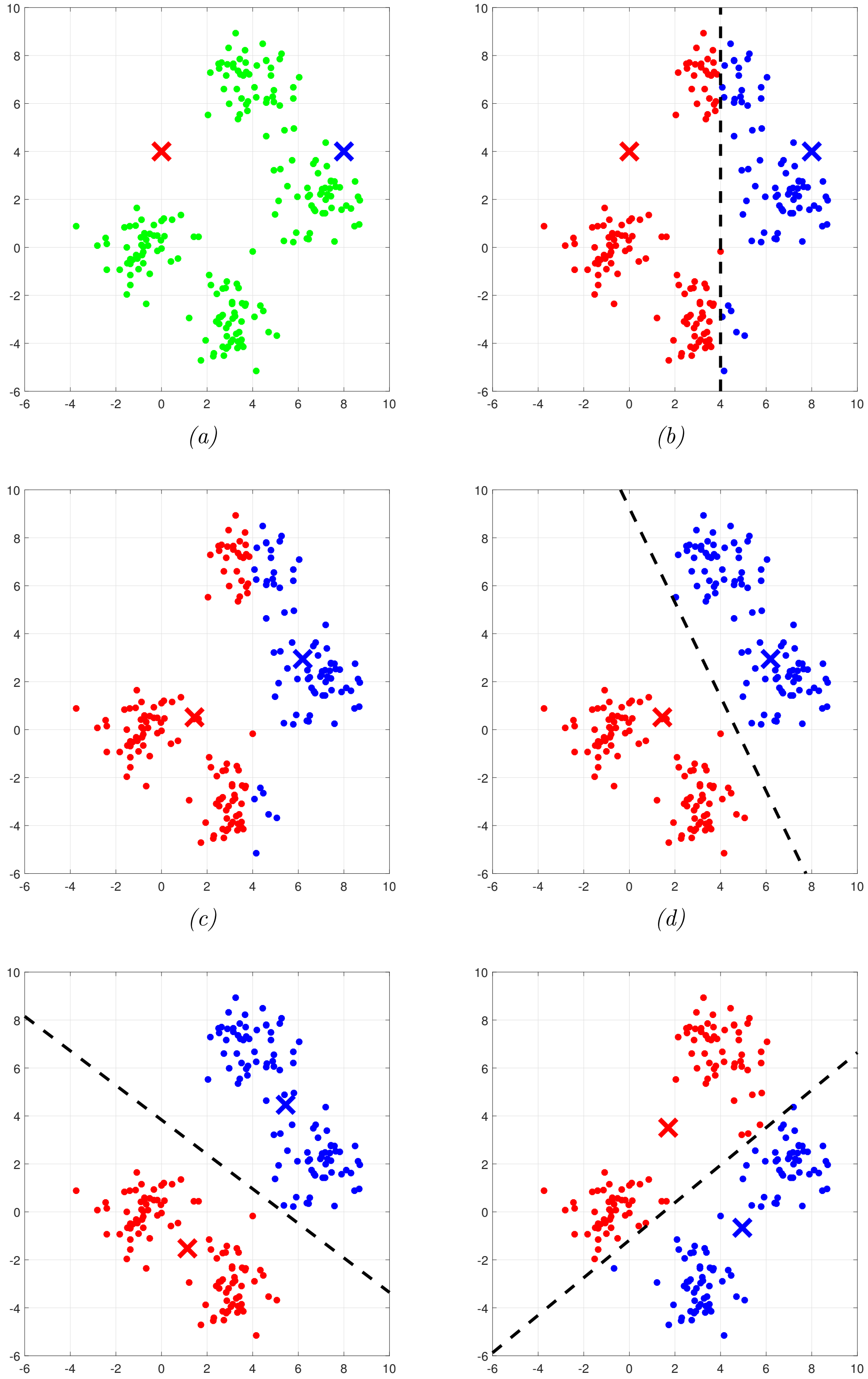
Both steps are iterated, either until a maximal number of iterations is reached or the algorithm converges and there is no longer any change in an iteration. Since every iteration reduces the cost function, the algorithm converges to a minimum. However, it might end up in a local instead of the global minimum, depending on the start values of the prototypes (see Figures 2.5e and 2.5f). Therefore, repeating the clustering multiple times for different start values is recommended. In addition, instead of choosing the start values completely randomly, they can be selected as the centers of random subsets of the dataset. To avoid a strong influence of outliers on the prototypes by the mean calculation, the K-medoids algorithm was proposed, which allows other distance functions and thus the computation of the medoid instead of the mean.

Hierarchical clustering

Following the book chapter by Rokach and Maimon (2005a), in hierarchical clustering, either small clusters are recursively merged to larger clusters (agglomerative), or large clusters are recursively divided into smaller clusters (divisive). The agglomerative hierarchical clustering starts with all samples being their own clusters. In every merge step, the clusters with the smallest cluster distance are combined into a new cluster. The merging is repeated until the clustering requirements are reached, or only one large cluster remains. Analogously, the divisive hierarchical clustering starts with one cluster that contains all samples. Clusters are divided into new clusters based on the cluster distance measure. The recursive splitting or merging leads to a tree of clusters called *dendrogram*. The dendrogram shows the hierarchical grouping of clusters and the similarity level of the groupings. An example is shown in Figure 2.6. Cutting the dendrogram at any desired similarity level or number of clusters leads to the clustering result.

Multiple cluster distance measures exist that describe the dissimilarity of two clusters (Rokach and Maimon, 2005a):

- **Single-link clustering:** The distance between two clusters corresponds to the **minimal** distance between any member of one cluster and any member of the other cluster. Single-link clustering is most versatile in the shape of the clusters. However, a few samples can mistakenly link two clusters (‘chaining effect’).
- **Complete-link clustering:** The distance between two clusters corresponds to the **maximal** distance between any member of one cluster and any member of the other cluster. Complete-link clustering leads to more compact results, but is less versatile.



(a) (b) (c) (d) (e) (f)

Figure 2.5: Illustration of the steps in the k-means algorithm: (a) distribution of samples (green dots) and initial prototypes (red/blue crosses) before clustering, (b) first step: assignment of samples to cluster prototypes, (c) second step: re-computation of cluster prototypes, (d) second iteration of first step, (e) final clustering result, (f) alternative result if initial prototypes are chosen differently and the algorithm converges to a local minimum. The illustration is based on Bishop (2006).

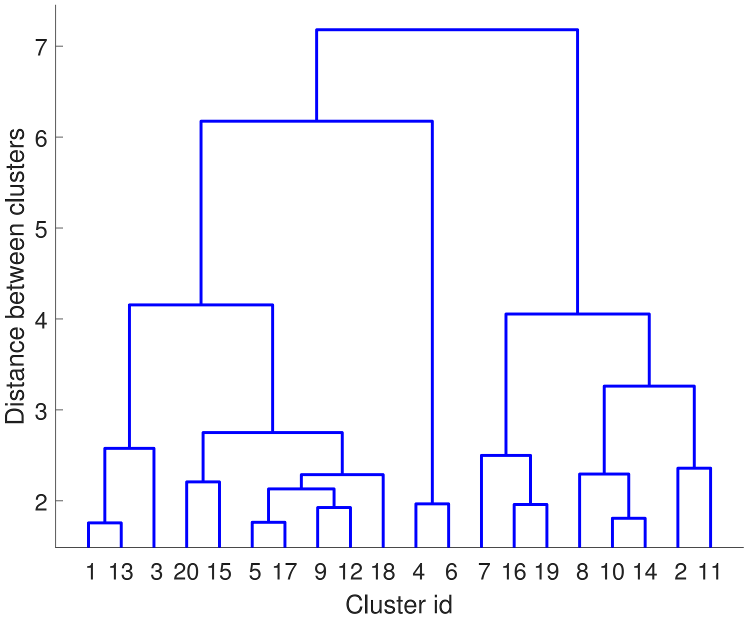


Figure 2.6: Example for a dendrogram generated with hierarchical clustering.

- **Average-link clustering:** The distance between two clusters corresponds to the **average** distance between any member of one cluster and any member of the other cluster. Average-link clustering may lead to wrong cluster splits for elongated clusters.

Hierarchical clustering algorithms have the advantage of generating many nested cluster assignments that can be represented in the dendrogram. However, their time complexity is at least quadratic with the number of hierarchy steps (which increase non-linearly with the number of samples).

Cluster validation techniques

Several measures were defined to indicate the quality of a clustering result and the most appropriate number of clusters. In this thesis, the Silhouette value and the Davies-Bouldin-Index were used.

The **Silhouette value** is a measure for one sample X_i belonging to a cluster A in a clustering result (Rousseeuw, 1987). It measures the difference between the average distance (dissimilarity) $a(i)$ of the sample and all other members of cluster A and the average distance $b(i)$ of the sample to the members of the most similar cluster, normalized by the maximum of both values $a(i)$ and $b(i)$:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i),b(i))} \tag{2.12}$$

The average distance $a(i)$ of the sample from all other cluster members is computed as:

$$a(i) = \frac{1}{N_A - 1} \sum_{j=1,i\neq j}^{N_A} d(X_i,X_j) \tag{2.13}$$

where N_A is the number of samples in cluster A, X_j are the members of cluster A, and $d(X_i,X_j)$ is the distance between the samples X_i and X_j. If the sample i is the only member of its cluster, $a(i)$ cannot be computed. Thus, $s(i)$ is set equal to zero in this case.

The most similar cluster is determined by the average dissimilarity of the sample i from all members of a potential cluster C:

$$b(i) = \min_{C \neq A} d(i,C) = \min_{C \neq A} \frac{1}{N_C} \sum_{j=1}^{N_C} d(X_i,X_j) \tag{2.14}$$

To evaluate the clustering result, the average of the Silhouette values for all samples is computed. The higher the final value, the better the clustering result.

The **Davies-Bouldin-Index** $\bar{R}$ describes the average similarity of all clusters with their most similar clusters (Davies and Bouldin, 1979):

$$\bar{R} = \frac{1}{N} \sum_{i=1}^{N} R_i \tag{2.15}$$

where N is the number of clusters and R_i is the similarity value of a cluster i with its most similar cluster:

$$R_i = \max_{j \neq i} R_{i,j} \tag{2.16}$$

The similarity value $R_{i,j}$ of two clusters i and j is the ratio between the sum of dispersions $S_i + S_j$ and the Minkowski distance of the cluster centroids $M_{i,j}$.

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}} \tag{2.17}$$

The dispersion S_i describes the deviation of the cluster samples X_j from their cluster centroid C_i. $\|X_j - C_i\|$ is the Euclidean distance, N_i is the number of samples in cluster i and q is the order of the momentum (q=1 simply corresponds to the average Euclidean distance, q=2 to the standard deviation of the distance):

$$S_i = \left(\frac{1}{N_i} \sum_{j=1}^{N_i} \|X_j - C_i\|^q\right)^{1/q} \tag{2.18}$$

Finally, the Minkowski distance is

$$M_{i,j} = \left(\sum_{k=1}^{D} |c_{k,i} - c_{k,j}|^p\right)^{1/p} \tag{2.19}$$

where D is the number of dimensions in the dataset, $c_{k,i}$ is the k-th element of the cluster centroid vector C_i, and p is the order of the Minkowski distance. $M_{i,j}$ describes the distance of the clusters, while the dispersions $S_i + S_j$ measure the compactness of the clusters themselves. A smaller value of the Davies-Bouldin Index $\bar{R}$ means a better clustering result, since the dispersions of the clusters are small, when compared with the Minkowski distance between the clusters.

2.2.5 Time series analysis

A sequence of observations over time is called *time series* (Makridakis et al., 2008). In this thesis, mainly parking information such as the number or rate of occupied parking spaces over time is considered as time series. Moreover, trajectories can be seen as time series with measurements of the coordinates at subsequent time instants.

Autocorrelation function

To further explore a time series and to identify the periodicity of the time series, the autocorrelation function can be used. The autocorrelation is based on the covariance, which describes the similarity between two time series X and Y (Makridakis et al., 2008):

$$\mathrm{Cov}(X,Y) = \frac{1}{N-1}\sum_{t=1}^{N}(X_t - \bar{X})(Y_t - \bar{Y}) \tag{2.20}$$

where N is the length of the time series, X_t and Y_t the values of the time series at time step t, and $\bar{X}$ and $\bar{Y}$ are the mean values of the time series. The (Pearson) correlation is the covariance value normalized by the standard deviations σ_X and σ_Y.

$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{t=1}^{N}(X_t - \bar{X})(Y_t - \bar{Y})}{\sqrt{\sum_{t=1}^{N}(X_t - \bar{X})^2}\sqrt{\sum_{t=1}^{N}(Y_t - \bar{Y})^2}} \tag{2.21}$$

The *autocorrelation function*(ACF) is the (Pearson) correlation of a time series Y and its lagged time series with lag value k:

$$\rho(Y,k) = \frac{\sum_{t=k+1}^{N}(Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^{N}(Y_t - \bar{Y})^2} \tag{2.22}$$

It describes the similarity between the lagged time series and the time series itself. Thus, it is a good indicator for periodic patterns. The plot of the autocorrelation value against the lag k is called *correlogram.*

Fourier analysis

An alternative to identify the periodicity of a time series is the *Fourier analysis.* For a time series of discrete time steps, the *Discrete Fourier Transform* transfers the series Y with N time steps from time to frequency space (Shumway and Stoffer, 2010):

$$\begin{aligned} d(\omega_j) &= N^{-1/2}\sum_{t=1}^{N} Y_t \mathrm{e}^{-2\pi i \omega_j t} \\ &= N^{-1/2}\sum_{t=1}^{N} Y_t(\cos(2\pi\omega_j t) - i\sin(2\pi\omega_j t)) \end{aligned} \tag{2.23}$$

where $d(\omega_j)$ are the *Fourier coefficients*, $\omega_j = j/N$ the *Fourier frequencies*, and i is the imaginary unit. The Fourier coefficients can be transformed back to time space similarly:

$$\begin{aligned} Y_t &= N^{-1/2}\sum_{j=0}^{N-1} d(\omega_j)\mathrm{e}^{2\pi i \omega_j t} \\ &= N^{-1/2}\sum_{j=0}^{N-1} d(\omega_j)(\cos(2\pi\omega_j t) + i\sin(2\pi\omega_j t)) \end{aligned} \tag{2.24}$$

This means that every time series can be decomposed into a linear combination of sine and cosine functions. The Fourier coefficients are complex numbers. Thus, the magnitude of the frequencies is described by the *periodogram* as an estimate of the spectral density:

$$I(\omega_j) = |d(\omega_j)|^2 \tag{2.25}$$

The periodogram reveals the most prominent frequencies in the time series, which can be used to determine the periodicity for the time series decomposition. In addition, the Fourier transform can also be used for dimensionality reduction, by keeping only the coefficients of the most relevant frequencies.

3 Related work about mobile crowdsensing of on-street parking spaces

Fundamental and recent research about on-street parking technologies (Section 3.1) and mobile crowdsensing (Section 3.2) is presented in this chapter. In addition, based on the related work presented, the research gaps addressed in this thesis are described (Section 3.3).

3.1 On-street parking

Parking in urban areas can be divided into off-street and on-street parking. Off-street parking relates to parking lots and parking garages, often with hundreds of parking spaces. Since access to them is usually regulated, the occupancy can be determined easily, by counting the incoming and outgoing vehicles. In this work, the focus lies on the more challenging case of on-street parking, such as parking lanes and parking stalls directly accessible from the road. The detection of parking occupancy with static and mobile parking sensors is described in Subsection 3.1.1. When parking occupancy information is detected for at least some of the parking spaces, the overall parking occupancy in a road segment or an area can be estimated and the future occupancy predicted (Subsection 3.1.2). Finally, field and simulation studies about parking search, as well as the assistance of drivers on the search by parking guidance systems is described in Subsection 3.1.3.

3.1.1 Parking occupancy detection

For most smart parking applications, the current parking situation needs to be determined first. Approaches to detect parking occupancy can be divided by the usage of either static or mobile parking sensors:

Static parking sensors

Static parking sensors are used to continuously observe the occupancy of parking spots. Magnetometers, cameras, and radar sensors are the most prominent sensor types considered for this task.

Magnetometers are installed in the pavement of every parking spot. They measure the magnetic field and detect differences when a car is parked above the sensor. Detections are communicated to repeaters and gateways via an ultra-low power mesh network (San Francisco Municipal Transportation Agency, 2014). The gateways collect sensor information from all nearby sensors and send the data to a central instance via mobile communications. This technology is already used in busy roads in several cities around the world, such as San Francisco, Moscow, and London (Lin et al., 2017). However, a main disadvantage of static sensors is that deployment and maintenance are expensive for large scale deployment (Xu et al., 2013). For example, San Francisco Municipal Transportation Agency (SFMTA) paid more than five million dollars for installation and maintenance of about 8,000 parking spaces with sensors (there are more than 270,000 on-street parking spaces in San Francisco), but stopped running the sensors after the batteries were dead and the end of the SFpark project was reached (San Francisco Municipal Transportation Agency, 2014). More details about the SFpark project in San Francisco will be described in Section 6.1, as data from 5,000 of these sensors (pilot area) are used in parts of this thesis.

To reduce the sensor costs, Evenepoel et al. (2014) proposed an approach to thin out the sensor coverage by reducing the number of equipped parking spaces and to interpolate the parking saturation. In their case study, they built a detailed cost model for a parking sensor system and a probabilistic model to assess the reliability of the approach. They concluded that costs could be reduced up to a factor of 30 while retaining sufficient performance. However, the performance measure was based only on an estimation of parking-related congestion, but did not consider further effects such as reduced search time. In addition, a strong assumption was made that the occupancy probability is the same for all parking spaces in the city. This assumption makes an interpolation much easier than in a real-world heterogeneous environment.

An alternative for reducing the number of sensors is the installation of sensors such as cameras and radar sensors on elevated spots. They are installed on walls, lamp posts, or traffic light poles, to survey several parking spaces of the parking lane. Cameras detect vehicles from the video images using computer vision techniques (Cleverciti, 2017; Fabian, 2008). However, a lack of public acceptance for cameras at every corner could be a main problem, as they might also be used for other surveillance tasks. Thus, radar sensors seem to be a good alternative. In 2015, Siemens started a test field in Berlin, Germany, where they equipped a road segment of 250 meters with radar sensors (Siemens, 2017).

A more detailed survey of sensors for parking occupancy detection is provided by Lin et al. (2017).

Mobile parking sensors

Mobile sensors can be a meaningful alternative. Mobile sensing means that the sensor moves through the streets and can thus cover a larger area. In the context of parking, sensors in vehicles and smartphones are mainly investigated. In addition, the use of unmanned aerial vehicles (UAVs) has also been proposed (Li et al., 2017).

Sensors in vehicles can detect the parking occupancy both by observing parking-related events of the vehicle itself and by scanning the parking lanes while passing. For the detection of the parking lane occupancy during the trip, ultrasonic, radar, camera, and LiDAR (Light Detection And Ranging) sensors have been proposed. Ultrasonic sensors are the cheapest. If they are facing the side of the vehicle, they continuously scan the distance to the next object (Mathur et al., 2010; Park et al., 2008; Robert Bosch GmbH, 2016). Thus, gaps in the parking lane can be detected. However, the interpretation of the object reflections and also the frequency of the ultrasound signals are limited. To increase the resolution of the scanning, multiple ultrasonic sensors can be combined. Alternatively, radar sensors can also be used to create a detailed occupancy grid of the surroundings (Dubé et al., 2014). Video images from in-vehicle cameras can also be exploited to directly detect parked vehicles (Sivaraman and Trivedi, 2013). Both Ono et al. (2002) and Thornton et al. (2014) used 2D LiDAR scanners to conduct parking surveys. The scanners were installed perpendicular to the driving direction. Ono et al. (2002) identified parked vehicles by analysis of the coverage of the adjacent lane, assuming that the vehicle was driving next to the parking lane. Thornton et al. (2014) identified ground and curb in the point clouds. If the curb was occluded, they inferred that the parking lane was occupied. To avoid misinterpretations, they applied an occlusion reasoning approach.

Most smartphone parking availability applications aim to detect parking and leaving events (Stenneth et al., 2012; Ma et al., 2014). These events are detected either directly from GPS and accelerometer signals or indirectly from detections of travel mode changes or connection buildups between vehicle and smartphone (e.g. via Bluetooth). Moreover, parking payment applications can be exploited to extract parking and leaving events (Lin et al., 2017). From these sources, the start and end of the trip can be extracted. Usually, the start and end points correspond to occupying and leaving a parking space. In addition, there are also approaches to detect parking search by the identification of characteristic movement patterns in the GPS traces (Montini et al., 2012; Hampshire et al., 2016; Liu et al., 2017). Characteristic movement patterns can be, for example, reduced velocity, detours from the shortest path, and/or driving in circles. Note that these parking-related events could also be detected by vehicle sensors. In addition, there are smartphone applications considered that are based on active interaction with the user to answer questions about the parking availability (Chen et al., 2012).

The detection of parking-related information by many vehicles or smartphones leads to parking crowdsensing (Chen et al., 2012; Margreiter and Mayer, 2015; Robert Bosch GmbH, 2016). Vehicles and smartphones communicate their information to a central server instance (see also Figure 1.1 on page 14). The server collects the data from all sources and aggregates them to a complete map of parking availability. This map can then be transferred back to other users searching for an empty parking space.

Many of the approaches with mobile sensors only detect gaps in parking lanes, but miss the information about whether parking in this gap is illegal (e.g. at a garage entrance or close to a crosswalk). Thus, maps containing locations of parking spaces are necessary. Since many cities do not have parking maps in this level of detail, approaches were proposed to learn the legality from parking occupancy detections at different time instants (Coric and Gruteser, 2013; Ge et al., 2013). The approach of Ge et al. (2013) only compares the number of vehicles detected at a location to a threshold. Coric and Gruteser (2013) proposed a more elaborate method. They split the road into small segments of one meter and computed a weighted occupancy rate at each segment. The weights were based on the average occupancy level of the road. The weighted occupancy rate was compared with a threshold. Afterwards, a smoothing was applied to remove sections of legal or illegal parking smaller than a threshold. Both thresholds were manually defined without any calibration.

3.1.2 Parking availability estimation and prediction

Based on the collection of parking-related information, the parking availability can be estimated and predicted. In this context, estimation means that the current parking availability is estimated from incomplete current observations and historic information. The prediction guesses the parking availability in the near future, for both incomplete and complete observation scenarios.

Few studies exist about the parking availability estimation from crowdsensed, and thus irregularly sampled, datasets. In a simulation environment, Farkas and Lendák (2015) evaluated a parking search scenario where drivers used an application that detected empty parking spaces with a smartphone app. Only the empty spots detected were recommended to other drivers, but the availability was not estimated for the remaining parking spaces. Xu et al. (2013) compared approaches using a Kalman filter, weighted average, and historical availability average to estimate the parking availability rate in a street segment based on vehicle leaving/parking detections, such as from mobile apps. They considered different user rates and simulated error rates of the sensors. Evaluations with real-world data from the SFpark project in San Francisco showed that the Kalman filter and weighted average approaches achieved similar results, both better than the historical availability average.

Much of the literature addresses the prediction of parking availability (Lin et al., 2017), especially for the case of parking lots, as parking data is easier to obtain at the lot entrances than for on-street parking. Caliskan et al. (2007) used a Markov model to predict the parking occupancy based on current arrival and leaving rates at the parking lot. In their scenario, vehicles exchanged this information in an ad hoc network and predicted parking availability for their expected time of arrival. Wu et al. (2014) proposed the use of real-time data only for short-term prediction in combination with historical data. For longer terms, the average day trend from historical data was used. An alternative is the clustering of parking occupancy daily patterns to find similar patterns and the prediction using a k-nearest-neighbor classifier (Tamrazian et al., 2015).

For on-street parking, the prediction is more challenging, because the parking occupancy rate of a street segment with some parking spaces significantly changes for every parking/leaving event, compared with parking lots with hundreds of parking spaces. In a patent, Google claims an approach to estimate and predict the population density from smartphone data (Lookingbill, 2013). In a second step, they learn the relationship between population and parking occupancy. Artificial neural networks are also popular for prediction, either to predict the time series (Vlahogianni et al., 2016) or to predict from various attributes such as weather, traffic, events, and time of day (Zheng et al., 2015; Pflügler et al., 2016). Rajabioun and Ioannou (2015) proposed a multivariate autoregressive model for parking prediction in systems with static parking sensors, considering both spatial and temporal correlations. To reduce the amount of data for the daily patterns of parking occupancy per road segment, Richter et al. (2014) evaluated different levels of spatial and temporal clustering of the daily pattern.

3.1.3 Parking search and guidance

The search for an empty parking space is investigated from multiple perspectives. Bonsall and Palmer (2004) used a driving simulator to study the route decisions of drivers during the parking search. Several parking simulation platforms were developed to study parking search characteristics (Benenson et al., 2008; Horni et al., 2013; Bessghaier et al., 2012). In a real-world environment, Hampshire et al. (2016) analyzed the parking search behavior of drivers based on their GPS trajectories and in-car video recordings. They found that the combination of reduced velocity and head rotations accurately indicated the parking search phase.

As reported in the introduction, the observed average duration of parking search varies widely (e.g. 36 seconds (Van Ommeren et al., 2012) and 8.1 minutes (Shoup, 2006)). Moreover, the measurement of parking search length and duration is often biased, since parking studies are primarily conducted in problematic areas (Shoup, 2006). Millard-Ball et al. (2014) evaluated multiple years of parking data from SFpark in San Francisco. Based on the occupancy data, they simulated the parking search. As a result, they found that parking search is less severe during metered hours than outside metered hours, and might be more severe in non-metered parking lanes. Thus, the sensors in SFpark capture only parts of the city's parking situation.

The first systems that inform drivers about the location and the current availability count in parking lots via variable message signs date back to the 1970s (Axhausen et al., 1994). Those systems are called *Parking Guidance and Information* (PGI) systems. Studies claim that PGIs lead to a significantly increased probability of finding available parking spaces (Teodorović and Lučić, 2006) and reduced search time (Axhausen et al., 1994). In parking garages, the search time could be more than halved, if proper stall availability information was available (Caicedo et al., 2006). Modern PGI approaches can be integrated into navigation systems and provide personal search route recommendations. Multiple approaches have been proposed for the computation of an optimal search route (Guo and Wolfson, 2016; Jossé et al., 2013), for an optimized parking space assignment (Ayala et al., 2011), and for parking reservation methods (Kotb et al., 2017). The restriction of information dissemination was also discussed, to avoid all drivers rushing to the same parking space (Delot et al., 2013).

3.2 Mobile crowdsensing

Mobile crowdsensing (also called *collaborative sensing* by Ilarri et al. (2014)) is a modern approach to collecting large amounts of data from mobile devices. Guo et al. (2014) defines it as “a new sensing paradigm that empowers ordinary citizens to contribute data sensed or generated from their mobile devices, aggregates and fuses the data in the cloud for crowd intelligence extraction and people-centric service delivery”. Crowdsensing applications can be divided into two classes: participatory and opportunistic sensing (Ganti et al., 2011). Participatory sensing means that an action is required by the user to generate the data (e.g. marking the location of an empty parking space). In opportunistic sensing, no or low efforts by the users are needed (e.g. recording of GPS trajectories). In the following, crowdsensing approaches in transportation and parking in particular are described.

3.2.1 Mobile crowdsensing in transportation

In the field of transportation, mainly vehicles and smartphones are considered as mobile devices for crowdsensing. The most important sensors in traffic scenarios are GPS sensors and accelerometers (Lendák, 2016). According to Ilarri et al. (2014), those sensors fall into the category of *physical sensors*. Moreover, there are *virtual sensors* (also called *software sensors*; they provide an aggregation of multiple sensors), *social sensors* (data extracted from social media such as Foursquare[1] or Twitter[2]), and *human sensors* (explicit information provision by humans, e.g. as volunteered geographic information (VGI)) (Ilarri et al., 2014). To increase the coverage of vehicular crowdsensing, Masutani (2015) proposed a reservation-based approach. If a vehicle plans to drive along a road segment, it ‘reserves’ this road segment at the predicted arrival time. The reservation leads to an increased edge cost in the routing of other cooperative vehicles and thus to alternative route choices.

Ilarri et al. (2014) identified three main domains of crowdsensing in the field of transportation: traffic information, trajectories, and parking. The most widely used traffic information is travel time. The current speed and location of vehicles is transferred via cellular communication and collected in a central instance as *floating car data* (FCD). The aggregation of floating car data provides real-time traffic information that can help drivers to avoid busy roads and traffic jams. Several providers, such as TomTom[3], Google Maps[4], and Inrix[5] achieve accurate real-time traffic information from a very dense network of probes. Another prominent example for traffic crowdsensing on smartphones is Waze[6]. This application encourages users to press buttons to report traffic jams, accidents, speed traps, and more. Those notifications are directly provided to all other users in the vicinity. From trajectories collected, various types of knowledge can be extracted. For example, road network maps can be reconstructed (Kuntzsch et al., 2016; Ahmed et al., 2015), popular routes can be identified, and drivers with similar routes can be pooled for real-time ride-sharing (Ilarri et al., 2014).

---

[1] https://foursquare.com/

[2] https://twitter.com/

[3] https://www.tomtom.com/

[4] https://maps.google.com/

[5] http://inrix.com/

[6] https://www.waze.com/

3.2.2 Mobile crowdsensing for parking

For parking crowdsensing, vehicular and smartphone sensors can be used in various ways, as described in Section 3.1.1. In a simplified evaluation of taxi trajectories based on the coverage of small rectangular grid cells, Mathur et al. (2010) found that 536 taxis led to a mean inter-sampling interval of 10 minutes for 80 % of the area (25 minutes for 85 %) in downtown San Francisco. They assumed that this taxi activity would be enough to cover this area adequately. The smartphone application OpenSpot from Google used a participatory approach to guide drivers to empty parking spaces that were tagged before by other users (Sherwin, 2011). However, this service was cancelled in 2012 due to a too small user base and hence poor data quality. Lendák and Farkas (2016) investigated potential reasons for the failure. In their study, they assumed an application that provides information on empty parking spaces either right after the user leaves, or with drive-by detection. Only the empty spots detected are recommended to other drivers. They discovered that drive-by sensing is necessary to achieve good system quality, but a pure participatory approach such as in OpenSpot was not sufficient. However, the approaches were only tested in a rather simple simulation scenario.

The magnitude of search time reductions by parking crowdsensing applications is controversial. Rybarsch et al. (2017) found that cooperation on the search, by exchange of the planned routes and of the past trajectories with unsuccessful search, can lead to a significant search time reduction of up to 30 %. In a simulation study with sensing vehicles by Kokolaki et al. (2012), the benefit depended on the distribution of parking availability. In a scenario with a parking hotspot, a centralized system with a reservation approach achieved the best search time, followed by a decentralized information exchange between the vehicles without reservation and the baseline without crowdsensing. However, in a scenario with uniformly distributed high parking demand, the centralized system performed even worse than the baseline. The decentralized approach still led to reduced search times. In two further simulation studies, the exchange of parking and leaving information did not significantly reduce parking search time in rather homogeneous parking settings (Tasseron et al., 2015, 2016). However, they did not consider a reservation protocol, and as a result, several vehicles tried to catch the same empty spot. Moreover, they discussed the possibility that a more realistic environment (e.g. different capacities of the parking lanes) might lead to reduced search times. Still, a search time reduction was observed in the case of static sensors instead of crowdsensing. In an extension of their simulation, they found that a decentralized reservation system for parking spaces with sensors further reduced the search times for intelligent vehicles (Tasseron and Martens, 2017). However, normal vehicles suffered in the same extent, so that the search time averaged over all vehicles was not reduced.

3.3 Research gaps addressed in this thesis

Collection of parking occupancy information with a LiDAR-equipped vehicle

Using LiDAR-equipped vehicles to detect parking gaps or parked vehicles is seldom studied in the literature. The approach by Thornton et al. (2014) is applicable only in the presence of curbs. Moreover, they needed some prior knowledge about the location of the parking lanes (like also Ono et al., 2002). However, a more general approach for the extraction of parking information was missing. Thus, a new approach is presented in this thesis, based on a segmentation of 3D point clouds and a classification of the resulting segments. The approach requires neither the presence of curbs nor prior explicit knowledge of vehicle shapes or parking lane locations. Thus, vehicles parked illegally outside of parking lanes can also be detected.

Learning parking legality maps from crowdsensed occupancy information

Both approaches from the literature (Ge et al., 2013; Coric and Gruteser, 2013) are based on manually defined parameters which might vary significantly for different investigation areas. Furthermore, Coric and Gruteser (2013) assumed that learning on more occupied roads is more relevant. However, they disregarded the fact that illegal parking grows considerably with increasing occupancy level (White, 2007). In this thesis, an approach is described that turns the identification of legal parking areas into a data mining problem. Several additional features are proposed to learn more spatio-temporal details in the parking data. Manual parameter optimization is reduced by calculation of the features at different length scales. In the case of a clustering-based classification, the parameter choice could be completely avoided.

Estimation and prediction of parking occupancy for irregularly observed parking spots

While a great deal of research has already been done for the occupancy prediction of parking lots and on-street parking with static parking sensors, the estimation and prediction of parking crowdsensing approaches is still largely unstudied (see Subsection 3.1.2). Studies using real-world parking data are also rare. Three methods for parking estimation and prediction are evaluated in this thesis: spatial interpolation, a persistence method based on the last observation, and binary classification. Additionally, the classification approach is used to estimate the probability of at least one empty parking space in a road segment. The quality of the approach is compared for different coverage levels of the mobile sensors, based on sensor data from the large scale parking project SFpark in San Francisco.

Evaluation of the quality of parking crowdsensing with probe vehicles

The data quality in parking crowdsensing is seldom studied. For the case of probe vehicles, only Mathur et al. (2010) investigated the coverage of taxis based on GPS trajectories. However, they did not perform a real map-matching of the taxi trajectories and did not consider the presence of intersections, the driving directions, and the number of lanes per road. In this work, the spatio-temporal coverage of taxis as probe vehicles is investigated in detail, solving the aforementioned shortcomings. Moreover, the probe vehicle coverage is combined with parking data to evaluate the parking data quality in the crowdsensing scenario. Finally, the relevance of crowdsensed parking data is compared with other parking-related information in a typical parking search scenario based on real-world data.

4 LiDAR-based parking availability data acquisition

This chapter presents the detection of parked vehicles using a LiDAR mobile mapping system. The recording of the 3D point clouds on a predefined test track is described in Section 4.1. The processing consists of the segmentation to gain object point clouds, their classification, and their matching to the road network (see Section 4.2). A detailed evaluation of the processing steps and the complete processing chain is presented in Section 4.3, followed by some concluding remarks in Section 4.4. As it is hardly possible to distinguish between stopping and parked vehicles in a single time instant without further knowledge about the lanes, the more precise term *non-moving vehicles* is used in the following. This work was also published in Bock et al. (2015).

4.1 Data recording

To evaluate the approach of detecting non-moving vehicles in an inner-city scenario, a test track was recorded ten times with a LiDAR mobile mapping system. The system with the sensors used for the measurements is described in Section 4.1.1. Details about the test track are given in Section 4.1.2.

4.1.1 Sensor equipment

The mobile mapping system Riegl VMX-250 (Riegl LMS GmbH, 2012) was used for the recording of the 3D point clouds. The system was installed on the roof of a VW T4 multivan. It consists of two 2D laser scanners VQ-250, four cameras, and a localization unit (see Figure 4.1). Each laser scanner has a sampling rate of 300,000 measurements per second. Measurement accuracy and precision are 10 mm and 5 mm, respectively. The scanners face backwards with an angle of 60° between each other from top view and each 35° tilted compared to a vertical scanning plane. This skew has the advantage of identifying movements of objects since the two scanners see these objects at different times and therefore at different positions. However, curbs were often occluded by parked vehicles in the measurements because of this skewed laser scanner orientation. Therefore, considering curbs for the detection approach like by Thornton et al. (2014) was hardly applicable in this situation. The recording was limited to dry weather conditions as the laser scanning system is only splash-proof.

A high precision global navigation satellite system (GNSS) receiver with Inertial Measurement Unit (IMU) (Applanix POS-LV 510) is contained in the mobile mapping system for precise positioning. In urban scenarios, its positioning accuracy is a few decimeters in height and horizontal directions (Hofmann, 2017). Therefore, the positioning unit dominates the position accuracy of objects detected. The four cameras of the system face to the sides and backwards. They were used in this investigation to extract ground truth information about the presence of non-moving vehicles.

4.1.2 Measurement campaign

Two areas in Hannover (Germany) with parking lanes were defined for the recording of 3D point clouds with the mobile mapping system. The Nordstadt district was chosen as a residential area

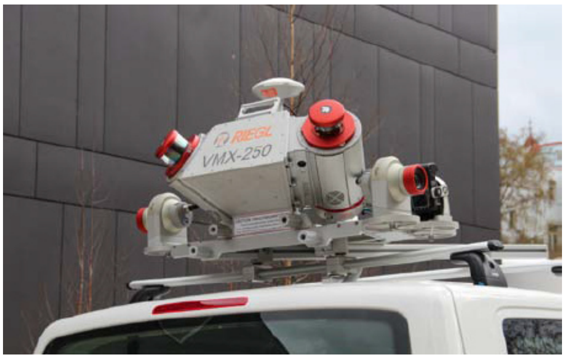


Figure 4.1: Riegl VMX-250 mobile mapping system with two 2D laser scanners, four cameras, and a global navigation satellite system (GNSS) receiver.

and the center of Hannover as a commercial district. A suitable route with an effective length of 5.5 km (excluding connecting parts without measurement evaluations) was defined to cover single-lane streets in both districts with parallel and perpendicular parking (see Figure 4.2). For this evaluation, data was recorded on this route ten times between 9 am and 8 pm.

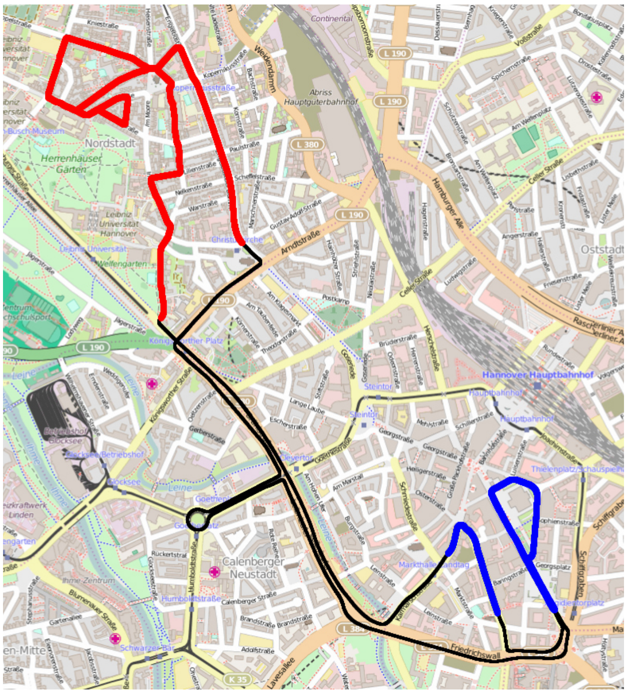


Figure 4.2: Map of the measurement track: the evaluated part in Nordstadt is colored in red, the city center in blue, and the connecting track in black. The connecting track is excluded from processing because of multi-lane roads. The background map is taken from OpenStreetMap.

4.2 Methodology

The approach for detecting non-moving vehicles from laser scanning measurements consists of five main steps, as illustrated in Figure 4.3. First, the point clouds recorded were structured and enriched in the preprocessing step (Subsection 4.2.1). Then, the point clouds were split into ground surface and distinct objects in the segmentation step, using a region growing approach (Subsection 4.2.2). The classification step (Subsection 4.2.3) divided the objects into non-moving vehicles and other objects, based on multiple characteristic features. To reduce *under-segmentation* (splitting that is too weak, so that object point clouds still contain multiple objects after segmentation), segmentation and classification were repeated in a finer raster resolution (Subsection 4.2.4). In a last step, the non-moving vehicles detected were matched to the road network, to extract statistics on distinct road segments (Subsection 4.2.5). Note that this work is based on the implementations for preprocessing, segmentation, and feature extraction by Eggert (2017). These implementations were extended and adapted for this specific case of detecting non-moving vehicles.

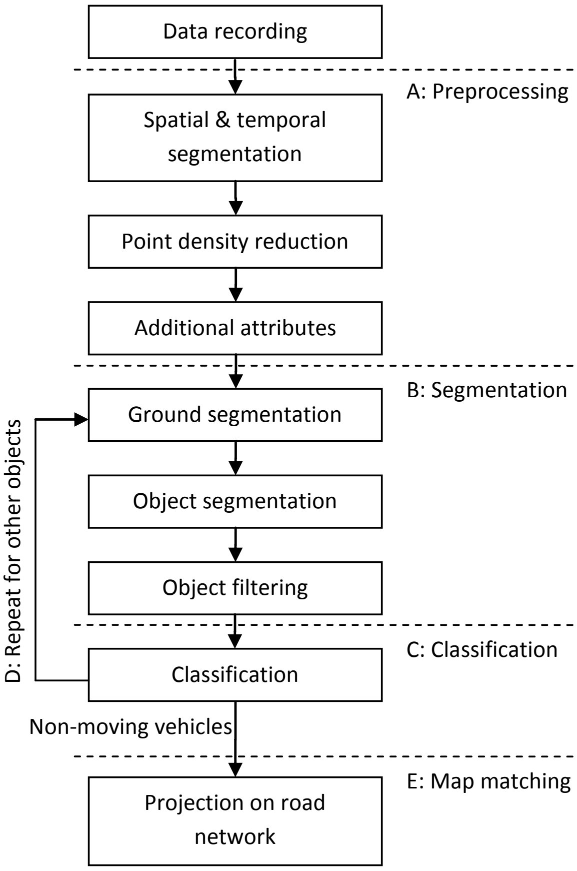


Figure 4.3: Block diagram of processing steps to extract non-moving vehicles from LiDAR mobile mapping records.

4.2.1 Preprocessing

In the preprocessing step, the point clouds needed to be structured into tiles and enriched with additional attributes. As the point clouds had more than half a billion points per measurement trip, they needed to be split into smaller units. Therefore, the area of the point cloud was spatially divided into tiles of 25 m x 25 m (see also Eggert and Sester, 2013). Some locations were passed multiple times during one trip. Therefore, the records were also separated temporally, based on a simple threshold. If the time gap exceeded this threshold (e.g. five minutes), a new time segment for the corresponding tile was created. To avoid a local agglomeration of points at almost the same positions (e.g. at very low velocity), the point cloud density of every tile was reduced by keeping only one arbitrary point per cell in a dense raster (raster edge length of 1 cm in this case).

Finally, additional attributes for each scan point were computed to facilitate further processing steps. The normal vector of every point was estimated by a principal component analysis (PCA) via the covariance matrix, using the point itself and its k-nearest neighbors. Based on the eigenvalues, every point was also assigned to one of four categories to describe the local structure: planar, linear, cylindrical, and scatter. This dimensionality classification was based on Demantké et al. (2012), which was extended by Monnier et al. (2012).

4.2.2 Segmentation

For segmentation, the preprocessed point clouds per tile were separated into distinct objects and ground surface using the points' Euclidean distance, their normal vector orientation, and their dimensionality label. To avoid the truncation of objects by tiling (Subsection 4.2.1), adjacent tiles were buffered and neighboring parts at the border of tiles were included. The segmentation approach consists of three steps: first, the ground surface was identified and extracted by basic region growing (Zucker, 1976), based on the points' normal vectors. Second, a region growing approach was again used for object segmentation in the ground-free point clouds. Remaining points were assigned to an object point cloud if they were enclosed by this object's convex hull. A raster data structure with a cell size of 15 cm was used to speed up the nearest neighbor computation for region growing. In a final step, duplicate and irrelevant objects, caused by the tile buffer, were filtered, based on their center of mass.

In the *ground segmentation step*, seed points were defined as the points with lowest z-coordinate values (the lowest 0.5 % in this case) and nearly vertical normal vector. Their cells and the adjacent cells were taken as seed cells for region growing if at least a certain number of points contained were classified as planar. All points of the identified ground cells with a nearly vertical normal vector were labeled as ground points. This procedure was repeated for adjacent cells of ground points until no further ground points could be identified. From all ground points, a plane was estimated. All points close to this plane with nearly vertical normal vector were used as seeds for a second region growing. The repeated region growing was computed to also identify ground surface points not directly connected to the initial seed points. Finally, all ground points were excluded from the point clouds for the subsequent computations.

For the *object segmentation step*, the region growing started with raster cells that contained a large number of points with planar dimensionality to first extract large building facades. A seed cell of an object grew to adjacent cells having a minimal number of points. The growing finished when there were no adjacent cells fulfilling this requirement. If the resulting object had more points than a certain threshold (e.g. 1,000 points), the object was retained. Smaller objects were discarded and the points were kept in the remaining point cloud. After the region growing was computed for every cell, the convex hulls of identified objects were calculated. Remaining points within such a hull were added to the object, assuming that small enclosed parts belonged to the enclosing object. Still unassigned points were labeled as noise and removed.

In the *filtering step*, duplicate and irrelevant objects that were caused by the additional buffer area around the tiles were removed. As the buffer area of the tiles overlapped, the same object could be generated from two different tiles. To avoid these cases, the center of mass was computed for every object. Objects were retained if they had their center of mass within the tile extents. In addition, objects were removed if the center of mass was too far from the track of the measurement vehicle (10 m in this case) since these objects were located outside of the region of interest.

4.2.3 Classification

In the classification part, all objects from the previous segmentation were assigned to the classes *non-moving vehicles* and *other objects*. The class of non-moving vehicles contains cars and also trailers, minibuses, and transporters, as those vehicles typically cover on-street parking spaces. Larger vehicles, such as buses and trucks, as well as small vehicles, such as motorbikes and bicycles, are not considered for parking statistics and were therefore assigned to the class 'other objects'. Further prominent examples for this class are building facades, trees, poles, and persons.

For this binary classification task, a random forest classifier was used, as it is fast, robust against overfitting, and easy to parametrize (see also Section 2.2.3). For classifier training and evaluation of classification performance, the segmented objects from one measurement drive were labeled manually. Multiple features were defined based on object geometry, point density, point dimensionality, and comparison of the scanner-specific point clouds to identify vehicle movements:

- **Object geometry:** Bounding boxes were estimated for every object point cloud based on a 3D principal component analysis (PCA), on a 2D PCA with fixed vertical vector, and on the driving direction of the measurement vehicle. The extents of these bounding boxes as well as the ratios between the axes were used as features since vehicles have characteristic sizes. For example, the length of a vehicle is typically between 3 and 6 meters and the height is lower than the length. As point clouds of parallel and perpendicular parked vehicles often appeared differently, consideration of the driving direction was also relevant.
- **Point density:** The point density is the ratio between the number of points and the volume of the vertical bounding box. This feature was useful to distinguish vehicles from objects with low point density, such as tree branches and scaffolding.
- **Point dimensionalities:** The ratio of points with different dimensionalities (see also Section 4.2.1) was used, because vehicles consisted of several planar parts, while leaves of trees and bushes mainly consisted of points with the dimensionality type 'scatter'.
- **Vehicle movements:** Moving vehicles appeared different from non-moving vehicles in the point clouds, depending on the relative speed between measurement vehicle and observed vehicle. If both vehicles moved in the same direction at similar speed, the scanned vehicle was visible for one or both laser scanners for a long time. This elongated shape of the vehicle point cloud could be distinguished from the regular shape of non-moving vehicles by the object length. If the observed vehicle was moving in the opposite direction, the resulting point cloud appeared like a squeezed vehicle (see Figure 4.4a). In addition, both laser scanners saw this vehicle at different positions, due to their different mounting orientation. If the relative speed was high, both point clouds did not overlap. This led to two objects in the segmentation step, each recorded by one scanner only (see Figure 4.4a). The ratio of points from each scanner allowed the identification of these objects. Slower vehicles appeared as overlapping point clouds (see Figure 4.4b). These situations were revealed by a feature consisting of the distance between the centers of the scanner-specific object point clouds. Since center differences also existed for partially occluded objects, the length difference of both scanner-specific object point clouds was also used as a feature. In other words, objects leading to scanner-specific point clouds with different center positions but the same object length were very likely to be moving.

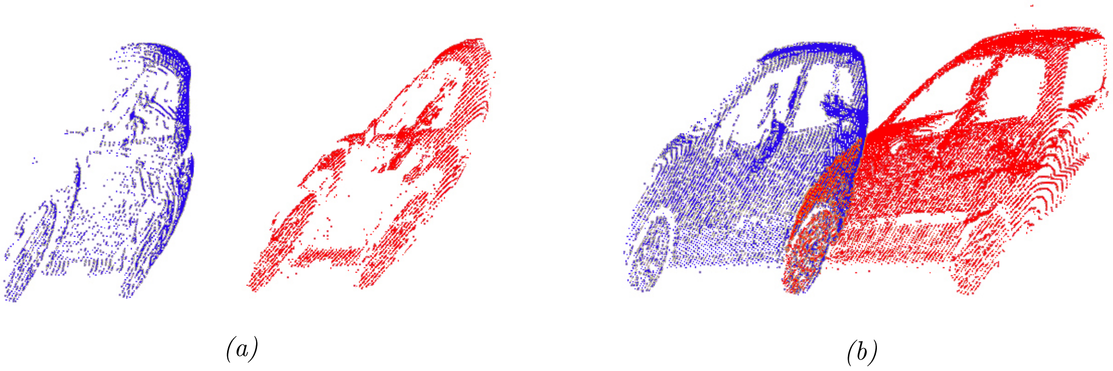

(a) (b)

Figure 4.4: Examples for moving vehicles with colors indicating which scanner recorded the points: fast vehicles lead to two separate objects (a) while the point clouds of slower vehicles overlap (b).

4.2.4 Repetition of segmentation and classification

To reduce under-segmentation (e.g. an object consists of two closely parked vehicles in Figure 4.5c), the segmentation step was repeated with a finer raster resolution after classification. Only objects classified as 'other objects' were segmented again in the same way as in Section 4.2.2, but with a finer raster cell size of 8 cm (instead of 15 cm). In the second ground surface estimation step, points need to be removed only if the first ground surface estimation is incomplete. Thus, the newly estimated ground points were removed only if these points were lower than the average height of the remaining points. After ground and object segmentation, the resulting objects were classified again with the same parameters.

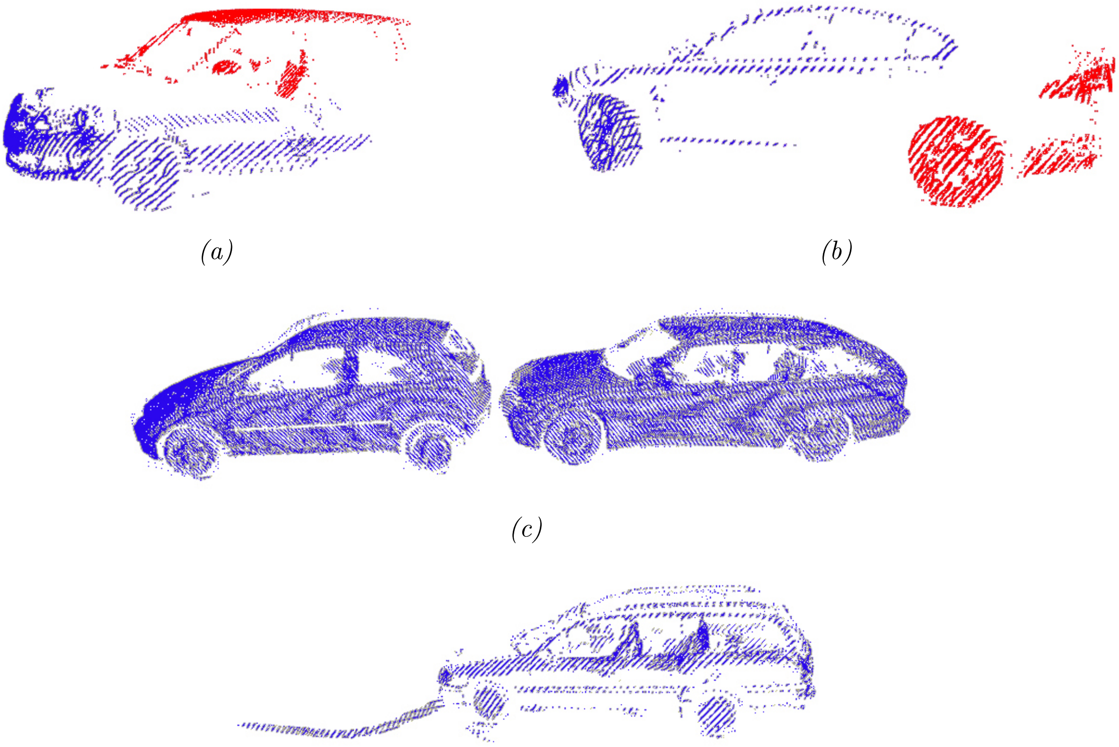
4.2.5 Matching to road network

The absolute coordinates of the detected non-moving vehicles' centers were calculated. These coordinates were fused with the road network from OpenStreetMap. Small trails and walkways were excluded from the road network. Every road on this network was split at intersections to obtain intersection-free road segments. Then, every non-moving vehicle was matched to the closest road segment based on Euclidean distance. This assignment allowed the computation of parking statistics for every road segment.

4.3 Results

To evaluate the proposed methods and their interaction, the data recorded from the first drive was analyzed in detail. First, the resulting objects of the segmentation were labeled as correctly segmented, over-segmented, or under-segmented, based on the visual appearance. The labeling focused only on the correct segmentation of non-moving vehicles. The correct segmentation of other objects was not investigated. Over-segmentation means that a physical object (e.g. vehicle, tree, person) results in multiple separate point clouds after segmentation (see Figures 4.5a and 4.5b). Under-segmented vehicle point clouds contain not only the shape of the vehicle itself, but also of another object such as another vehicle or the curb next to the vehicle (see Figures 4.5c and 4.5d). In a second step, correctly segmented objects were labeled with one of the two classes: non-moving vehicle and other objects.

Based on this labeling, the quality of the segmentation and classification is evaluated in Section 4.3.1 and Section 4.3.2, respectively. Note that partially occluded vehicles lead to truncated ve-



(a)

(b)

(c)

(d)

Figure 4.5: Examples for over- and under-segmentation: weakly reflecting vehicles lead to multiple object point clouds in (a) and (b), very closely parked vehicles are not separated in (c), and curb is not separated properly in (d).

hicle shapes, which are very challenging in the classification step even though they are correctly segmented. To also consider strongly occluded vehicles and the classification of incorrectly segmented objects, an end-to-end evaluation is performed in Section 4.3.3. For this, videos of the measurement drive were evaluated manually and a ground truth was created. Finally, an example for parking statistics over the course of one day is given in Section 4.3.4.

4.3.1 Object segmentation

From the records of the first measurement drive, the object segmentation generated 2,741 distinct objects in the investigation area. An example of a segmentation result is shown in Figure 4.6. For the first segmentation, 777 non-moving vehicles and 1,896 other objects were correctly segmented, while 13 vehicles were split into a total of 26 object point clouds (sum of first two lines of Table 4.1). Under-segmentation occurred for 42 object point clouds, which contained both vehicle and other elements. That means 97.5 % of all objects and 91.9 % of all vehicle-related objects are correctly segmented regarding vehicles after the first segmentation. Note that object point clouds with only a few points (e.g. small vehicle parts) were eliminated in the segmentation process and are therefore not part of these statistics.

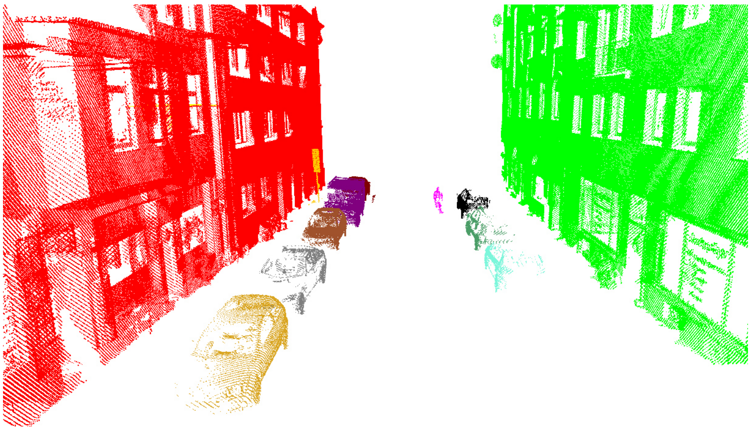


Figure 4.6: Example for the result of the object segmentation with each object in a different color.

The main reason for over-segmentation was the weak reflectance of some vehicle paint types (22 of 26 over-segmented objects, e.g. Figures 4.5a and 4.5b). Missing laser reflections led to very sparse point clouds. If the point density is too low, the region growing to adjacent cells stopped too soon and split the vehicle into multiple object point clouds. Under-segmentation was mainly caused by incomplete ground segmentation (12 of 42 under-segmented objects, e.g. Figure 4.5d) and objects very close to non-moving vehicles (25 of 42 undersegmented objects). These close objects included other vehicles which were parked very closely (14 times, e.g. Figure 4.5c), persons loading the vehicle, and close vegetation. When the segmentation step was repeated after classification with a finer cell raster for all objects classified as ‘other objects’, 33 additional non-moving vehicles were correctly segmented (last line in Table 4.1).

4.3.2 Classification

For classification evaluation, the dataset of correctly segmented objects after the first segmentation was split into training and testing datasets, according to the two observation regions, to show the generalization of the approach. Data from the city center containing 133 non-moving vehicles and 669 other objects were used as the training set. An example of the classification result is

Table 4.1: Object segmentation results. Note that there is no 2^{nd} segmentation for the city center, as this area was only used for training of the classifier.

Area	**Correctly segmented non-moving vehicles**	**Other objects**	**Over-segmented vehicle objects**	**Under-segmented vehicle objects**
1st segmentation city center	133	669	4	2
1st segmentation Nordstadt	644	1,227	22	40
2nd segmentation Nordstadt	677	1,512	26	11

given in Figure 4.7. The feature importance evaluation showed that the features of the bounding boxes (vertical orientation and extents) and the ratio of planar points contributed most to the classification performance.

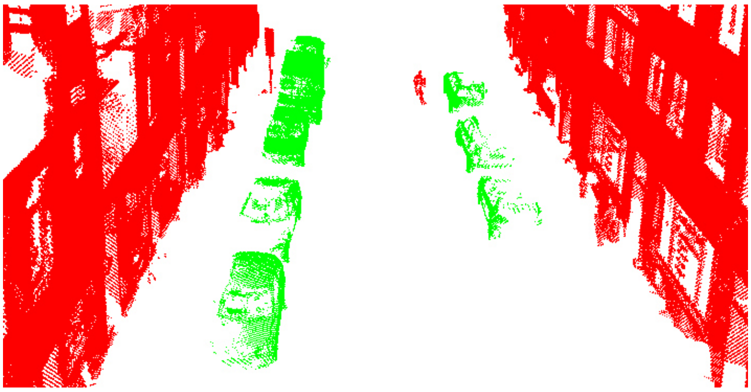


Figure 4.7: Example for the classification result with the classes non-moving vehicle (green) and other object (red).

Table 4.2: Confusion matrix of the classification result.

		Actual Class	
		Non-moving vehicles	Other objects
Predicted	Non-moving vehicles	617 (TP)	10 (FP)
Class	Other objects	27 (FN)	1,217 (TN)

The test dataset consisted of data from Nordstadt district, with 644 non-moving vehicles and 1,227 other objects. Applying the learned classifier to this dataset led to 617 correctly classified non-moving vehicles (true positives) and 1,217 correctly identified other objects (true negatives). However, 27 non-moving vehicles were not identified (false negatives) and 10 other objects were incorrectly classified as non-moving vehicles (false positive) (see also Table 4.2). This result corresponds to a precision of 98.4 % and a recall of 95.8 %. To balance between false positives and false negatives, the decision threshold of the classifier can be varied. This variation of the threshold led to the Precision-Recall curve shown in Figure 4.8. The red circle shows the parametrization chosen for the previous results in this section.

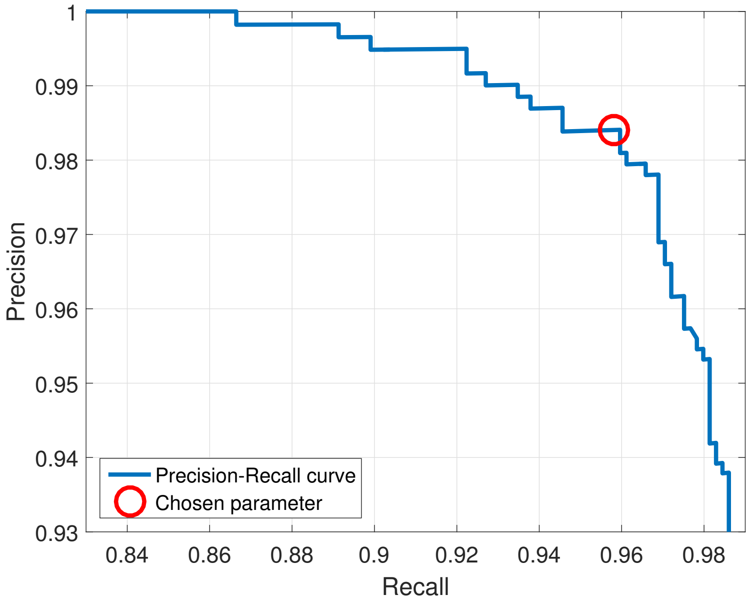


Figure 4.8: Precision-Recall curve of classification result for different decision thresholds (blue) and chosen parametrization for detailed evaluation (red).

The main sources of false positives were objects next to the street with a vehicle-similar shape (5 of 10, e.g. Figure 4.9a). False negatives mostly resulted from parking spots perpendicular to the lane (13 of 27, e.g. Figure 4.9b), even though only about 10 % of all parking spots were perpendicular. The point cloud of many perpendicular parked vehicles consisted only of the front or back of the vehicle. Therefore, the object did not have the characteristic shape of a vehicle. In addition, poorly reflecting vehicle objects were not identified in five cases.

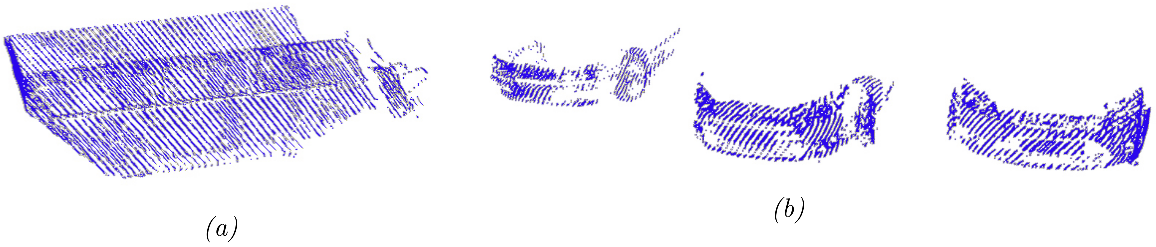


Figure 4.9: Examples for a false positive (construction trailer in (a)) and false negatives (vehicles parking perpendicular to the lane in (b)).

4.3.3 End-to-end evaluation of complete approach

The results from the complete approach, including the second segmentation and classification, were also evaluated. For this, a ground truth was generated by visual identification of non-moving vehicles in camera images from the measurement drive. In total, 672 out of 717 vehicles were correctly detected in the area inspected. Eighteen objects were wrongly marked as non-moving vehicles (false positives) and 45 non-moving vehicles were not identified (false negatives). This corresponds to a precision of 97.4 % and a recall of 93.7 %. The recall value was clearly lower than the pure classification result because the classifier did not identify most over-segmented vehicles. Occlusions caused only six missing vehicle detections. In two cases, the same non-moving vehicle was detected in two tiles, because of insufficient filtering at the tile borders.

4.3.4 Parking occupancy statistics over the day

Multiple statistics can be derived from the detections of non-moving vehicles. A highly relevant evaluation for parking management is the parking occupancy over the course of a day. An example of these statistics is shown in Figure 4.10. It shows the count of non-moving vehicles at four different street segments for all ten measurements. The lines in red and green represent the city center with high occupancy during the main business hours. Inversely, the blue and black lines from Nordstadt district show high occupancy in the morning and evening, but lower values during the day.

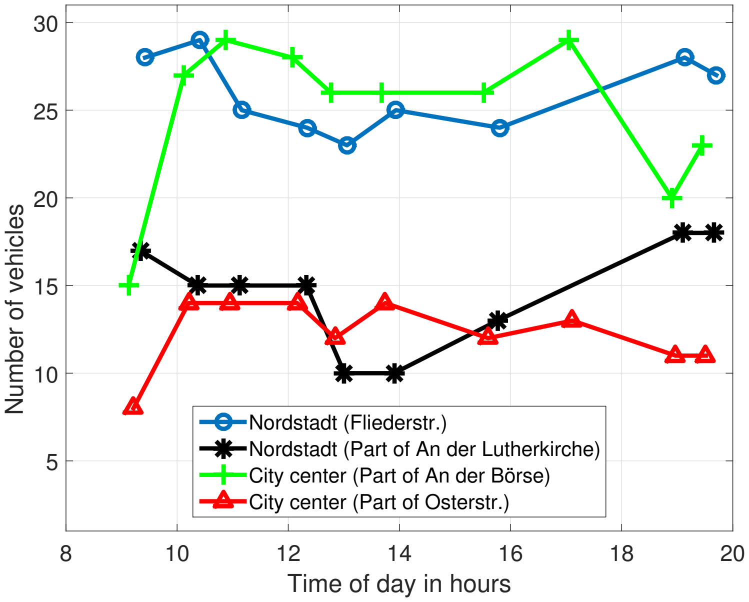


Figure 4.10: Example of parking occupancies over the course of a day for selected street segments in both districts. Note that there is one observation missing for both Nordstadt segments because of system failure during data recording in one drive.

4.4 Concluding remarks

This chapter presented a novel approach to collecting parking statistics from 3D point clouds. It has the advantage of being independent of assumptions about curb existence, road design, and parking space locations, compared with the approaches by Thornton et al. (2014) and Ono et al. (2002). The evaluation of the complete procedure showed a recall of 93.7 % at a precision of 97.4 %. While the classification step achieved a clearly better performance, the quality was mainly reduced by the segmentation step. Over-segmentations were observed for vehicles that weakly reflected the laser beam. Under-segmentations often occurred, when objects were very close to each other. Therefore, it was challenging to select a suitable cell size for segmentation based on region growing. To reduce this prolem, the repetition of segmentation and classification was introduced. The localization accuracy of the mobile mapping system is in the order of some decimeters in urban environments (Hofmann, 2017). Thus, the position accuracy of detected vehicles was in the same order. This quality is much higher than results reached by approaches using other sensors, such as ultrasonic sensors with series vehicles' GPS solutions (Mathur et al., 2010). However, as such mobile mapping systems are much more expensive, this approach is suitable only for dedicated measurement campaigns. Cheaper sensors must be used for a crowdsensing approach.

A limitation of this approach is the missing discriminability between stopping and parked vehicles. Stopping vehicles (e.g. at intersections) affect the evaluation of parking statistics. However, it was found that this is a minor issue in this dataset, as fewer than 0.5 % of non-moving vehicles were identified as stopping instead of parking, according to the manual evaluation.

5 Learning parking legality maps from parking observations

In many cities, maps of on-street parking spaces do not exist. However, this knowledge is necessary for automated detection of empty parking spaces, as they need to be distinguished from gaps in the parking lane where parking is not allowed (e.g. garage entrances). In addition, many visitors to a city do not know which streets allow on-street parking. Thus, on-street parking maps could help for a number of applications. In this chapter, a learning-based approach to automatically generate on-street parking maps from parked vehicle positions, detected by sensing vehicles, is presented. For learning, multiple sets of features are presented to characterize the occupancy of small road segments at different time instants. The use of the k-means algorithm as unsupervised learning and random forests as supervised learning are compared by applying these feature sets. The proposed method is evaluated with repeated LiDAR measurements on more than five kilometers of potential parking space length, based on the detection approach presented in Chapter 4. The work presented in this chapter is also published in Bock et al. (2016b).

5.1 Methodology

The steps to generate on-street parking maps consisted of the preprocessing of the parked vehicle detections, the computation of features based on information from multiple time instants, and the learning based on these features (see Figure 5.1). Inputs of the method were the positions of parked vehicles at different time instants (Section 5.1.1) and a road network from OpenStreetMap. The preprocessing step (Section 5.1.2) consisted of splitting the road network into small road segments and of considering each side of the road individually (called *road subsegments* in the following to distinguish them from road segments between intersections) and matching the parked vehicle information to the subsegments. The occupancy information for all road subsegments was used to calculate several features regarding the occupancy of the road subsegments themselves and the occupancy level in the neighborhood. Finally, the parking legality was determined in the learning step (Section 5.1.3). Both supervised and unsupervised approaches were evaluated. The parking legality information of road subsegments was then fused to a complete parking map.

5.1.1 Location of parked vehicles as method input

For the generation of on-street parking legality maps, the main input was the position of parked vehicles at different time instants (e.g. at different times of the day or on different days), annotated with a timestamp. This information could be obtained from various sensors, as described in Section 3.1.1. This approach is likely to work for any of these sensor types. While laser scanners, cameras, and ultrasonic sensors provide information about the position and the extent of the vehicle directly, GPS trajectories from smartphones or in-vehicle recordings provide only a single position. In the latter case, a typical length for a vehicle has to be assumed.

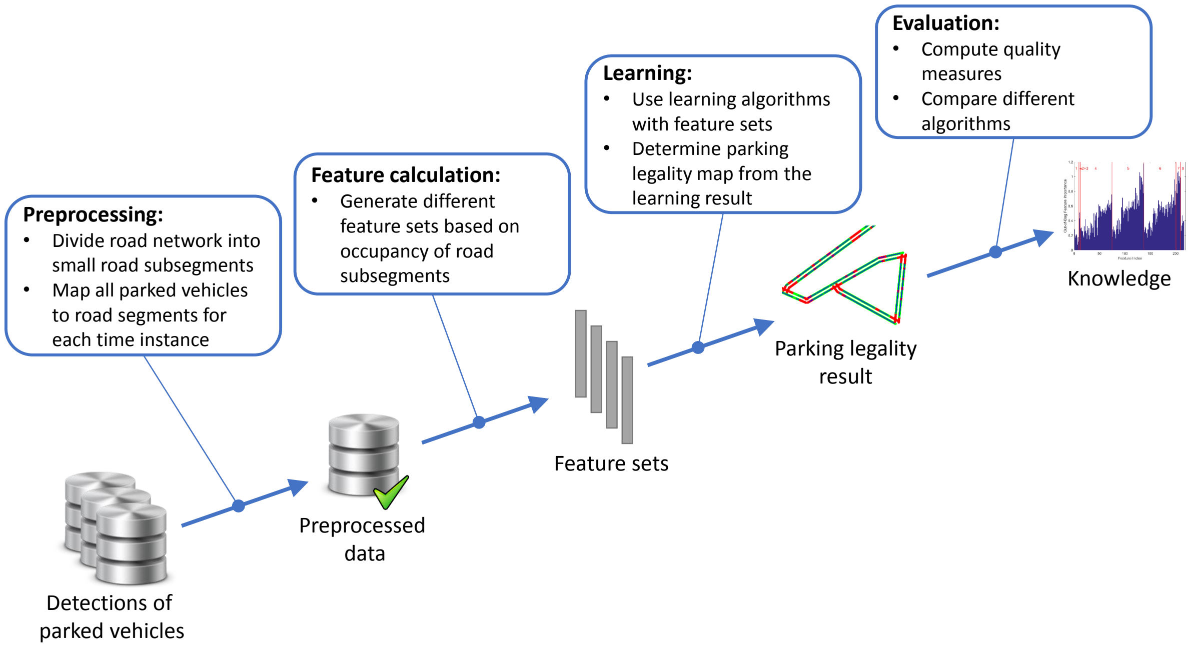

Preprocessing:
- Divide road network into small road subsegments
- Map all parked vehicles to road segments for each time instance
Feature calculation:
- Generate different feature sets based on occupancy of road subsegments
Learning:
- Use learning algorithms with feature sets
- Determine parking legality map from the learning result
Evaluation:
- Compute quality measures
- Compare different algorithms
Detections of parked vehicles
Preprocessed data
Feature sets
Parking legality result
Knowledge

Figure 5.1: Overview of the steps for the generation of the parking legality map from parked vehicle positions. The illustration is adapted to the KDD process in Figure 2.1.

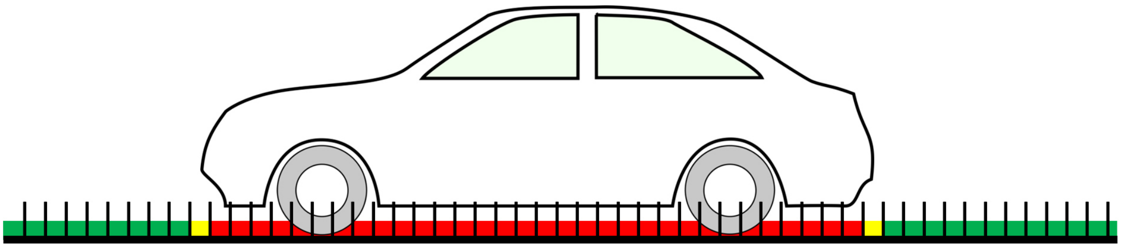


Figure 5.2: Projection of a detected vehicle on the road subsegments (space between small strokes). The color represents the occupancy: the occupancy value is 1 for completely occupied road subsegments (red), a float between 0 and 1 for partially occupied road subsegments (yellow), and 0 for unoccupied road subsegments (green).

5.1.2 Data preprocessing

The road network for the region of interest was obtained from OpenStreetMap. To obtain a road network with nodes at every intersection, links were merged and split accordingly. Then, each road segment was split into small subsegments to learn the parking legality for each of them. A subsegment length of 10 cm was chosen, but a coarser partition should also be feasible as long as small parking prohibition areas can be represented. The road subsegments were connected to a graph to determine spatial relationships. The extent of detected vehicles was projected on the road subsegments (see Figure 5.2), also considering the side of the road, and an occupancy value between 0 and 1 was determined.

5.1.3 Definition of feature sets

Eight feature sets were defined for later use in the learning step. They contained both raw and aggregated data based on the road subsegments themselves and their neighboring road subsegments. An overview is given in Table 5.1. Many feature sets contained a distance parameter for the neighborhood considered. To keep the model generic and avoid parameter optimization, the feature sets were computed for multiple generic values, similar to histogram features. The distance parameter of the features was set to 0.5 m, 1 m, 3 m, 5 m, 10 m, 20 m, and 40 m to cover the effects in both the short and more distant surroundings.

Table 5.1: Overview of all feature sets used in this evaluation.

Feature set number	Name	Size of feature vector
1	Raw occupancy	#(measurement drives)
2	Occupancy rate	1
3	Weighted occupancy rate	1
4	Raw neighbor occupancy	#(measurement drives) · #(distance values)
5	Average neighbor occupancy	#(measurement drives) · #(distance values)
6	Gaussian average neighbor occupancy	#(measurement drives) · #(distance values)
7	Segment saturation	#(measurement drives)
8	Road subsegment attractiveness	#(distance values)

FS1: Raw occupancy This feature set is the input data itself. At each road subsegment, the occupancies for each time instant are taken as features. Therefore, this feature set contains a feature column for each time instant.

FS2: Occupancy rate This feature set is the (temporal) average occupancy of the road subsegment over all time instants.

FS3: Weighted occupancy rate Weighted occupancy rate is the concept suggested by Coric and Gruteser (2013). It contains the weighted average occupancy over all time instants per road subsegment. The average is weighted by the (spatial) average occupancy of the complete road segment at every time instant.

FS4: Raw neighbor occupancy In addition to the low and high level information for the road subsegments, information about their neighbors is included. For this, the road subsegment graph is traversed and the average occupancy of all the neighbors at a certain distance for each time instant is computed. Note that there are usually two neighbors at the same distance, because there are two directions of the road. In some cases (e.g. at intersections), the number of neighbors that are at the same distance to the current subsegment can even be more than two. Therefore, just the average occupancy of all possible neighbors at each distance and time instant is calculated as a feature.

FS5: Average neighbor occupancy For each time instant, the occupancy averaged over all neighbors within a predefined range is computed. The identification of neighbors is calculated by traversing the neighborhood graph, as in the previous feature set.

FS6: Gaussian average neighbor occupancy The weighted average occupancy of neighbors is similar to the ordinary average occupancy of neighbors. The only difference is that the average is weighted by a Gaussian function. The width of the Gaussian function is chosen so that its value is 10 % of the maximal value at the distance limit. This weighting is applied assuming that closer road subsegments give a stronger hint of the parking legality, but the occupancy of distant road subsegments still provide some valuable information.

FS7: Segment saturation The segment saturation describes the occupancy level of a complete road segment. The number of occupied road subsegments of a road segment at one time instant is divided by the maximal number of occupied road subsegments of all time instants on this road segment. It is assumed that the maximum value represents the fully occupied road segment. If the value of this feature is low, parking demand is low at this time instant (or parking is not allowed in this subsegment), while the parking demand is high if this value is high.

FS8: Road subsegment attractiveness The road subsegment attractiveness represents the occupancy rate of the specific road subsegment compared with other road subsegments in the neighborhood. The occupancy rate is divided by the maximum occupancy rate of the neighbors within a certain range. A low value means that this road subsegment is less attractive than others in the surroundings. This is likely to be a clue that parking might be not allowed there. If this value is high, this road subsegment is about as attractive as the most occupied subsegment. Therefore, it is more likely that parking is allowed there.

5.1.4 Learning the parking legality of road subsegments

The decision as to whether parking at a given road subsegment is legal is a classification problem. Each road subsegment belongs to the classes *legal* or *illegal*. As described in Section 2.2.3, most classification algorithms belong to the group of supervised learning and need a training dataset with labeled data before application to another dataset. Clustering methods as a subset of unsupervised learning algorithms do not have this requirement. They group objects to clusters based on their similarity. For assignment to the correct class, however, a manual or automatic post-processing step is necessary. Nevertheless, it has the strong advantage that it can be applied to different areas without the need to generate representative training data. In the following, both approaches are described and evaluated.

Unsupervised learning: clustering-based classification with k-means

The k-means algorithm was used as a basic and established clustering algorithm (see Section 2.2.4). It has the advantages of being fast and of allowing users to define the number of clusters. The latter was important in this problem setting, since there were the two classes 'legal' and 'illegal'. However, it was unclear which cluster represents legal parking spaces. Thus, to assign the clusters to the classes, it was assumed that legal parking spaces had a higher average occupancy rate than parking prohibitions on average. This means that the algorithm assigned the cluster with a higher average occupancy rate to the class 'legal'.

Supervised learning: random forests

For supervised learning, a random forest classifier was used (see Section 2.2.3). Since the training data was assigned to the two classes 'legal' or 'illegal', the classifier directly estimated the classes for the test data and there was no need to guess the class assignment.

Approach from literature: weighted occupancy rate threshold (WORT) with smoothing

Coric and Gruteser (2013) described an algorithm that decides the legality based on a weighted occupancy rate. This value was compared with a threshold followed by post-processing to smooth the results. Since there was no method suggested to choose proper threshold values, a grid search was applied for the best parameters with cross-validation and cost function $c = \mathrm{FP} + \alpha \cdot \mathrm{FN}$ (α is a weighting parameter, FP are the false positives, and FN are the false negatives). In the following, the approach is abbreviated as *WORT*.

5.2 Evaluation

This approach to learn parking legality from the position of parked vehicles was evaluated with data using the parked vehicle detection method described in Chapter 4. The results were compared with a manually recorded ground truth. Finally, the proposed learning algorithms were compared with the WORT method of Coric and Gruteser (2013).

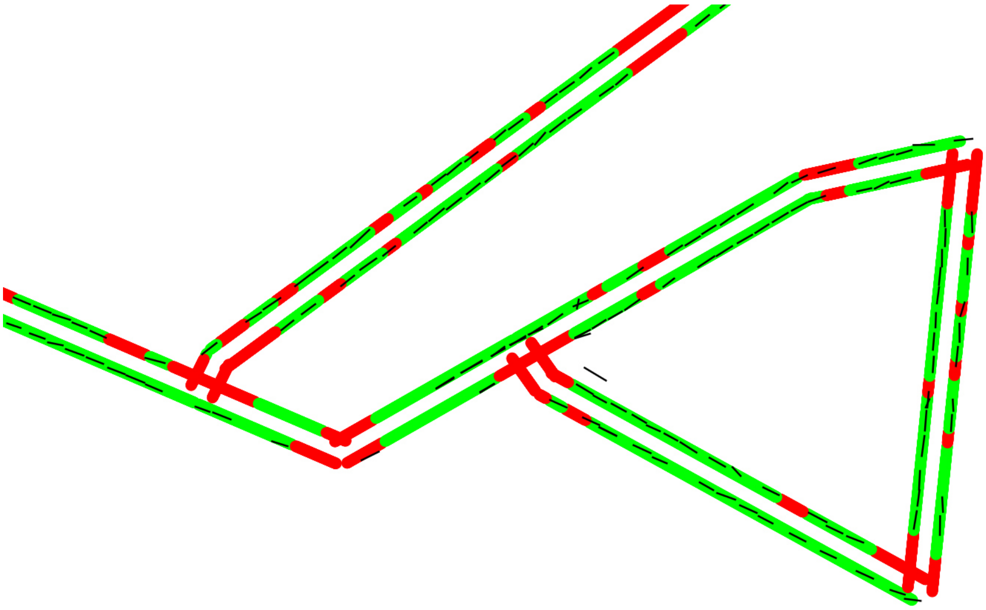
5.2.1 Evaluation approach

Test scenario

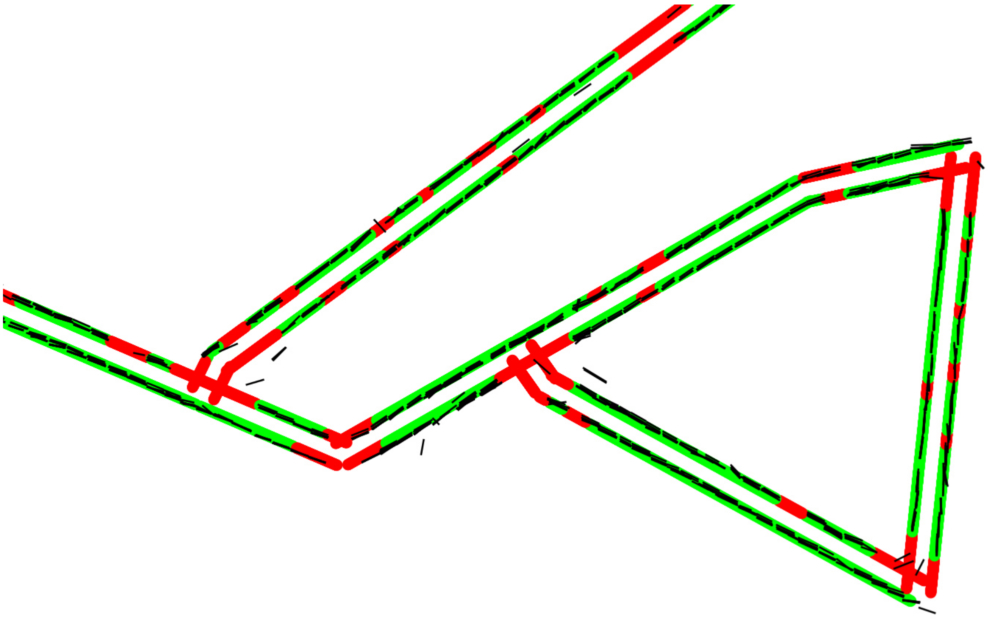
The detections of non-moving vehicles with the mobile mapping system, as described in Chapter 4, were used for the evaluations in this chapter. It was assumed that all detected non-moving vehicles were parked vehicles. Since a system breakdown occurred in one of the ten measurement drives, only nine drives were considered in this evaluation. Two examples of the extracted vehicle coordinates are shown in Figure 5.3. The data was restricted to the Nordstadt part of the test track, where parking spaces were parallel to the road. Road subsegments in areas with parking perpendicular to the road as well as parking areas for special groups such as taxis were excluded from the evaluation. In the end, the length of the track considered was more than 2.5 kilometers and thus more than five kilometers of potential parking space, as both sides of the road were observed. Parking was allowed on about 3.1 kilometers for about 500 vehicles.

Ground truth recording

The ground truth to evaluate the results of the parking space map was obtained by a combination of field measurements and precise post-processing. For most of the roads, a handheld differential



(a)



(b)

Figure 5.3: Examples for input data of (a) one measurement drive and (b) all nine measurement drives. The black lines represent the one-dimensional extent of the parked vehicles, red (illegal) and green (legal) are the ground truth classes of the road subsegments.

GPS device was used to record the beginning and end of the legal parking spaces. The standard deviation of the GPS device measurements ranged from a few centimeters to multiple meters. For streets with low GPS accuracy due to obstructions (e.g. tall buildings) and thus limited sky view, Google satellite images were used for a first estimation of the parking legality. Then, 3D point clouds from the laser scan data were overlaid, in order to refine these locations. The boundaries of the parking area, such as curb stones or traffic signs, could be clearly identified from the laser scan point clouds.

Uncertainty in ground truth

In addition to the measurement inaccuracy of the equipment, the start and end of a parking space could not be identified precisely in several cases. For example, the curb at the end of a parking lane was often not perpendicular to the road, or the parking restriction traffic sign was not aligned with the street markings. To account for both measurement and identification uncertainties, the borders of the legal and illegal parking spaces were not evaluated. More precisely, 0.5 m to each side of the borders between the legal and illegal parking spaces were not considered in the quality measures.

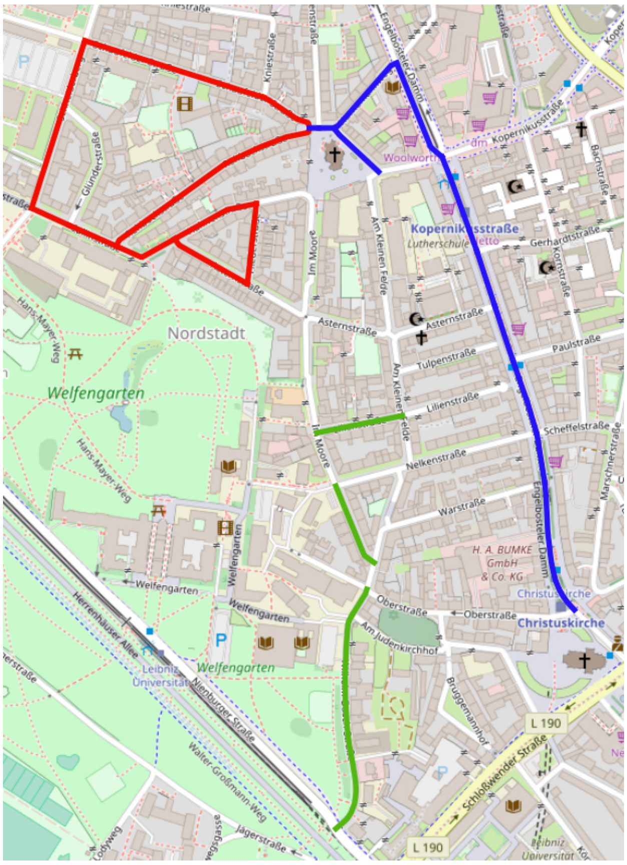


Figure 5.4: Illustration of regional subsets of investigated road segments. Each color represents a subset. Note that parts of the Nordstadt track from Figure 4.2 are missing, because they are multi-lane roads or have parking slots perpendicular to the road. The background map is taken from OpenStreetMap.

Cross-validation

A three-fold cross-validation was applied to the supervised approach and the WORT approach. The k-means approach was also evaluated for different subsets of the dataset, to investigate the impact of the area size. For this, each road segment (not subsegment) was assigned to one of three similar sized sets. Two kinds of cross-validation sets were generated to investigate the generalization of the models. For the first cross-validation sets, the track was divided based on the spatial vicinity of the road segments (see Figure 5.4; called *regional road subsets* in the following). In the second case, the road segments were assigned randomly to the three sets, keeping the length of all three sets about the same. Note that cross-validation with a random split of subsegments would lead to an improper evaluation, since adjacent subsegments have very similar neighborhood features (features 4-6), and was therefore not evaluated.

5.2.2 Results

Overview of results

A comparison of results for all three approaches is shown in Table 5.2. Both the accuracy without variation of cost weights (unweighted) and the optimal accuracy values of each approach were computed for different choices of cross-validation sets. The results showed that the random forest with random road subsets reached the highest quality scores, with 93.0 % accuracy in the unweighted case and 93.1 % optimal accuracy. Most other cases also reached accuracy values above 90 %. Only the WORT approach achieved an accuracy of no more than 85.5 % if the suggested parameters by Coric and Gruteser (2013) were used. This result might be caused by potentially different parking characteristics in their observation area, such as different parking occupancy levels or different arrangement of parking stalls.

Figure 5.5 shows the Receiver Operating Characteristic (ROC) curves, which illustrate the pairs of False Positive Rate (FPR) and True Positive Rate (TPR) for different weightings. For random forests, this curve was computed by variation of the classification score threshold. The separation plane between the cluster centers was incrementally shifted for k-means. For WORT, the relative weight of FP and FN was varied with parameter α in the parameter grid search. The best results (highest TPR at a given FPR) were achieved by k-means and random forest with random subsets for almost the complete curve. Clearly worse were the curves for WORT and random forest with regional subsets.

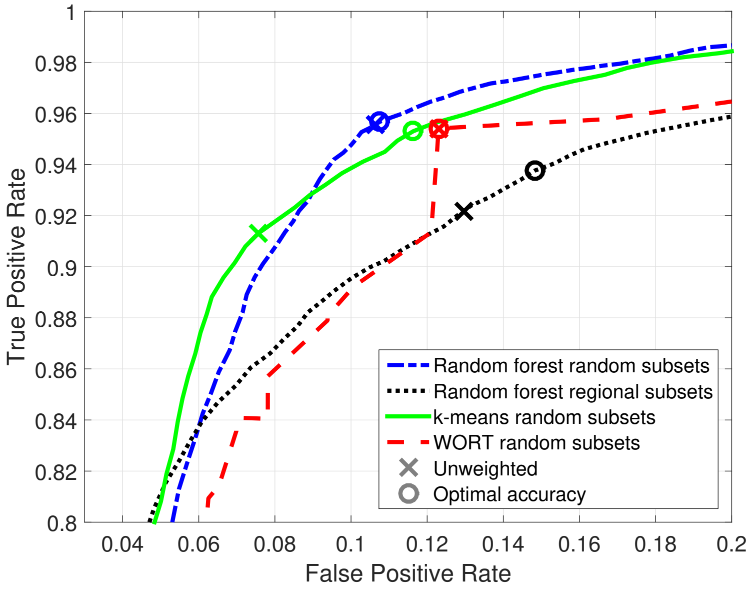


Figure 5.5: Comparison of Receiver Operating Characteristic (ROC) curves for random forest, k-means, and WORT.

Table 5.2: Results for all methods with different choice of data subsets and parameters.

Method	Choice of data subsets and parameters	Accuracy (unweighted)	Optimal accuracy
Random forest	regional road subsets	90.1 %	90.3 %
	random road subsets	93.0 %	93.1 %
k-means	regional road subsets	91.3 %	92.2 %
	random road subsets	91.8 %	92.5 %
	full dataset clustered	91.8 %	92.1 %
Weighted occupancy rate thresholding with smoothing (WORT)	suggested parameters	85.5 %	–
	optimized (regional road subsets)	92.0 %	92.3 %
	optimized (random road subsets)	92.3 %	92.3 %

Impact of subset choice in cross-validation

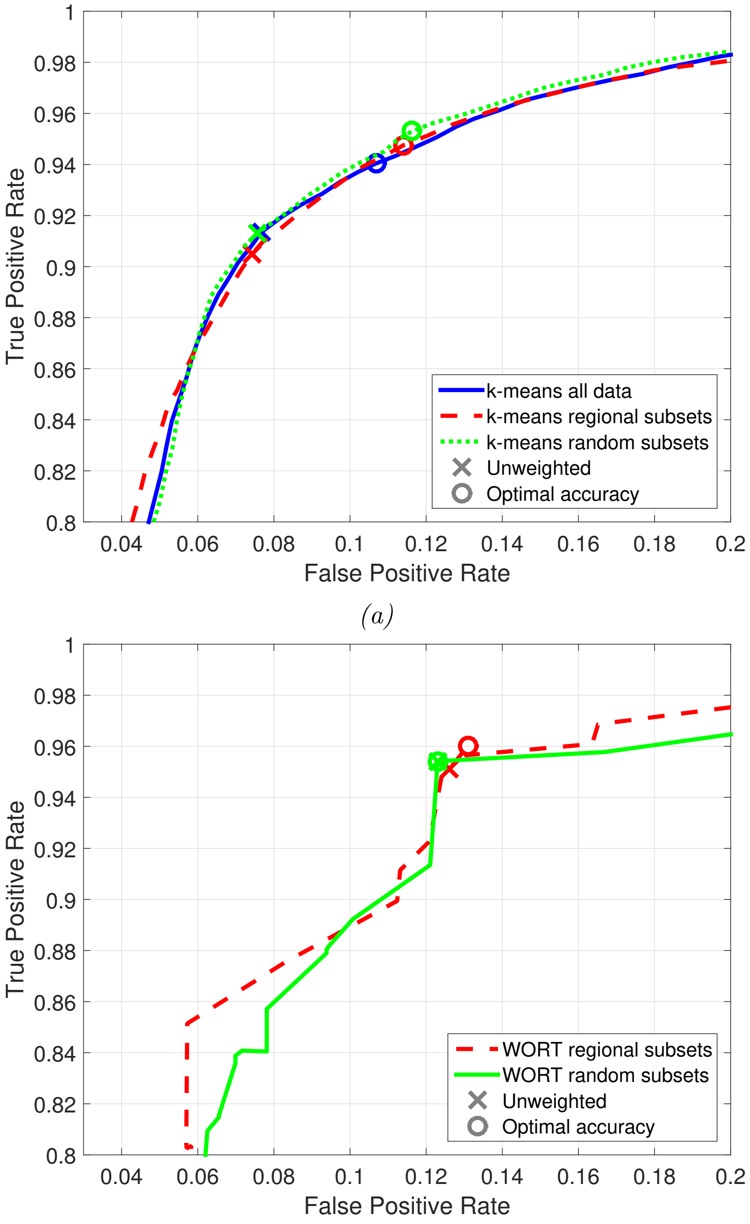
Figure 5.5 showed clear differences in the random forest approach for different subsets in the cross-validation. If the roads were divided into three sets according to their geographical locations, the ROC curve was clearly worse than for a random split of roads into three sets. The area under the curve was 0.953 for the regional subsets, compared to 0.966 for the random split of roads. Such differences between different subsets in the cross-validation did not exist for k-means and WORT (see Figures 5.6a and 5.6b). It is assumed that different parking characteristics (e.g. different parking occupancy levels) in the different regional subsets caused learning that was too specific for model transfer. When the road segments were chosen randomly for the subsets, each subset contained about the same characteristics and was therefore more representative for the other subsets. Since the random forest classifier was able to learn a finer separation than k-means with just a linear separator, the random forest results depended more on the representative choice of training data than the other methods.

Qualitative evaluation of methods

The qualitative evaluation of the results showed that all methods provided a reliable decision for parking legality in most situations. In particular, long parking lanes and highly occupied parking spaces were well identified with all methods. However, false positives mainly occurred in small areas of parking prohibition, such as in front of garage entrances, if vehicles were occasionally parked there during the observation time (e.g. road furthest to the right in Figure 5.7). False negatives were less frequent. They appeared at rarely used parking places, such as at the end of parking lanes (e.g. Figure 5.8a). Moreover, systematically missing detections of parked vehicles occurred at a few places. Without any detection, the learning also failed at these places. Differences between the methods were found at the beginning of illegal zones. The random forest approach often classified a few additional meters as legal in the illegal zone. In situations with only one vehicle parked illegally for a few time instants, k-means and random forest interpreted these situations correctly, while the WORT approach marked these spots as legal (see Figure 5.8b).

Evaluation of features

The relevance of the feature sets was evaluated both for the unsupervised and supervised learning approach. For unsupervised learning with k-means clustering, the results for runs with all and with a subset of feature sets were compared (see Figure 5.9). If only the features on the road subsegments themselves were considered (feature sets 1-3) and the neighborhood features were



(a)

(b)

Figure 5.6: Comparison of different cross-validation subsets for (a) the clustering-based classification with k-means and (b) the WORT approach.

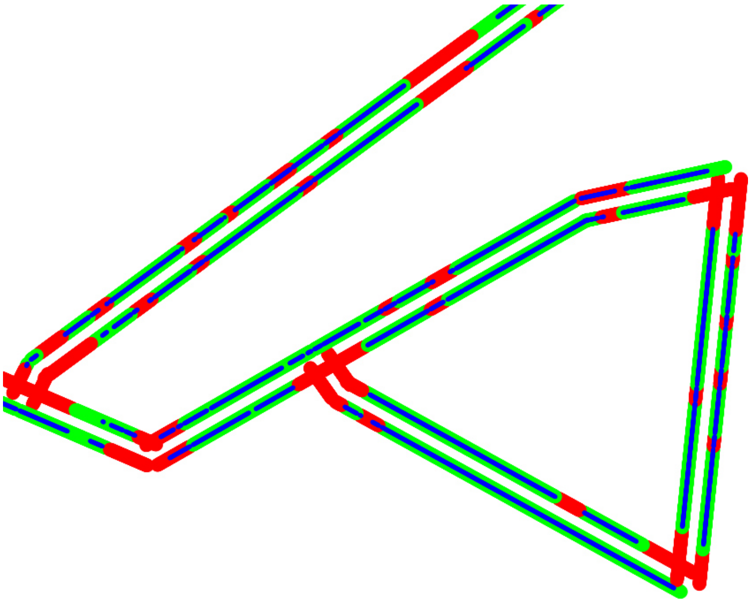
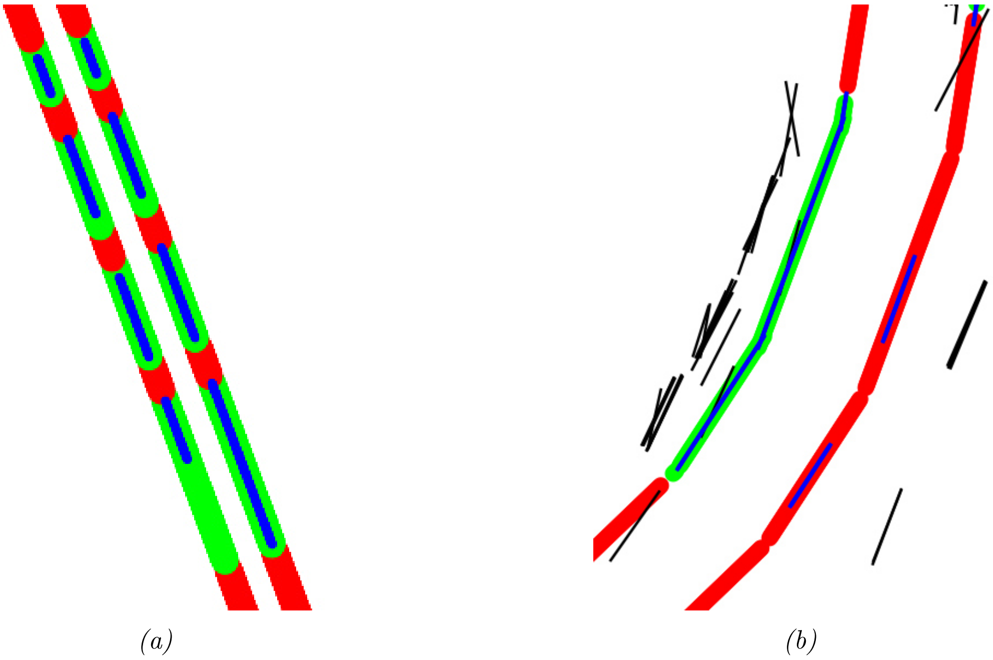


Figure 5.7: Example for the resulting parking map. Blue is the estimated parking space, red (illegal) and green (legal) show the ground truth.



(a) (b)

Figure 5.8: Examples for wrong classification results (estimated parking space in blue, ground truth for legal/illegal parking in green/red): (a) shows false negative results at the end of a parking zone. (b) visualizes two false positive parking spaces for WORT. The black lines represent the raw vehicle detections. At the two wrong parking spaces (false positives), the same vehicle was detected two and four times, respectively.

ignored, the resulting ROC curve was considerably worse than the curve with all features. A main cause for this result stemmed from marked parking spaces where gaps between the parked vehicles were always at the same positions and cars rarely covered that space. Thus, these gaps were wrongly classified as illegal. If only the neighborhood features (feature sets 4-8) were used, the result was very similar to the usage of all features. Most differences were only at the end of parking lanes, where this result was less accurate than using all features.

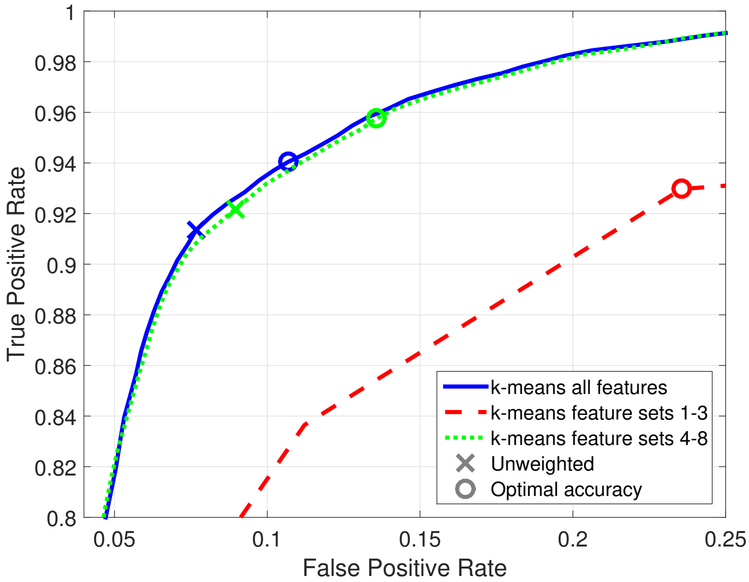


Figure 5.9: ROC curve for different feature sets with k-means.

For supervised learning, the random forest method computed the feature importance in the training step with the out-of-bag samples of each tree. A plot of the feature importance is given in Figure 5.10. All features had a positive effect on the classification result. The most important feature sets were the average neighbor occupancy (5), weighted neighbor occupancy (6), and segment saturation (7). For the feature sets 4 to 6, an increasing trend for increasing feature index is visible. As the feature index increases with the distance threshold in the feature computation, the first values correspond to the shortest distance of 0.5 m and the last values are for long distances of 40 meters. Consequently, the neighborhood features were more important when including farther distances, but still relevant for shorter distances.

Evaluation of required number of measurement drives

To investigate the influence of the number of measurement drives on the parking map result, the methods presented were computed for different numbers of measurement drives. For each number of measurement drives, the random choice of drives and computation was repeated nine times. The result is illustrated in the boxplots in Figure 5.11. The figure shows that the (unweighted) accuracy values mostly improved with increasing number of measurement drives for all methods. For all settings, the random forest reached the best results. K-means was clearly worse than WORT for a small number of drives, but became about equal for more drives. At seven and more measurement drives, only small improvements were observed for all approaches.

5.3 Concluding remarks

A novel approach for the generation of on-street parking maps from parked vehicle positions using supervised and unsupervised learning methods was presented in this chapter. Multiple feature

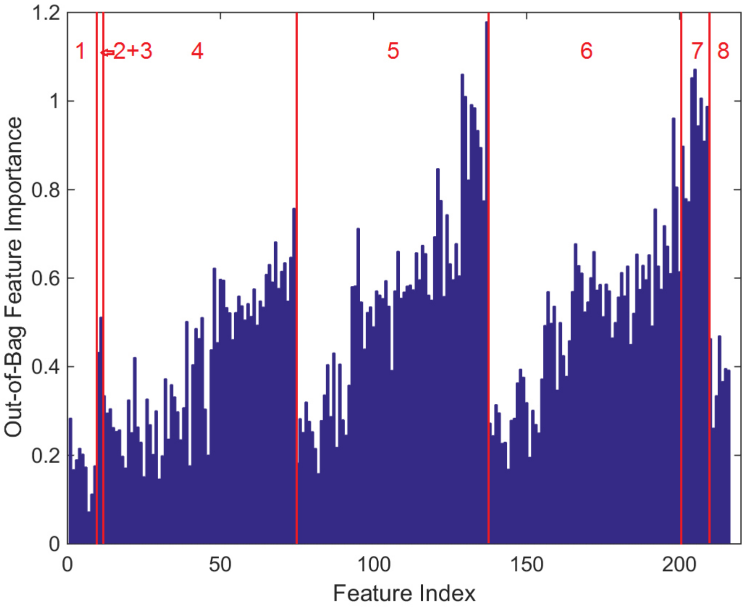


Figure 5.10: Feature importance from random forest training. The intervals of the feature index correspond to the feature sets separated and named in red.

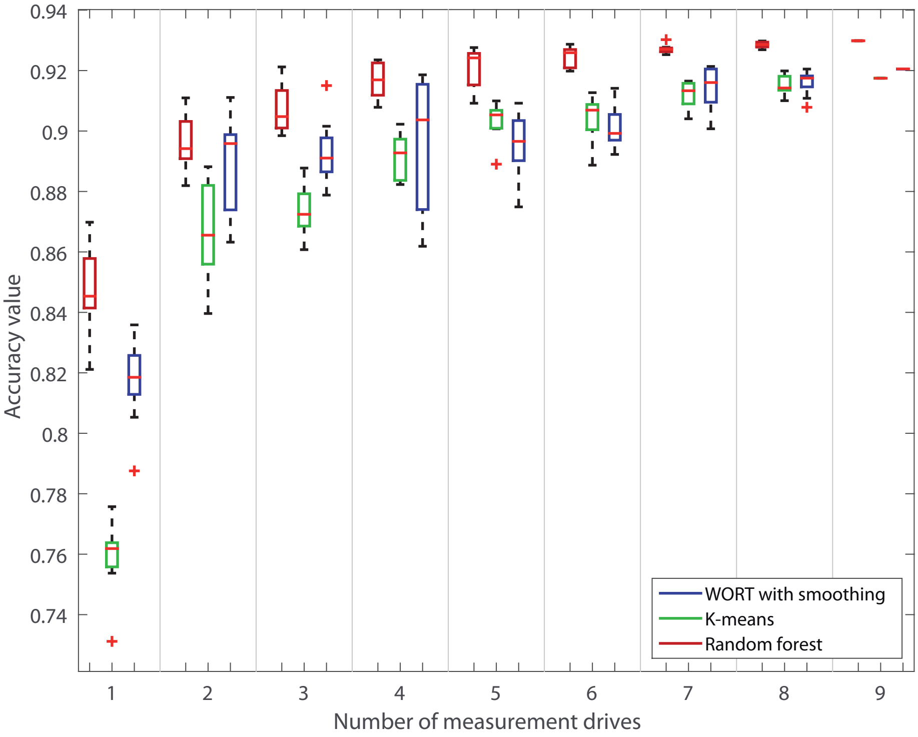


Figure 5.11: Accuracy boxplots of different methods for different numbers of measurement drives. Note that for nine drives, there is no variation for different permutations, since all drives are used in the calculation.

sets were proposed to describe the occupancy characteristics of small road subsegments and their surroundings. It was shown that both proposed learning methods showed slightly better results than a method from the literature, while keeping the model more generic. Most interestingly, the main advantage of the unsupervised clustering-based classification is that it is free from any parameter choice and optimization. Still, it reached results that were competitive with the supervised learning. In addition, it was robust against the variation of parking characteristics for different areas. The random forest method reached the best results for any number of measurement drives. However, it revealed a clear dependence on the representative choice of the training dataset.

All approaches were sensitive for untypical occupancy behavior. If a legal parking space was never occupied in the dataset, it was mostly not identified. Similarly, if no-parking zones were occupied by parked vehicles most of the time, the prohibition was barely detected. This was often the case at garage entrances. However, a very high accuracy of a parking legality map is likely to be necessary, because drivers should not be guided to a spot where parking is illegal. Thus, even the accuracy of more than 93 % achieved by the random forest classifier might be insufficient for a successful parking guidance. Since the evaluation was based on data from only one day, the dataset contained multiple situations where a parking space was occupied for a long time by the same vehicle. Thus, measurements from different days and therefore with a higher heterogeneity are likely to improve the results. However, it can be expected that some places are still incorrectly classified with large amounts of data. Finally, a highly accurate mobile mapping system was used for the detection of parked vehicles. It can be assumed that more measurement drives are necessary for cheaper sensor systems, such as sensors in series vehicles.

6 Spatio-temporal analysis of large scale parking availability data and simulation of crowdsensing

Parking availability is a highly dynamic phenomenon, with spatially heterogeneous distribution and temporally recurrent patterns. In this chapter, the spatio-temporal characteristics are described for a detailed parking dataset from San Francisco, where measurements from more than 5,000 parking spaces with static sensors were recorded. In Section 6.1, a basic description of the SFpark project is given, including the most relevant features. In addition, as the dataset is also affected by some sensor errors, the data cleaning from irregular patterns is described. In Section 6.2, a detailed analysis of the dataset regarding temporal periodicity and characteristics is presented. In addition, a clustering of the average parking occupancy over the day is presented in Section 6.3 to show their similarities and differences in these daily patterns. Spatial properties are described in Section 6.4. Finally, the simulation of parking crowdsensing based on observation rate estimations (Section 6.5) and taxi trajectories (Section 6.6) is described. Some of the work in this chapter is also presented in Bock and Sester (2016); Bock et al. (2016a, 2017a,b).

6.1 Description and processing of parking dataset from SFpark

The SFpark project in San Francisco (US) was one of the largest smart parking projects in the world (Lin et al., 2017). The San Francisco Municipal Transportation Agency (SFMTA) ran this project with a total budget of more than 46 million dollars, in order to improve parking efficiency and drivers' experience (San Francisco Municipal Transportation Agency, 2014). A main part of the project was the evaluation of demand-responsive pricing, i.e. the price at parking meters was changed every three months, based on the average occupancy in the specific places and time slots. To monitor parking occupancy continuously, more than 8,000 on-street parking spaces were equipped with sensors in a pilot and a control area. The pilot area is considered in this thesis. It contains more than 5,000 on-street parking spaces in 579 parking segments[1]. The project covered business districts (Financial District, South of Market, Civic Center), residential districts (Marina District, Fillmore District, Mission District), and a touristic area (Fisherman's Wharf). A map of the pilot area is shown in Figure 6.1.

Each parking space incorporated at least one magnetometer in the asphalt to detect the parking occupancy changes. Detections were transmitted to a roadside unit and then to a central server. The number of available parking spaces per parking segment was computed there. During the project duration, some sensor inaccuracies were noticed: electromagnetic interference from overhead power lines and other sources caused inaccurate measurements. The battery lifetime of the sensors also decreased faster than expected and caused missing measurements. Thus, data cleaning was necessary. A more detailed description, experiences, and results of the project were extensively documented by the San Francisco Municipal Transportation Agency (2014).

---

[1] The parking segments are defined based on the grouping of parking spaces by SFpark (called *block face* in their documentation). A parking segment consists of a row of metered and marked parking spaces on one side of the road between two major intersections. In the following, a distinction is made between *road segment* and *parking segment*. For every parking segment, there is a corresponding adjacent road segment.

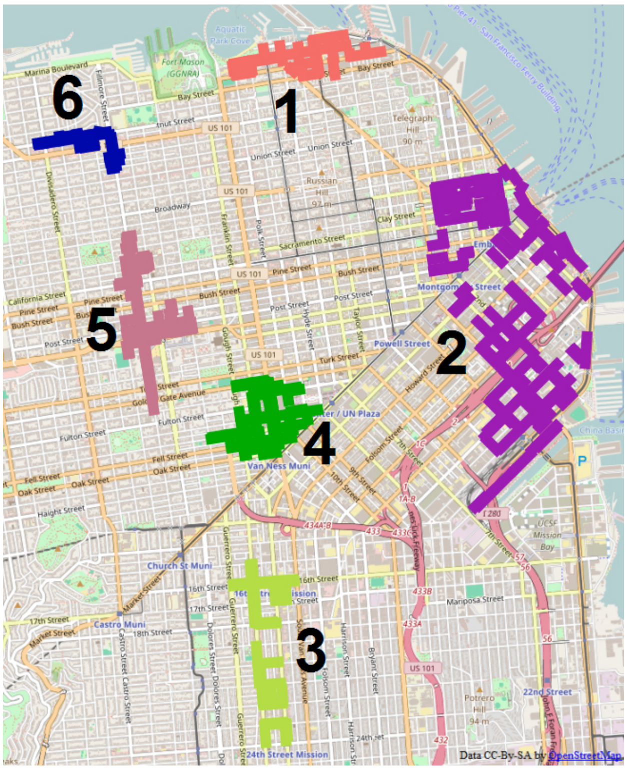


Figure 6.1: Map of pilot areas of SFpark: Fisherman's Wharf (region 1), Financial District and South of Market (region 2), Mission District (region 3), Civic Center (region 4), Fillmore District (region 5), and Marina District (region 6). The background map is taken from OpenStreetMap.

For the pilot area, the parking availability and capacity per parking segment were provided to the public through a smartphone app and a public API, so that drivers were informed about the current parking situation. For this thesis, a recording of this API every five minutes was used for a six-week period from June 13th, 2013 until July 24th, 2013 for all on-street parking segments. Each record contained the ID, name, and coordinates of the parking segment, the number of occupied parking spaces, the current parking capacity (number of parking spaces where parking was allowed at that time instant), and the timestamp. An example of the time series for parking capacity and occupied parking spaces for one parking segment over three days is shown in Figure 6.2. In this case, the maximal capacity of the parking lane was six, but changed over time. On the morning of July 12th, 2013, parking was completely prohibited, presumably due to street cleaning. For the number of occupied parking spaces, a typical day trend was visible, with high occupancies during the day and low values during the night.

The data showed some irregular patterns for several parking segments, which were removed from the dataset. Out of 579 parking segments, for which at least one measurement existed in the observation period, the records for 51 parking segments contained a data gap or a constant occupancy value for at least three days. Among the remainder, there were also 108 parking segments which never reached an occupancy rate of at least 85 %, which was an indicator of broken sensors, and were therefore filtered. The remaining cleaned set of 420 parking segments was used for all the following evaluations.

Finally, the locations of the sensors needed to be matched with the OpenStreetMap road network. Each parking segment in the SFpark project is described by a line geometry. They were assigned to the road segment in OpenStreetMap closest to the middle point of the parking segment and not

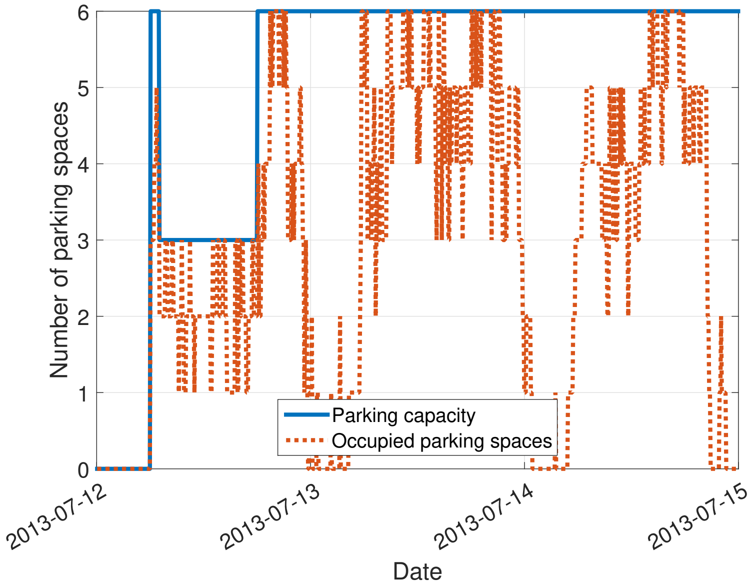


Figure 6.2: Example for the occupancy and capacity time series of a parking segment over three days.

intersecting it. A detailed visual evaluation of the results showed that this simple approach was sufficient to match all parking segments correctly.

Table 6.1: Road classes of the parking segments for the investigated districts. The ratios are based on the lengths of the corresponding road segments.

Region	**Primary**	**Secondary**	**Tertiary**	**Unclass.**	**Resid.**	**Service**	**Other**	**km**
Region 1	20.7 %	0 %	6.1 %	0 %	73.3 %	0 %	0 %	4.3
Region 2	32.5 %	29.0 %	15.1 %	2.8 %	19.1 %	1.4 %	0 %	18.9
Region 3	0 %	59.4 %	3.8 %	0 %	31.3 %	0 %	5.5 %	3.2
Region 4	25.4 %	28.8 %	16.1 %	14.1 %	7.5 %	8.1 %	0 %	7.1
Region 5	6.1 %	1.0 %	38.5 %	2.6 %	43.6 %	0 %	8.2 %	7.6
Region 6	3.9 %	0 %	46.1 %	0 %	50.0 %	0 %	0 %	3.8
All regions	21.0 %	21.2 %	20.2 %	3.9 %	30.1 %	1.9 %	1.8 %	44.9
SF City	9.2 %	8.0 %	8.8 %	3.3 %	53.3 %	9.8 %	7.5 %	2,062.9

The distribution of road classes for considered parking segments in the different districts is shown in Table 6.1[2]. Based on the lengths of the corresponding road segments, the relative lengths of road classes were computed. Note that the distribution of road classes was influenced by the choice of the SFpark pilot area and does not necessarily represent the complete district. Still, it corresponds to the main description of the districts: the parking segments in the business districts (regions 2 and 4) have the highest ratio of large roads and relatively small ratios of residential roads. The residential areas (regions 3, 5, and 6) contain many residential roads and a few primary roads. The touristic area in region 1 has both a significant ratio of primary roads and many residential roads. Compared to the complete road network for the city of San Francisco, the distribution is similar, but residential roads are underrepresented in the SFpark pilot area, presumably because the SFpark project focused on the city center with high parking demand and metered parking.

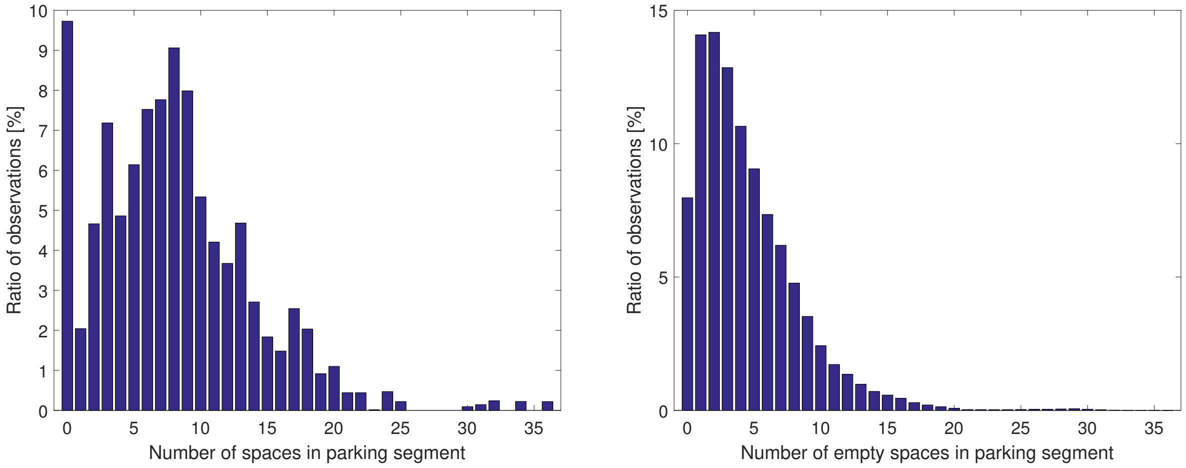
---

[2] Note that the sum of ratios may slightly differ from 100 % due to rounding.

6.2 Time series analysis of parking availability data

Characteristics of parking capacity and availability

The distribution of parking capacities and empty parking spaces over the complete dataset period is illustrated in Figure 6.3. Parking was fully prohibited in nearly 10 % of all cases. The main reasons for parking prohibition were street cleaning and peak hour drive way[3] periods. When parking was allowed, the capacity in a street segment ranged from 1 to 36 parking stalls with an average of about nine and a mode at eight parking spaces. All parking spaces were taken in 8 % of the cases. In more than 28 % of the cases, there were only one or two parking spaces left.



(a) Parking capacities

(b) Empty parking spaces

Figure 6.3: Histograms with ratios of (a) parking capacities and (b) empty parking spaces accumulated over all parking segments and time instants (frequency of specific number of spaces divided by all cases). For the number of empty parking spaces, all cases with parking prohibition (capacity = 0) were excluded.

Periodicity of parking availability

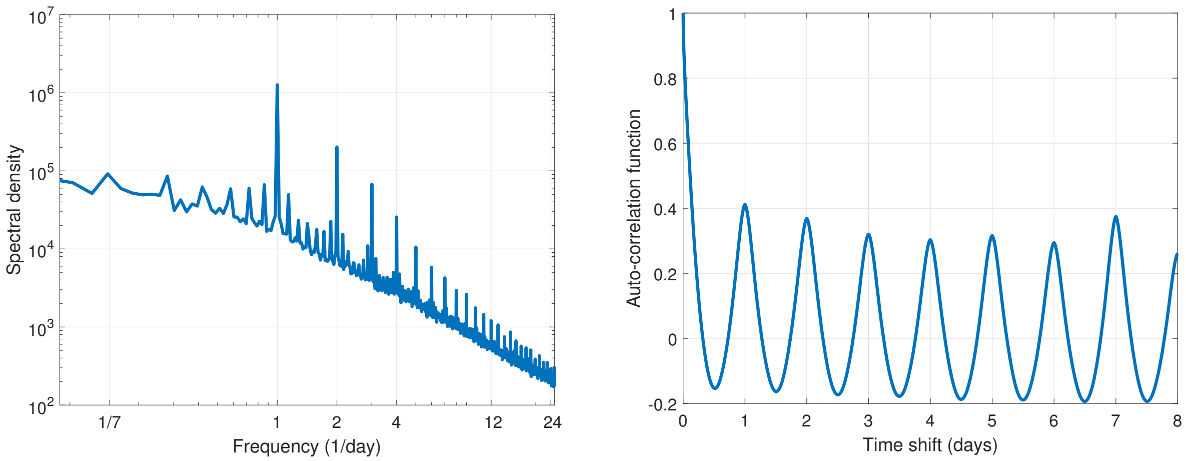
The temporal characteristics of the parking availability time series were addressed by a Fourier analysis and the auto-correlation function. Figure 6.4a shows the Fourier transformation for parking availability averaged over all parking segments in the six-week period. The global maximum was at a frequency of 1/day, meaning that the time series had a highly repetitive daily pattern. A further notable maximum was at 1/7 1/day (=1/week) indicating that there was also a weekly pattern, much less distinct than the day peak. The peaks at multiples of 1/day represented the harmonics of the 1/day main peak. For the auto-correlation function in Figure 6.4b, there was also a main peak at one day. However, the peaks for multiples of one day were just a little lower. In this result, it is visible that there was a small week characteristic, as the peak at seven days is just slightly higher than the other day shifts.

6.3 Clustering of parking occupancy daily pattern

The parking segments showed a repetitive daily pattern, however these patterns can be very different from one parking segment to another. Consequently, a clustering of the parking occupancy daily patterns was computed. For this, the parking occupancy daily patterns were calculated by averaging the occupancies over the complete dataset per time instant of the day. Weekend days and

---

[3] Some parking lanes are used as drive ways during peak hours. Parking is prohibited during these periods.



(a) Fourier transformation

(b) Autocorrelation function

Figure 6.4: Analysis of the periodicity characteristics of the parking availability rate using (a) the Fourier transformation (periodogram) and (b) the autocorrelation function (correlogram).

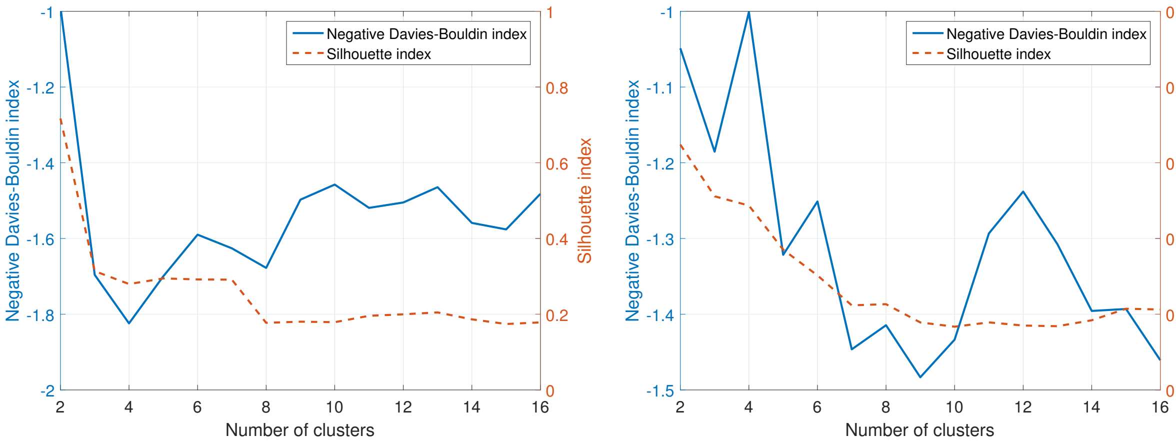
a national holiday (July 4th) were excluded, as there was a different daily pattern on those days. In addition, segments with parking prohibition at specific hours for every day were not considered, reducing the dataset to 317 parking segments. Then, a hierarchical clustering with complete linkage was computed for both the Euclidean distance and the Pearson correlation as (dis-)similarity measures (see Section 2.2.4 for details about the method). The Euclidean distance was used to find parking segments with similar absolute parking occupancy rate. The Pearson correlation was used to identify parking segments with similar curve shapes. Thus, for the Pearson correlation, the average occupancy values $\bar{\text{occ}}(p,t)$ were normalized per parking segment (subtracted by the temporal mean $\bar{\text{occ}}_{\text{mean}}(p)$ and divided by the standard deviation $\bar{\text{occ}}_{\text{std}}(p)$):

$$\bar{\text{occ}}_{n}(p,t) = \frac{\bar{\text{occ}}(p,t) - \bar{\text{occ}}_{\text{mean}}(p)}{\bar{\text{occ}}_{\text{std}}(p)} \tag{6.1}$$

where p is the parking segment and t is the time of the day in 5-minute steps. The Davies-Bouldin index and the Silhouette index were used to indicate reasonable numbers of clusters. The negative value of the Davies-Bouldin index was used in the following, so that higher values mean better clustering results for both measures.

The results of the cluster validation measures are shown in Figure 6.5. For the Pearson correlation (Figure 6.5a), these measures indicated that two clusters were the best quantity. The results of the average daily patterns are presented in Figure 6.6. For two clusters and the Pearson correlation as a similarity measure, the averaged daily patterns were two very distinct shapes (Figure 6.6a): a large cluster with 306 members and the main day behavior with low occupancy during the night hours and maxima in the early afternoon and the evening (Figure 6.6c). The small cluster had only 11 members and had lower occupancy values during the day (Figure 6.6e).

In addition, six clusters represented a local maximum of the negative Davies-Bouldin index and a medium value for the Silhouette index. The average daily patterns of those clusters were more versatile, representing the different possible shapes of the curves (Figure 6.6b). For example, cluster 5 had a medium occupancy during the night, a small maximum in the early afternoon and a prominent maximum in the evening (Figure 6.6d). Cluster 4 had a similar shape, but with the global maximum at about 1 pm (Figure 6.6f). In Figure 6.7, a map of the spatial distribution of the six clusters is shown. Although no spatial information was used for the clustering, spatial patterns



(a) Pearson correlation *(b) Euclidean distance*

Figure 6.5: Negative Davies-Bouldin index and Silhouette index as cluster validation measures for different cluster sizes: (a) the Pearson correlation as similarity measure with normalized average parking occupancy and (b) the Euclidean distance as dissimilarity measure with average parking occupancy.

are also clearly visible. This means that the shape of the average daily parking occupancy was often similar for parking segments in the vicinity.

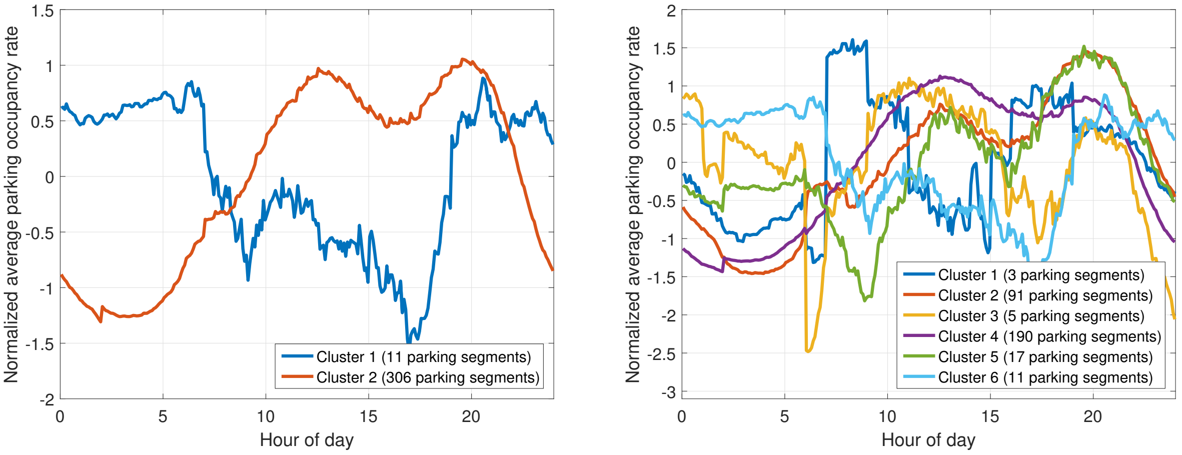
For the Euclidean distance, the results of the cluster validation measure are shown in Figure 6.5b. With four clusters, the negative Davis-Bouldin index had its global maximum and the Silhouette index only slightly decreased. In this case, the clustering led to three clusters with very similar shapes of the average occupancy daily pattern, but at different levels (Figure 6.8a). Only one cluster with two members was strongly different, with very low occupancy during the day. A detailed look at this cluster in Figure 6.8b shows drastic jumps in the morning and in the evening. Parking was not allowed in these segments for most of the parking spaces between 7 am and 6 pm/7 pm. It is unclear whether the remaining legal parking space was really available most of the time or some sensor failure occurred. The other three clusters contained parking segments with similar level of occupancy. For example, cluster 4 contained all parking segments with high parking occupancy throughout the day (Figure 6.8c).

The clustering results showed that a very diverse parking occupancy daily pattern existed for the different parking segments, regarding both the shape and the magnitude. As the cluster validation measures are just indications for the optimal cluster size, changing the cluster sizes led to different relevant results. Depending on the clustering settings (not only the cluster size, but also, for example, the clustering or linkage method), different levels of detail can be identified, and different insights into the parking occupancy patterns can be revealed.

6.4 Spatial relations in parking availability

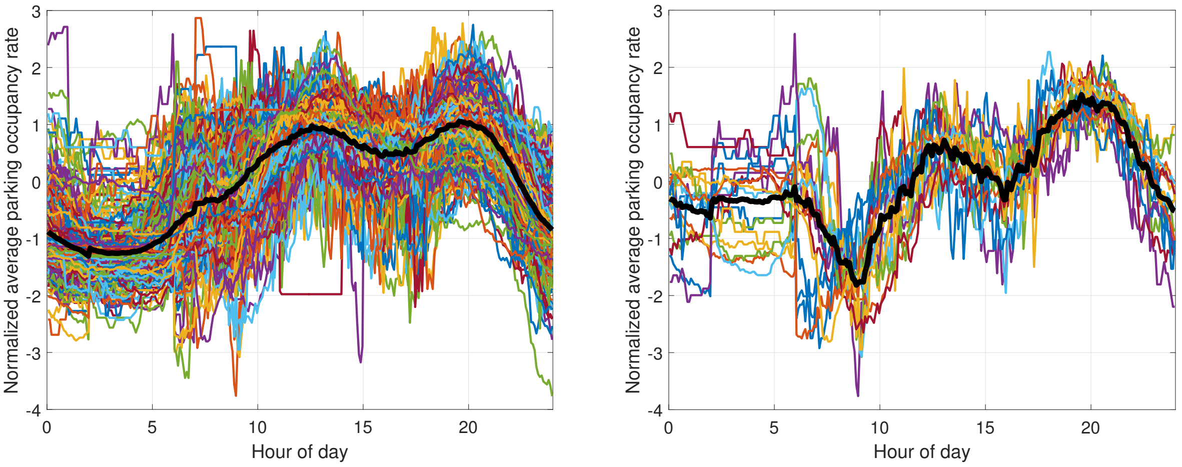
To assess the spatial relations of parking availability in the San Francisco dataset, the spatial mean absolute difference MAD_s at different distances (also called *madogram* in geo-statistics) was used:

$$\mathrm{MAD}_s(\boldsymbol{a}(t),d_c) = \frac{\sum\limits_{|d(i,j)-d_c|<\Delta, i\neq j} |a_i(t) - a_j(t)|}{\sum\limits_{|d(i,j)-d_c|<\Delta, i\neq j} 1} \tag{6.2}$$



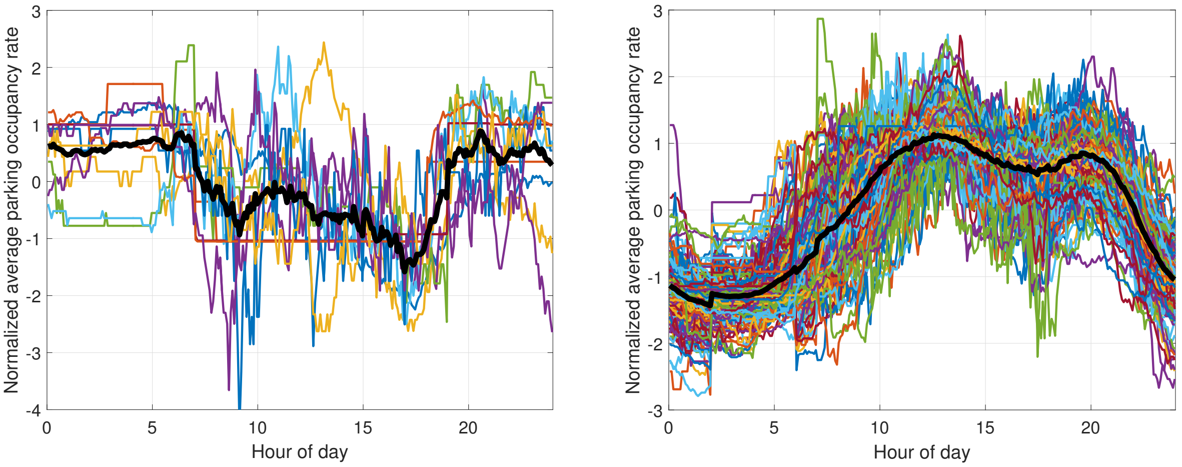
(a) Cluster averages for two clusters

(b) Cluster averages for six clusters



(c) Cluster 2 for clustering with two clusters

(d) Cluster 5 for clustering with six clusters



(e) Cluster 1 for clustering with two clusters

(f) Cluster 4 for clustering with six clusters

Figure 6.6: Clustering results for normalized average parking occupancy with Pearson correlation. (a) and (b) show the cluster averages for two and six clusters, (c)-(f) show exemplary clusters with the average daily patterns for the cluster members in different colors and the cluster average as a thick black line.

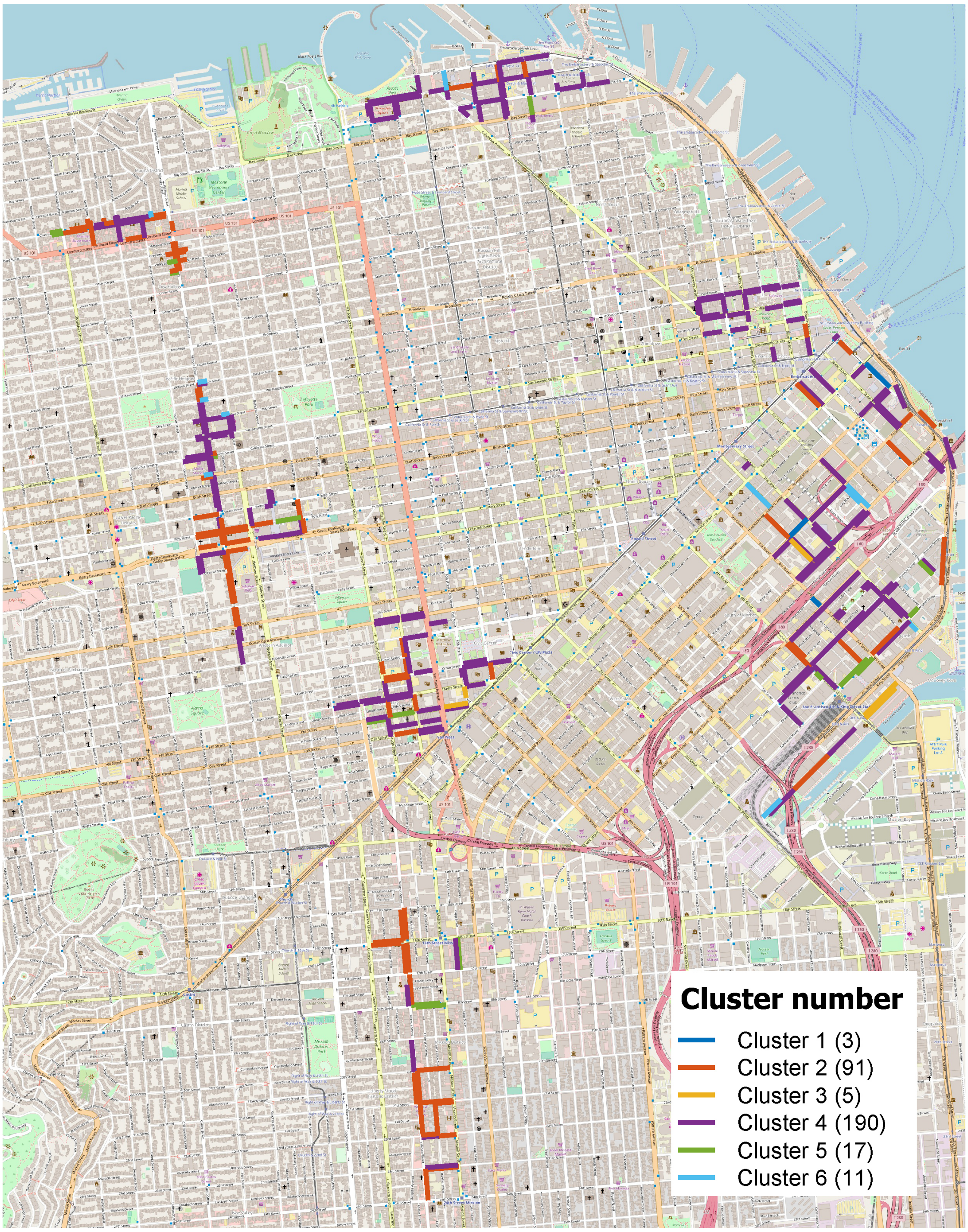
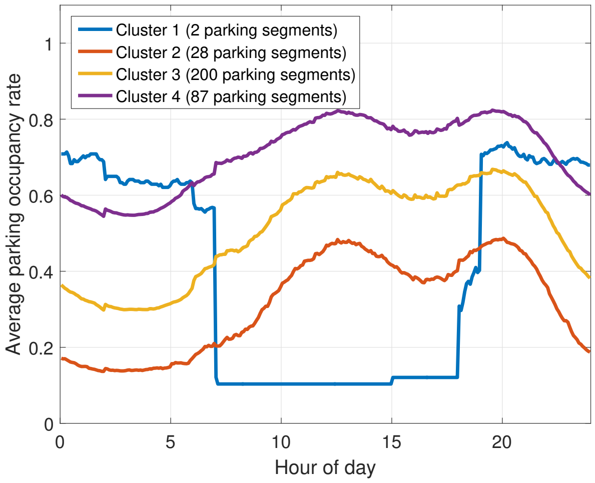
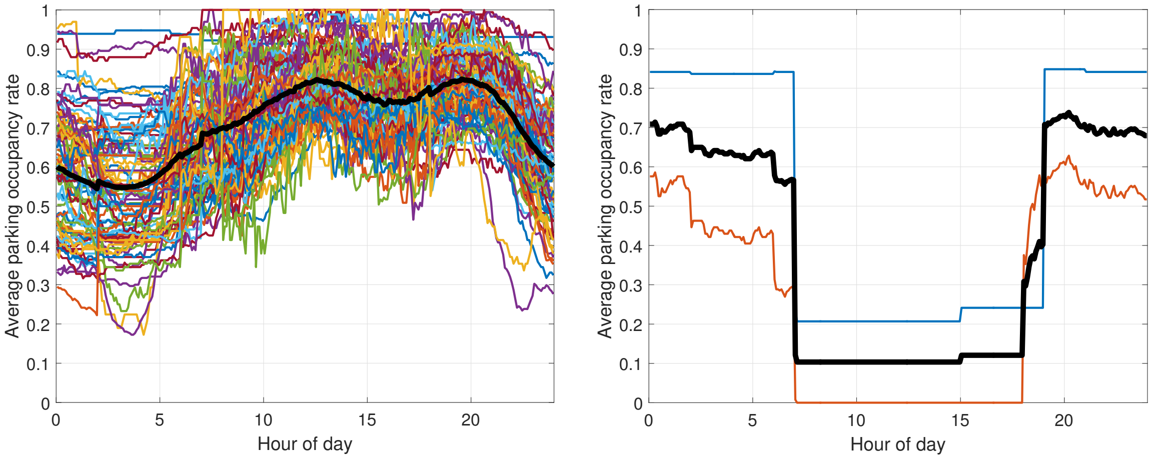


Figure 6.7: Spatial distribution of the clustering result for normalized average parking occupancy with Pearson correlation. The background map is obtained from OpenStreetMap.



(a) Cluster averages for four clusters



(b) Cluster 4 for clustering with four clusters *(c) Cluster 1 for clustering with four clusters*

Figure 6.8: Clustering results for average parking occupancy with Euclidean distance. (a) shows the cluster averages for four clusters, (b) and (c) show the average daily patterns for the cluster members in different colors and the cluster average as a thick black line.

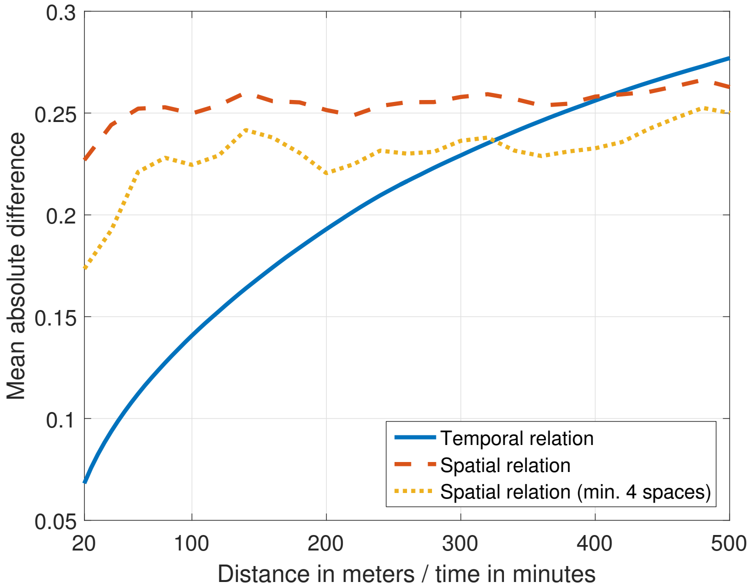


Figure 6.9: Comparison of mean absolute difference (MAD) for spatial relation of the parking availability rate (differences in availability at the same time at different distances) and for the temporal relation (differences in availability at the same parking segment for different times).

where $\boldsymbol{a}(t)$ is the vector of parking availability measurements in different locations at time t, d_c is the considered distance for similarity calculation, $d(i,j)$ is the distance between the locations of the measurements i and j, and Δ is the distance buffer (20 meters in this evaluation). The sums were computed over all parking segment location indices i and j.

The results of this measure, averaged over all time steps in the dataset, are shown in Figure 6.9 with a direct comparison with the temporal mean absolute difference:

$$\mathrm{MAD}_t(\boldsymbol{a}(t+t_c),\boldsymbol{a}(t)) = \frac{\sum\limits_i |a_i(t+t_c) - a_i(t)|}{\sum\limits_i 1} \tag{6.3}$$

where t_c is the considered interval. The spatial mean absolute difference showed a significant increase only at short distances of less than 100 meters. It hardly increased for longer distances. This means that only for short distances, e.g. in the case of opposite parking lanes, there was some correlation. Restricting the evaluation to parking segments with at least four parking spaces led to some reduction in MAD_s. This can be explained by the fact that parking availability rate changes drastically in small parking lanes, when the availability of only one parking space changes (e.g. change by 50 % in a parking lane with two parking stalls). In comparison with the temporal MAD_t, the MAD_s was much higher in most cases. For example, the MAD_s value at 20 meters was reached only after more than 5 hours for the MAD_t. This means that, on average, the similarity of availability rates was higher, even after multiple hours at the same street segment, than at adjacent streets at the same time.

6.5 Modelling of crowdsensing based on downsampling for probe vehicles and mobile apps

Since large scale parking data from probe vehicles would be very expensive to obtain, the crowdsensing of parking availability was simulated by a downsampling of the SFpark dataset according

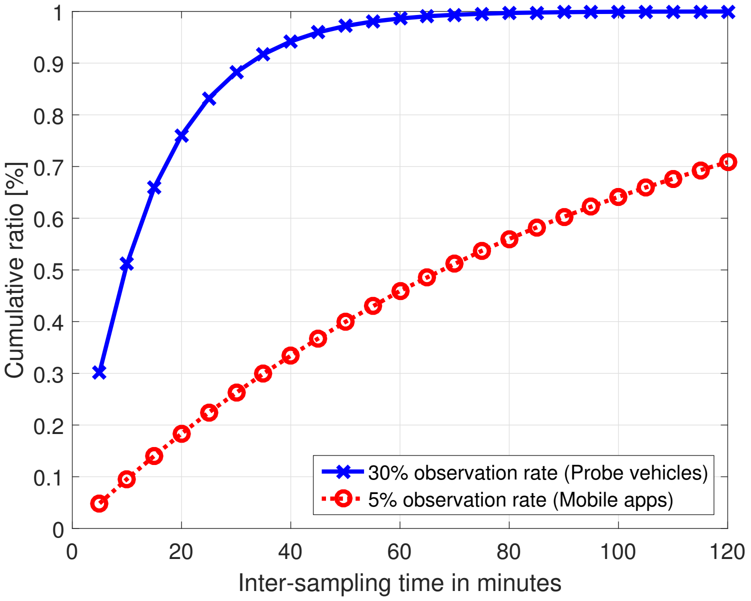


Figure 6.10: Cumulative distribution plot of inter-sampling times between two subsequent observations on a parking segment for 5 % (mobile apps) and 30 % observation rate (probe vehicles).

to models for the sensor coverage. As crowdsensing observations have an irregular temporal resolution, the original SFpark dataset was downsampled with a random choice of observations. In particular, an *observation rate* was estimated for taxis as probe vehicles and for an exemplary smartphone application. An observation rate of $k\,\%$ means that k percent of parking segments were observed every five minutes, or that each parking segment had a $k\,\%$ chance of being observed. Based on this chance, a subset of observations was extracted from the complete dataset according to the model assumptions described below.

6.5.1 Scenario based on probe vehicles

Crowdsensing probe vehicles observe the parking availability on parking lanes with sensors while driving. As discussed in Section 3.1.1, the number of empty spaces per parking segment is obtained by the detection of gaps between parked cars, cross-checked with a parking legality map to ignore driveways. Thus, at every passing of such a probe vehicle, a parking segment is scanned.

A result of the taxi trajectory analysis by Mathur et al. (2010) was that 300 taxis can observe, on average, 80 % of the parking segments in San Francisco downtown with an inter-sampling time (i.e. interval between two subsequent taxis) of up to 25 minutes. This result could be translated to an observation rate r_0 at every five minutes: the chance of missing observations in all t 5-minute time steps is $(1 - r_0)^t$. The probability of observing a parking segment at least once within t time steps corresponds to the complement of not observing it within that period, i.e. $1 - (1 - r_0)^t$. Based on the result of Mathur et al. (2010), 80 % of the parking segments were observed at least once within five time steps (25 minutes), thus $1 - (1 - r_0)^5 = 80\,\%$. Solving this equation results in an observation rate of $r_0 \approx 27.5\,\%$, or roughly 30 %. This calculation is based on the assumption that all parking segments have the same chance of being visited. The cumulative distribution of the observation intervals for this observation rate is shown in Figure 6.10.

6.5.2 Scenario based on mobile apps

As discussed in Section 3.1.1, there are several approaches for mobile applications that could be used to collect parking-related information. In this evaluation, a mobile app was modeled that requires explicit feedback from the user to enter the number of empty spaces for a parking segment, as described by Chen et al. (2012). The user inserts this number into the app after every parking and before every leaving event. This information is even more detailed than most automatic approaches, which just detect the parking and leaving events of the user itself.

To quantify the observation rate for this case, several assumptions were necessary, since the number of observations per parking segment depends on the number of parking/leaving events and on the percentage of drivers using this app:

Number of parking/leaving events

The average number of occupied spaces per parking segment is 3.7 in the evaluation dataset from San Francisco. San Francisco County Transportation Authority (2009) reported that the average parking duration was between 0.7 and 1.7 hours in the area of San Francisco. As an optimistic case, an average parking duration of 30 minutes is assumed in the downtown area, and therefore both 7.4 parking and 7.4 leaving events per hour, i.e. about 1.2 ($\approx 14.8/60 \cdot 5$) parking/leaving events every five minutes.

Percentage of app users

An optimistic user rate of 4 % of all drivers is assumed. For this number, the popular ride service app Uber[4] was taken as reference, which had about the same rate of monthly app users in the USA in June 2015 (about 8 million app users on a driver population of about 214 million licenses (Lella et al., 2015)). In addition, it was optimistically assumed that those drivers use the app for every parking and leaving event.

Despite these optimistic estimates, this scenario corresponds to an observation rate of about 5 % ($\approx 1.2 \cdot 0.04$) for every 5-minutes interval. Interestingly, a user rate of at least 24 % would be necessary to reach the same observation rate as estimated for the 300 taxis as probe vehicles (30 %). Note that the individual capacity per parking segment certainly affected the number of parking events, which was neglected in this estimation.

6.6 Modelling of probe-vehicle-based crowdsensing from taxi GPS trajectories

To improve the quality of the crowdsensing model, observations were simulated based on real taxi GPS trajectories and parking measurements from static sensors in the same area. By matching both datasets by time of day, day of the week, and location, it was assumed that taxis driving at these times would have observed parking availability in the corresponding streets as measured by the static sensors. First, an overview of the processing steps and a description of the taxi trajectories is shown in Subsection 6.6.1. After map matching and filtering the trajectories (see Subsection 6.6.2), the parking segment visits of the taxis were aggregated to a median week. Some statistical evaluations of the spatial and temporal coverage of the road network by the taxi are presented in Subsection 6.6.3. In Subsection 6.6.4, the daily patterns of taxi activities and parking fluctuations are directly compared. The simulation of parking crowdsensing, based on the combination of parking and taxi datasets, is presented in Subsection 6.6.5.

---

[4] `https://www.uber.com`

6.6.1 Processing overview and description of taxi trajectory dataset

An overview of the processing steps for the parking data and the taxi GPS trajectories is illustrated in Figure 6.11. The parking data from the static sensors were cleaned first, as already described in Section 6.1. Based on the geometry information of the parking lanes, the lanes were matched to the road network from OpenStreetMap. For the taxi data processing, the trajectories needed to be map matched in a more sophisticated way because of a low record frequency. A filtering was applied for irregular records and to limit the time frame. A downsampling of the parking segment visits was then necessary, as there were many multi-lane roads and current systems allow only the scanning of adjacent lanes. As the taxi dataset contained some periods of systematically missing periods, a median week was computed in the aggregation step. Finally, the crowdsensed parking availability was determined from the combination of parking and taxi datasets. A measurement of parking availability was added to the parking crowdsensing dataset for every time step a taxi visited the parking segment.

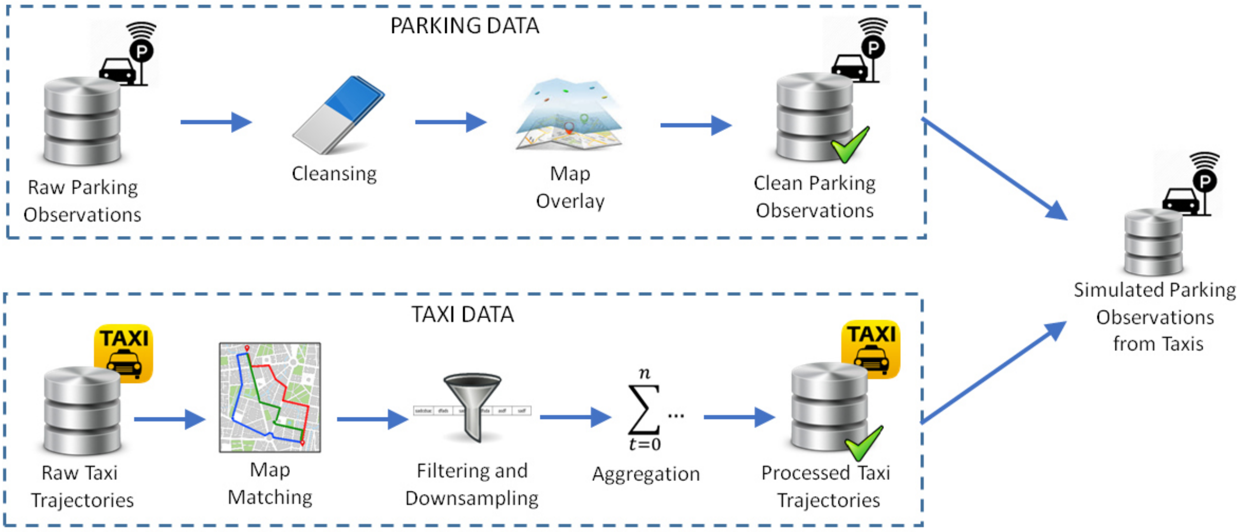


Figure 6.11: Overview of the processing pipeline for simulation of the parking crowdsensing with taxis based on static sensor parking data and taxi GPS trajectories.

The taxi trajectory dataset was collected in the *Cabspotting* project, which had the goal of extracting the socio-economic properties of city districts from the movement data, and is publicly available (Piorkowski et al., 2009). All trips of 536 taxis from the *Yellow Cab* company in San Francisco (US) were recorded with GPS receivers over 25 days from May 17th, 2008 until June 10th, 2008. Longitude and latitude, timestamp, taxi identifier, and passenger occupancy status (occupied / empty) were periodically stored and transferred to a central server. In total, the dataset consists of 11,219,955 GPS points, with a median time gap of 60 seconds between subsequent GPS measurements. About 86 % of all records showed a time gap between 30 and 120 seconds. An example of the activity of one taxi in the center of San Francisco is shown in Figure 6.12.

6.6.2 Taxi GPS trajectory processing

Map matching of GPS trajectories

To identify when a taxi passed a parking segment, the trajectories were matched to the road network from *OpenStreetMap*. In particular, since the record frequency was low and the taxi might have passed several road segments between subsequent GPS measurements, the most probable

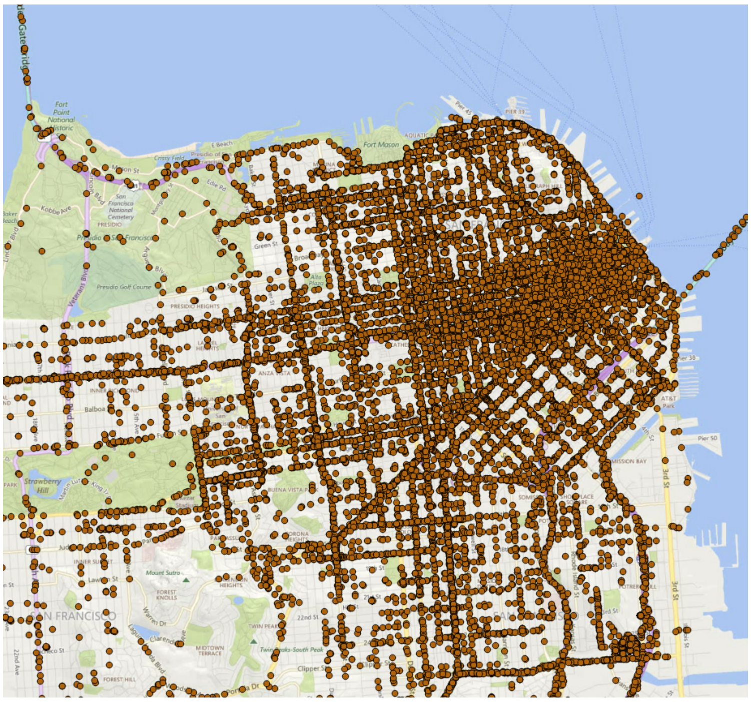


Figure 6.12: Example for the GPS trajectory of one taxi over the complete observation period.

route had to be estimated. For this, the map matching approach of Axer et al. (2015), based on Lou et al. (2009), was applied to the taxi trajectories[5].

This approach consisted of three steps: a splitting step to reduce the length of a trajectory, a candidate search step to determine the possible road segments near the GPS points, and a global optimization step to identify the most likely route connecting the candidates. In the splitting step, the trajectory of a taxi was split into shorter sequences (called *trips* in the following) whenever the occupancy status of the taxi changed, or when there was a time gap of more than three minutes. Trips with unrealistic speeds between two subsequent GPS points and short trips with less than five GPS points were removed from the dataset. The candidate search step identified all road segments in the vicinity of the GPS points. All possible combinations of candidates for a trip of GPS points were determined and connected via a shortest path calculation. For each combination, a score was calculated in the global optimization step, based on spatial and temporal criteria and assuming that the taxi did not take unnecessary detours. The route with the best score was then selected for further processing.

For the taxi trajectory dataset, 8,839,942 GPS points (78.8 % of the points) were successfully matched to the road network. Thus, all following analyses based on this map matching represent a lower bound of the taxi coverage.

Filtering of dataset and downsampling

The map-matched dataset was filtered temporally and spatially to obtain a consistent dataset. First, the dataset was limited to a three-week period (May 18th - June 7th) to represent every day of the week equally. Furthermore, trips by taxis that were inactive for more than a week were

---

[5] The map matching computations were performed by Steffen Axer from the Institute of Transportation and Urban Engineering at TU Braunschweig.

removed, reducing the number of taxis to 486. Spatially, movements were considered for further analyses only next to the parking segments, where parking data was also available.

In addition, some of the visits of road segments were filtered on multi-lane roads, since most current probe vehicles are able to monitor only the adjacent lanes. Therefore, if a taxi is not driving next to the parking lane, no parking information can be obtained. As the GPS trajectories were too inaccurate to determine the lane choice of the taxis, the choice was guessed, based on the number of possible lanes. The number of lanes was contained in OpenStreetMap for 52 % of the road segments in this investigation area. If the number of lanes was even on a two-way road, it was assumed that half of the lanes belong to each of the driving directions. In cases of uneven lane count or missing information, the number of lanes per driving direction was extracted manually from *Google StreetView*. For the following evaluations, two cases were distinguished:

- **Pessimistic case:** the lane choice of the taxi drivers was assumed to be random. Thus, only 1/#(lanes) of the visits in the corresponding direction were next to the parking lane. This subset was chosen randomly from the set of visits.
- **Optimistic case:** taxi drivers might receive a reward for driving next to the parking lanes. Therefore, nearly all visits could be considered, except in the case of one-way roads with multiple lanes and parking on both sides. Since in this case observing both sides is not possible, only half of the visits were considered randomly.

6.6.3 Characteristics and aggregation of taxi coverage

Average time gap as measure for crowdsensing coverage

To measure the spatio-temporal coverage of the taxis as probe vehicles, the taxi visits next to each parking segment had to be examined. In the study by Mathur et al. (2010), the average time between two subsequent taxi visits $\bar{T_s}$ (also called *average inter-sampling time*) at a location was used, defined as:

$$\bar{T_s} = \frac{1}{N_{taxi}-1} \sum_{i=1}^{N_{taxi}-1} (t_{i+1} - t_i) = \frac{t_{N_{taxi}} - t_1}{N_{taxi}-1} \tag{6.4}$$

where t_i are the ordered timestamps of taxi visits and N_{taxi} is the total number of visits at a parking segment in the investigated period.

However, the average inter-sampling time does not consider if the intervals were irregular. In Figure 6.13, two examples are illustrated to show this limitation of the measure. In the upper example, the taxis arrive in the same interval of 30 minutes. Therefore, the average inter-sampling time $\bar{T_s}$ is 30 minutes. If the same number of taxis arrive at irregular intervals (lower example in Figure 6.13), the average inter-sampling time is not affected. However, in a crowdsensing scenario, the upper example should be favored over the lower example.

As a more precise measure, the *time gap* between an instant of time t, when a parking change could happen, and the time of the next taxi visit $t_{next}(t)$ is proposed in this thesis. To assess a complete observation period, the time gaps are averaged over all instants of time in that period:

$$\bar{T_g} = \frac{1}{t_E - t_0} \int_{t=t_0}^{t_E} (t_{next}(t) - t)dt \approx \frac{1}{N_{steps}} \sum_{i=0}^{N_{steps}-1} (t_{next}(t_0 + i \cdot \Delta t) - (t_0 + i \cdot \Delta t)) \tag{6.5}$$

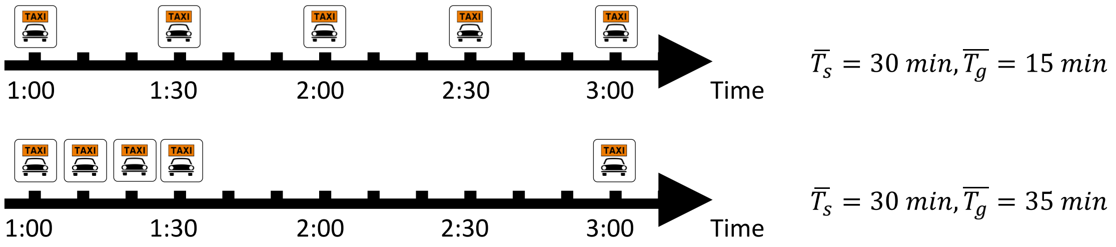

TAXI
1:00
1:30
2:00
2:30
3:00
Time
$\overline{T_s} = 30\,min, \overline{T_g} = 15\,min$
$\overline{T_s} = 30\,min, \overline{T_g} = 35\,min$

Figure 6.13: Examples for regularly and irregularly sampled taxi visits with values for their average inter-sampling time and average time gap.

where $t_{next}(t)$ is the timestamp of the next taxi visit after time t, and N_{steps} is the number of time steps between start time t_0 and end time t_E with interval Δt. Note that, for periodically sampled measurements, the mean time gap $\bar{T_g}$ is half of the inter-sampling time $\bar{T_s}$, but increases for irregular sampling. In the example in Figure 6.13, the average time gap well incorporates the regularity of the intervals. While the upper example has an average time gap $\bar{T_g}$ of 15 minutes, $\bar{T_g}$ increases to 35 minutes for the lower example.

As an additional example of irregular sampling, the measurements in Chapter 4 (see Figure 4.10 on page 55) are considered. The mean time gaps are about 41 minutes for ‘Osterstraße’ and 56 minutes for ‘Fliederstraße’[6]. If the measurements were sampled perfectly at the same interval, the mean time gap would be about 35 minutes and 39 minutes, respectively. However, the inter-sampling time is independent of the sampling, with 69 minutes (‘Osterstraße’) and 77 minutes (‘Fliederstraße’).

Evaluation of taxi visit time gaps

For the evaluation of the spatio-temporal coverage of the taxis, the time gap measure was used with an interval of $\Delta t = 5\,\text{min}$ in the following computations. For the pessimistic case, the cumulative distribution function of the time gap in Figure 6.14 shows that the time gap was less than an hour in about 80 % of the time instants for all parking segments. While nearly 60 % of time instants had a time gap of up to 20 minutes, and thus any changes in parking occupancy could quickly be detected, more than 10 % of the time instants were not yet observed, even after 100 minutes. The curve for the optimistic case showed very similar behavior as for the pessimistic case, with differences of only a few percentage points. For very small time gaps, the curves were similar to the modeling based on a constant observation rate of 30 %[7] (see Section 6.5). However, for time gaps larger than about 10 minutes, a clear difference is visible. While nearly all cases had a time gap of less than one hour for the 30 % observation rate, the curves for the modeling based on the taxi trajectories were much flatter. This difference was presumably caused by the spatio-temporal heterogeneity of the taxi coverage, since during the night hours and in small streets fewer taxis were active. Thus, the model with constant observation rate clearly overestimated the spatio-temporal coverage of the observation area.

Temporal characteristics and aggregation

As the time gaps were revealed to be quite heterogeneous, the temporal dependencies of the time

---

[6] In ‘Fliederstraße’, one measurement is missing because of system failure.

[7] Note that the observation rate of 30 % was modeled for a fleet size of 300 taxis, based on the basic trajectory evaluation by Mathur et al. (2010), while the taxi trajectory results are based on 486 taxis.

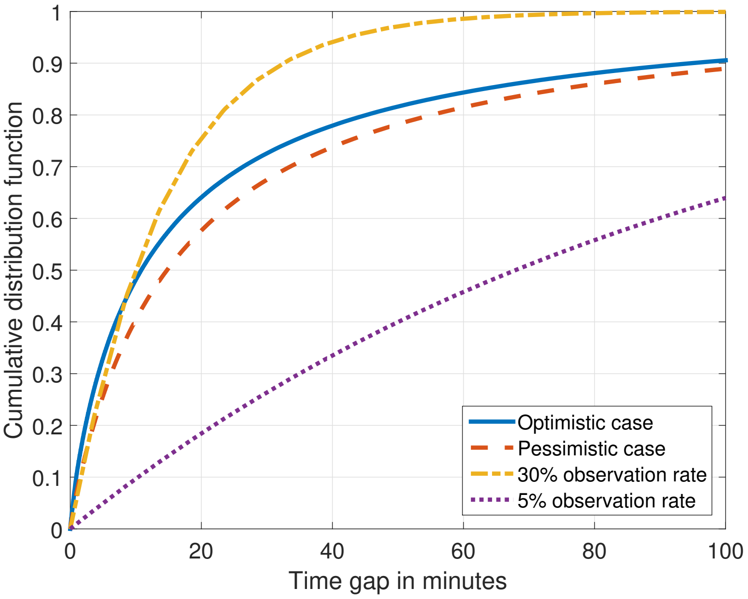
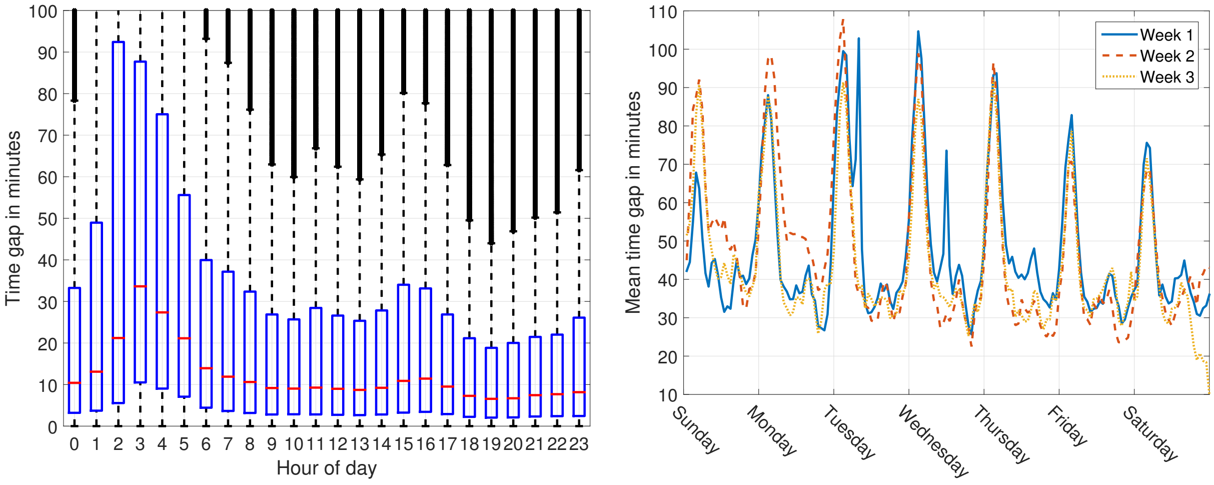


Figure 6.14: Cumulative distribution function of the time gap for the optimistic and pessimistic observation cases.

gap were investigated in more detail. A boxplot for the time gaps of the optimistic case at different hours of the day is illustrated in Figure 6.15a. The highest median values (red bars) were observed during night hours around 4 am. Throughout the day, the median was at about 10 minutes, with the lowest values around noon and in the evening hours. The difference between the upper and lower quartile (blue boxes) was also highest during the night and smallest in the evening. The outliers appear as a solid black line due to the magnitude of the data considered. In conclusion, the coverage of taxis as probe vehicles was strongly dependent on the hour of the day, and consequently the time of the day needs to be taken into account for further evaluations.



(a) Boxplot of time gaps

(b) Mean time gaps for each week

Figure 6.15: Temporal characteristics of time gaps for the optimistic case (a) as a boxplot for different hour of day and (b) mean values for each week. Note that the outliers in the boxplot appear as a solid black line, due to the magnitude of considered data.

The mean time gap also showed a regular pattern for every week (see Figure 6.15b). However, there were four cases of strongly irregular patterns. There were two irregular peaks on Tuesday and Wednesday morning in the first week, because of systematically missing values in the dataset for about 80 and 50 minutes, respectively. It could be assumed that there was an overall problem in the central recording system. On Monday of the second week, there were always larger time gaps than on the two other Mondays. This might be caused by the fact that this Monday was Memorial Day, a US holiday, and thus the travel behavior of taxi passengers differed. Finally, the time gaps were lower at the end of the third week, which can be explained by a computational issue, as there were no 'next taxi visits' anymore for some parking segments. Since all three weeks showed some irregular behavior, a median weekly pattern was calculated: for each hour of the week, the median week of the mean time gaps was determined and all taxi activities during these hours were combined to a 'typical' weekly taxi pattern for the following evaluations.

Spatial characteristics

The mean time gaps at different hours of the day showed some differences among the project regions (see Figure 6.16a). Differing from the general pattern, region 1 had very high mean time gaps during the night. These high values were presumably caused by the type of area (tourist area) and low through traffic due to its location close to the waterside. In region 3, the lowest mean time gaps were late in the evening, as there are many restaurants and bars in this area.

The coverage of taxis, and thus the mean time gaps, strongly depended on the road class, as shown in Figure 6.16b. The road class was extracted from OpenStreetMap and describes the priority of the road (see also Section 2.1.2). The most important urban road classes 'Primary', 'Secondary', and 'Tertiary', about 65% of all roads assigned to the parking segments, had a time gap of 11 min, 16 min, and 27 min on average over the day, respectively. The mean values per hour were less than 50 minutes throughout the day. For small roads of the classes 'Unclassified'[8] and 'Service', the mean time gaps were never lower than 50 minutes, even during the most active hours and going up to nearly four hours during the night. However, only 5% of all investigated parking segments belong to these two classes. The higher number of taxi visits on main roads corresponds to the general traffic flow, which is usually higher on main roads than on side roads. For a crowdsensing scenario, a trade-off is needed between a better coverage on small roads and the additional effort to run more probe vehicles.

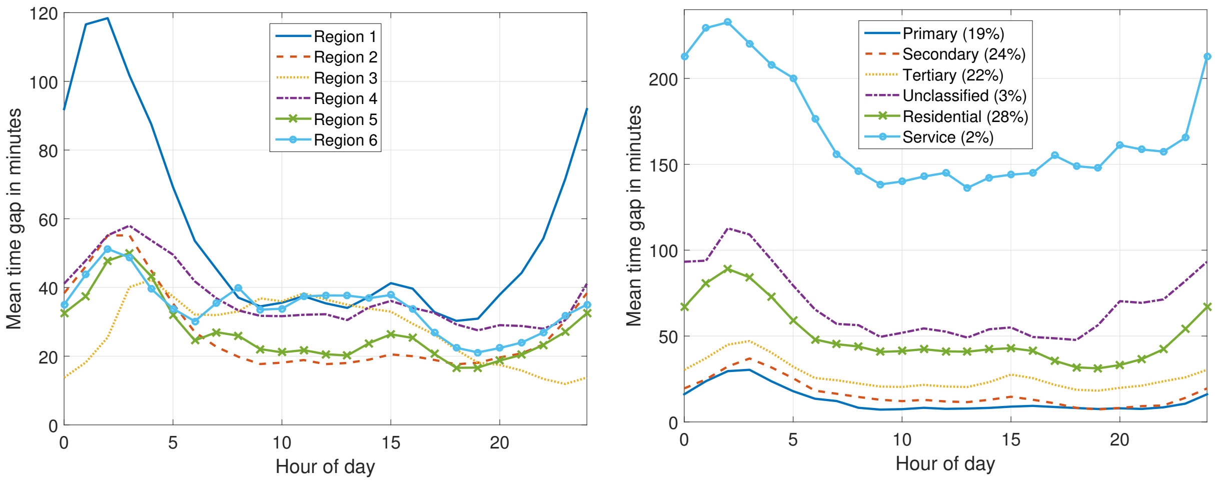
In Figure 6.17, a map of the parking segments, colored according to the mean time gaps, is shown. The mean time gaps were very heterogeneous in each region of the investigation area. Only a few spatial clusters were visible, such as the downtown area in region 2 with low time gaps and region 1 with rather high mean time gaps.

6.6.4 Comparison of parking and taxi daily pattern

The average daily patterns of the parking and taxi datasets over the course of the day are directly compared in Figure 6.18. The taxi activity was represented by the mean ratio of visited parking segments per time step of five minutes. The mean parking changes per time step were used for the parking activity. Note that the changes were calculated based on the number of occupied parking spaces in two subsequent time steps. However, if both parking and leaving events happened within an interval of five minutes, these changes remained unobserved. For example, a parking occupancy difference of zero for a certain parking segment at a given interval could be due either to the absence of parking events or to one or more vehicles leaving the parking lane and the same number of vehicles parking there, within this time interval. Thus, this measure does not represent the actual turnover rate. However, as a relative measure, it still indicates times and locations of higher

---

[8] Note that the road class 'unclassified' is a defined class and does not stand for an unknown class assignment.



(a) per region

(b) per road class

Figure 6.16: Spatial characteristics of time gaps for the optimistic case (a) for the different regions as defined in Figure 6.17 and (b) per road class as specified in OpenStreetMap with relative frequency in brackets. Note that this frequency relates to the number of roads, while the values in Table 6.1 are based on the lengths of the roads.

and lower parking dynamics. In addition, it is important to mention that the distinct peaks at every hour, especially at 6 am and 6 pm, were caused by the change in parking permission: in some cases, parking spaces were already occupied before the start of the parking period, but this information was not communicated by the system[9].

The plot in Figure 6.18 indicates that both parking and taxi activities were lowest during the night around 4 am and increased in the morning hours. However, while the parking activities decreased in the evening hours, the taxis were most active then. Thus, the relationship between parking and taxi activities was investigated in more detail in Chapter 7 to evaluate the quality of parking crowdsensing.

6.6.5 Simulation of parking availability observations

To simulate the parking crowdsensing by the taxis, the observed parking availability data was computed based on the spatio-temporal distribution of taxis for the typical week. For this, the parking availability of the corresponding day of the week and time of day was added to the crowdsensed dataset at every time step when a parking segment was visited by a taxi. In particular, it was considered that the parking availability is observed by a taxi in the same time step, if the time between the parking availability measurement and the next taxi visit was less than 2.5 minutes. If the time to the next taxi visit was between 2.5 and five minutes, the visit was assigned to the subsequent time step. This assignment was computed for every street segment separately over the six weeks of parking data. The results of this simulation were used in the next chapter to compare methods for parking availability estimation.

---

[9] For street cleaning, parking was not allowed during certain periods. However, if cleaning was finished already before the end of a period, parking was allowed. The SFpark system communicated an occupancy and capacity of 0 for the complete periods, although drivers might have parked there by that time.

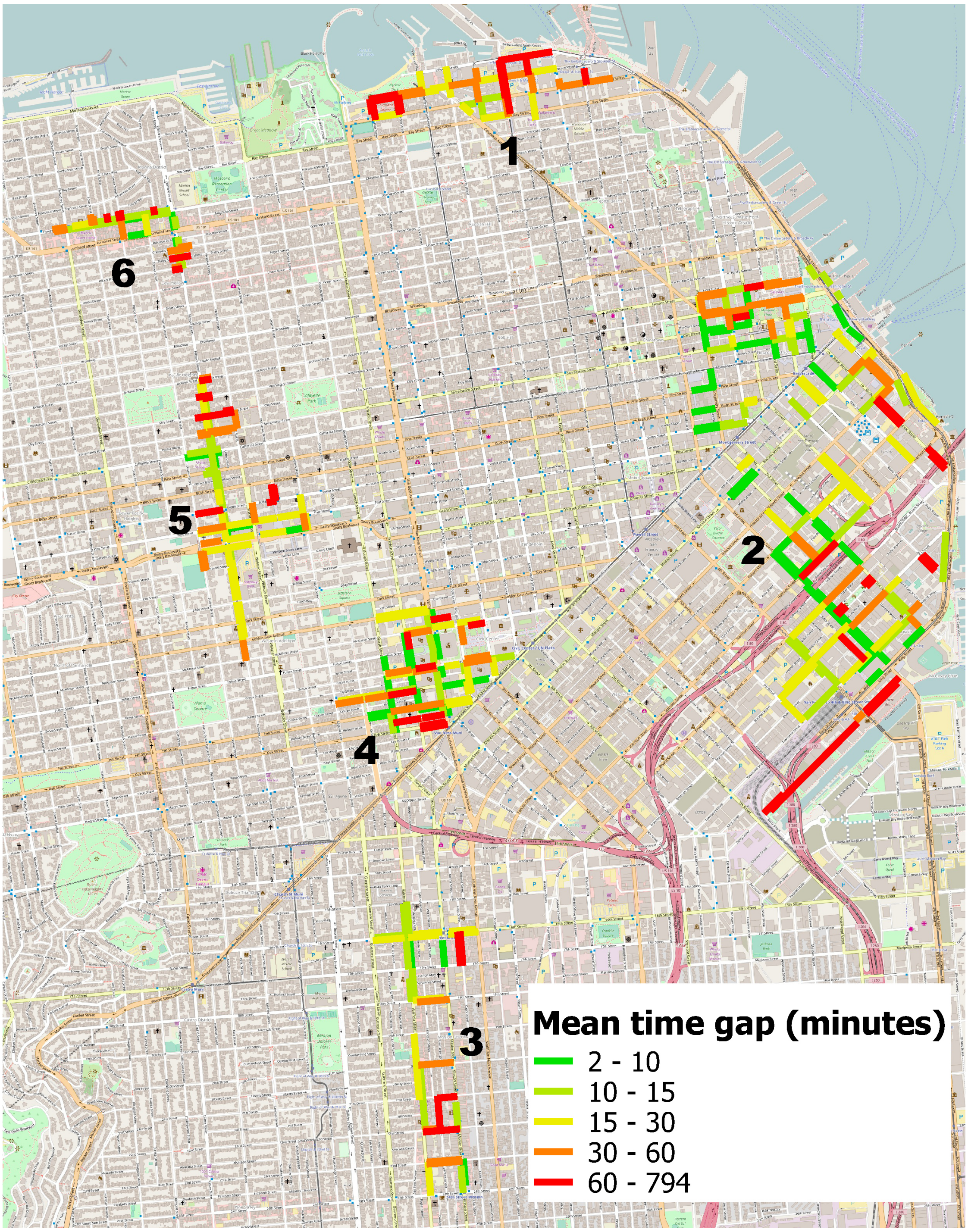


Figure 6.17: Map of the mean time gaps per parking segment for the optimistic case. The background map is taken from OpenStreetMap.

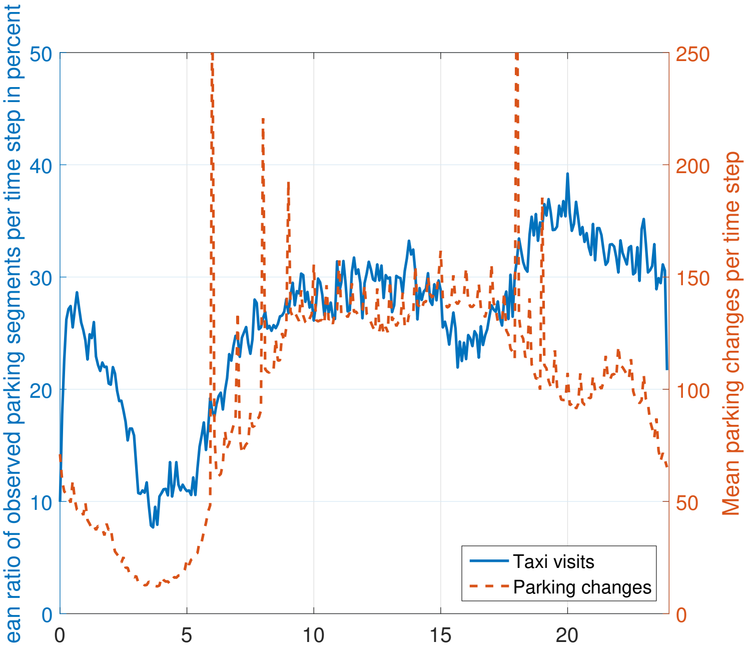


Figure 6.18: Plot of the taxi activity in terms of the mean ratio of observed parking segments compared with the mean parking changes in the parking segments per time step of five minutes.

6.7 Concluding remarks

Spatio-temporal properties of parking

The results of the detailed spatio-temporal analysis of the parking behavior in San Francisco showed that there are both spatial and temporal correlations, but the latter is much more relevant. This result stands in contrast to Rajabioun and Ioannou (2015), who stated that spatial and temporal correlations are about equally important. However, their paper showed some inconsistencies in their evaluations. In the following chapter, concerning the estimation of parking availability, both spatial and temporal aspects are considered in detail.

The analysis of the parking occupancy time series revealed that there are repetitive daily occupancy patterns. Based on this result, a clustering of similar occupancy patterns was computed. In line with the works by Richter et al. (2014) and Zheng et al. (2014), characteristic occupancy patterns were obtained, e.g. with the highest parking demand around noon or in the evening hours. In addition, the results showed that parking segments with similar occupancy patterns are often located nearby.

Limitations of the SFpark dataset

The SFpark dataset represents a valuable data source, since it describes the parking behavior in San Francisco in high temporal and spatial levels of detail. However, the parking sensors cover only metered parking spaces. Areas without parking fees might have an even higher parking demand, as discussed by Millard-Ball et al. (2014). Moreover, San Francisco Municipal Transportation Agency (2014) reported several issues for inaccurate sensor measures, such as early battery degradation, electromagnetic inference, and damages from road construction. Although detailed data cleaning was applied for this thesis, the evaluations still might be affected by some measurement errors.

During the project period, the parking meter prices were changed incrementally and individually per parking segment. This means that the parking behavior of the drivers was influenced by the prices on a parking segment level. In addition, drivers could inform themselves about the parking situation via the smartphone app, and thus head directly to less crowded parking segments.

Simulation of parking crowdsensing

Two approaches to simulate parking crowdsensing were presented, based on a constant observation rate and on a detailed taxi trajectory analysis. It can be assumed that the observation rate method is less accurate, since it does not consider the spatial and temporal heterogeneity of the mobile sensors. Still, it provides an interesting insight into the comparison between vehicular crowdsensing and smartphone crowdsensing. Although the smartphone case was modeled with several optimistic assumptions, it clearly achieved worse results than the probe vehicles.

The analysis of taxi trajectories revealed very interesting results about their coverage. The taxis were more active during the day than during the night, but the specific day distribution depended on the area. For example, in area 3, with many restaurants and bars, the taxi coverage was best at almost midnight. The coverage also strongly depended on the road class. However, the lowest road classes were still visited from time to time.

An elaborate map matching was used to identify the most likely routes of the taxis in the minute between two location measurements. However, due to implausible measurement values and very short trips, the map matching of only 78.8 % of the GPS points was used in the subsequent computations. Thus, the real taxi coverage was probably markedly better.

The taxi trajectory data from the year 2008 was matched to the parking data from 2013, based on time of day and day of the week. It is possible that socio-economic changes, seasonal factors, and the emergence of mobility services such as Uber[10] may have influenced the taxi movement patterns. In addition, the lane choice of the taxi drivers was unknown. Two different lane models were compared. Interestingly, the random lane choice (pessimistic case) achieved almost the same results as the optimistic case, where drivers were assumed to be rewarded for driving in the lane next to parking.

The simulation assumed that the taxis were equipped with sensors to observe the parking availability while driving. The vehicles could either be retrofitted with additional sensors (Mathur et al., 2010) or might already contain suitable sensors. However, in the latter case, cooperation with the car manufacturer might be necessary in order to gain access to the sensor readings.

---

[10] https://www.uber.com/

7 Parking availability estimation and prediction from crowdsensed data

The estimation and prediction of parking availability on the roads is an important assistance for drivers. If they are close to the destination, direct guidance to parking segments with better chances of finding parking is desirable. If drivers are informed about the predicted availability before the start of a trip, they can consider using another mode of transportation, select another destination, or change their departure time. In a crowdsensing setup, the parking availability is not observed continuously in all parking segments. Thus, the availability needs to be estimated.

Three methods to estimate parking availability are evaluated in this chapter. The estimations are computed on a parking segment level, as an estimation per parking space is hardly possible and the information per parking segment is usually sufficient for the guidance of drivers. The SFpark project also provided the availability information on a parking segment level only. In Section 7.1, the performance of spatial interpolation methods is investigated for different taxi fleet sizes. The repetition of the last availability observation in each parking segment, also called persistence method, is evaluated in Section 7.2. Finally, a binary classification approach is described in Section 7.3, estimating whether the parking lane is fully occupied or not. This approach can also be used for the prediction of parking availability. The results of this method are evaluated in great detail for different parameter settings, feature choices, and probability estimation quality. Finally, the results and their implications are discussed in Section 7.4. Most of the results in this chapter are also published in Bock and Sester (2016); Bock et al. (2016a, 2017a); Bock and Di Martino (2017).

7.1 Spatial interpolation of parking availability

One possibility for estimating parking availability in locations that are not crowdsensed in the current time step is spatial interpolation, using adjacent measurements to compute the missing values. For the interpolation of parking availability rates, two methods were compared: inverse distance weighting (IDW) which calculates a weighted average based on the distances (Shepard, 1968),

$$A_{\mathrm{IDW}}(\boldsymbol{x},t) = \begin{cases} A_i(t) & \text{if } d(\boldsymbol{x},\boldsymbol{x}_i) = 0 \text{ for any i} \\ \frac{\sum_{i \in R} w_i(\boldsymbol{x}) \cdot A_i(t)}{\sum_{i \in R} w_i(\boldsymbol{x})} & \text{if } d(\boldsymbol{x},\boldsymbol{x}_i) \neq 0 \text{ for all i} \end{cases} \tag{7.1}$$

with weights

$$w_i(\boldsymbol{x}) = \frac{1}{d(\boldsymbol{x},\boldsymbol{x}_i)^e} \tag{7.2}$$

and unweighted averaging of availability rates from all observed parking segments in the investigated area

$$A_{\mathrm{AVG}}(\boldsymbol{x},t) = \frac{\sum_{i \in R} \cdot A_i(t)}{\sum_{i \in R} 1} \tag{7.3}$$

where $A_i(t)$ is the parking availability rate in parking segment i at time t at the location $\boldsymbol{x}_i$, $\boldsymbol{x}$ is the location of the estimation, R is the set of all observed parking segments, and e is the weighting exponent. The unweighted averaging corresponds to the special case of $e = 0$ in the IDW method.

Additionally, the results were compared against a baseline approach, where a random value between 0 and 1 was guessed for the parking availability.

For evaluation, the parking availability dataset from crowdsensing with taxis (Section 6.6) was used for the complete six-week period with one time step per hour (always half past the full hour). As the parking availability rate changes drastically for any parking event in very small parking segments, only parking segments with at least four parking stalls were considered. For every time instant, the set of parking segments was divided into two sets: parking segments observed by taxis and unobserved parking segments. The Mean Absolute Error (MAE) of the estimated parking availability rate was computed for all unobserved parking segments as quality measure:

$$\mathrm{MAE}(A_{\mathrm{IDW}}) = \frac{1}{N_R \cdot N_t} \sum_t \sum_{i \in R} |A_{\mathrm{IDW}}(\boldsymbol{x}_i, t) - A(\boldsymbol{x}_i, t)| \tag{7.4}$$

where $A(\boldsymbol{x}, t)$ are the elements of the true parking availability rates, N_R is the number of elements in R, and N_t is the number of time steps.

Figure 7.1 shows the interpolation quality for a fleet of between 10 and 486 taxis. For this, out of the full taxi dataset of 486 taxis, a random subset of N taxis (in this investigation 10, 50, 100, 200, 300, and 400 taxis) was drawn. The plot indicates that all interpolation methods were clearly better than the random guess baseline. The more sophisticated Inverse Distance Weighting outperformed the simple average only for more than 100 taxis with $e = 1$. For smaller fleet sizes, the results are about equal. The IDW method with $e = 2$ achieved only somewhat worse results. In total, the difference between IDW and simple average was small. This means that only small benefits could be achieved by weighting nearby parking segments higher than distant ones, and that the overall parking demand in the city dominated. This result corresponds to the spatial correlation result in Figure 6.9 on page 80. Thus, a stronger focus is set on the temporal aspects in the following.

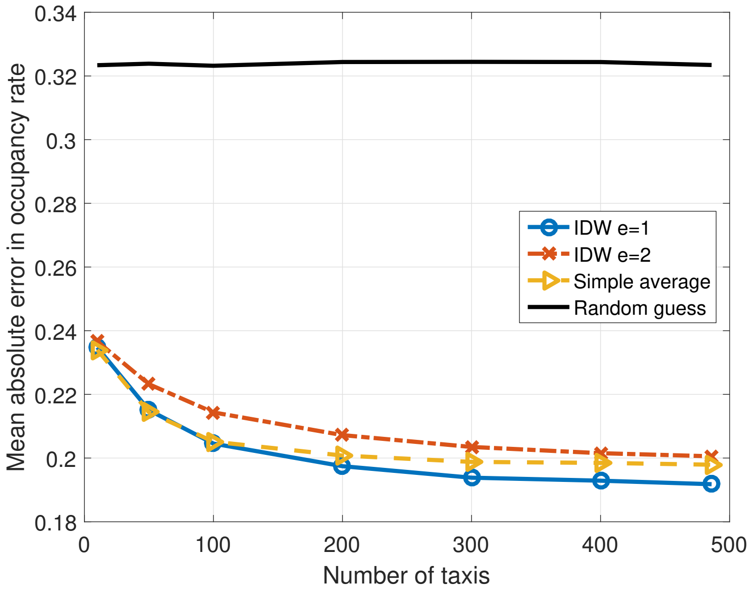


Figure 7.1: Comparison of methods for parking availability interpolation for different taxi fleet sizes.

7.2 Parking availability estimation with persistence method

As the previous results showed a low spatial influence in parking behavior, the temporal behavior is considered in this section. The dataset of crowdsensed parking availability (described in Section 6.6) contained only measurements at the time instants when a taxi visited the parking segment. For missing observations, the parking availability was estimated with the persistence method. This means that the last observed value was repeated until the next measurement was obtained. For evaluation, the estimated parking availability time series was compared to the complete parking data time series from the static sensors. For every time step and parking segment, the difference between both time series was computed. The frequency of the differences in the number of empty parking spaces was counted as a quality measure.

Figure 7.2 shows the count of time steps for every specific error difference. For both the optimistic and pessimistic case[1], crowdsensing achieved the correct number of parking spaces in about 70 % of time instants over all parking segments. If a difference of one parking space between estimated and actual parking availability is tolerated (called *sufficient* taxi coverage in the following), about 90 % of the cases matched. Only a small difference of a few percentage points was observed between the pessimistic and optimistic case. Thus, the evaluations are mostly limited to the pessimistic case (without rewarding the drivers for the lane choice) in the following. As with the comparison of the time gaps in Figure 6.14 on page 87, the spatio-temporal coverage of the model with 30 % observation rate (see Section 6.5) overestimated the true coverage of the taxis. The case with a constant observation rate of 5 % (smartphone application) was still considerably worse than the cases based on the taxi trajectories.

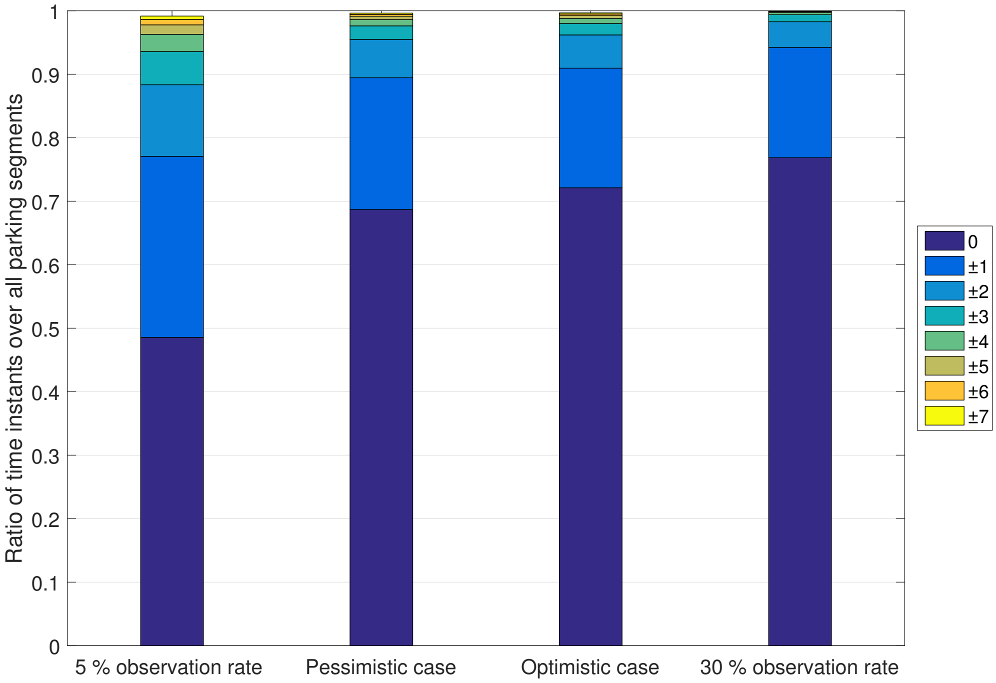


Figure 7.2: Frequency of differences between estimated and actual parking availability for the persistence method in the pessimistic and optimistic cases with 486 taxis and the cases with downsampling of 5 % and 30 % observation rate.

---

[1] These cases refer to the optimistic and pessimistic models of lane choice by taxi drivers as described in Subsection 6.6.2.

To estimate the necessary number of taxis to crowdsense parking availability, the previous computations were repeated for different sizes of taxi subsets. In addition, a baseline as reference was defined, assuming that half of the spaces in a parking segment were available and half occupied. If the capacity of the parking segment was uneven, the baseline counted one more parking space to be occupied than empty. Figure 7.3 summarizes the results of this comparison. Evidently, few taxis clearly outperform the baseline. With increasing fleet size, the quality of parking availability estimation continuously increases. With 300 taxis, sufficient observation coverage is achieved in more than 85 % of cases. Based on these results, the taxi fleet size required can be determined dependent on the quality requirements for such a system.

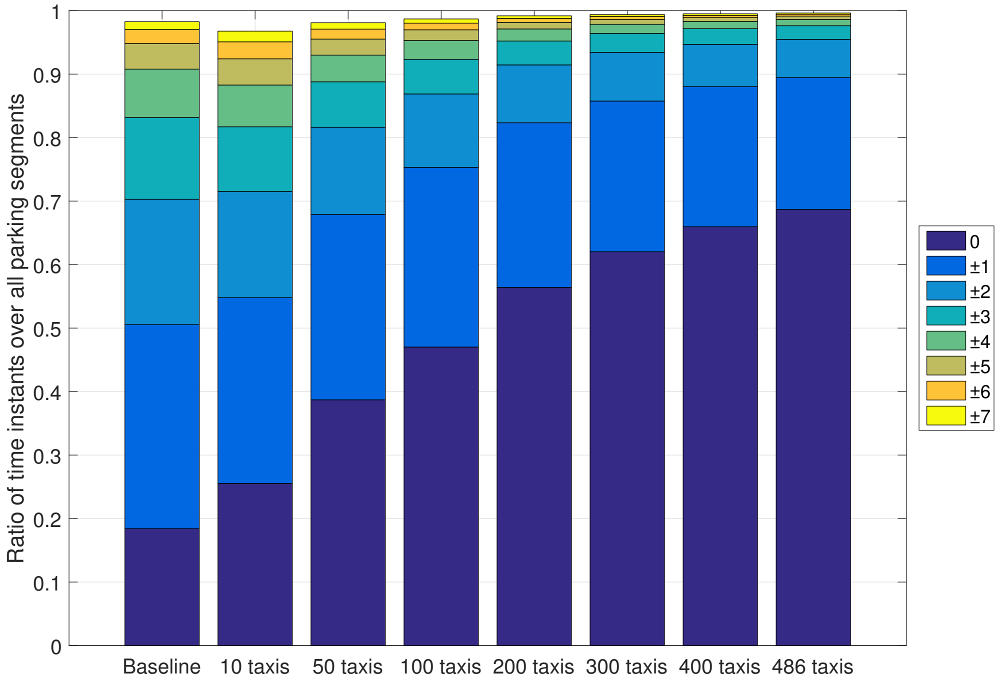


Figure 7.3: Frequency of differences between estimated and actual numbers of empty parking spaces for different taxi fleet sizes in the pessimistic case.

To compare the results with the interpolation results in the previous section, the Mean Absolute Error (MAE) of the parking occupancy rate was evaluated for the estimated (not-observed) instants on parking segments with at least four parking spaces. In Figure 7.4, the MAE is presented for different fleet sizes in the pessimistic case. The error for the persistence method decreased with increasing number of taxis, similarly to the result in Figure 7.3. In particular, the error reduced from 0.230 for 10 taxis to 0.076 for 486 taxis. The persistence method was better than the spatial interpolation for all taxi fleet sizes. Particularly for large fleet sizes, there was a large difference between both methods.

Figure 7.5 shows an evaluation of sufficient coverage for different road classes and time of day in the pessimistic case. It is apparent that there was strong dependence on the road class, in line with the evaluation of the taxi trajectories in Subsection 6.6.3. The coverage was best for all road classes during night hours. This is remarkable, since taxi activity was lowest in this period, but was overcompensated for by the low parking activity. The quality in primary roads was mostly lower than in secondary roads, because primary roads were usually multi-lane roads. Therefore, only a fraction of taxis on the road segment scanned the parking availability in the pessimistic case.

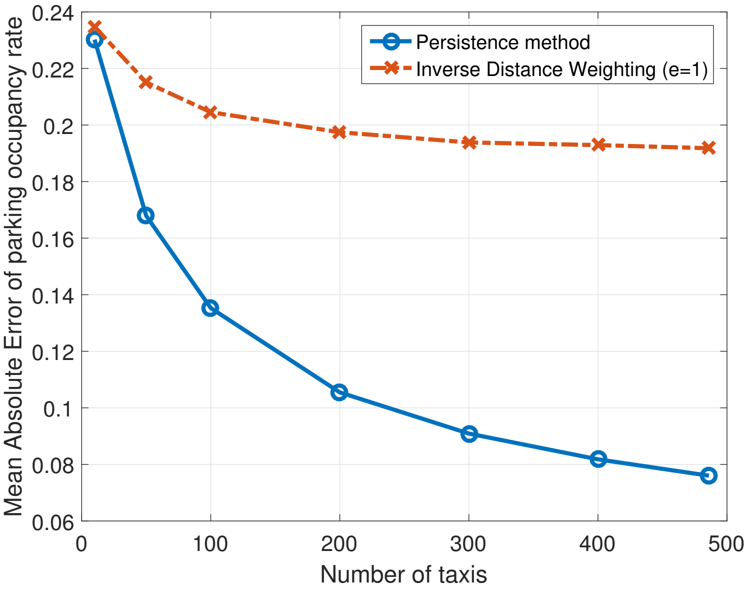


Figure 7.4: Mean Absolute Error (MAE) of the parking occupancy rate estimation for different fleet sizes in the pessimistic case comparing the persistence and interpolation (IDW) methods.

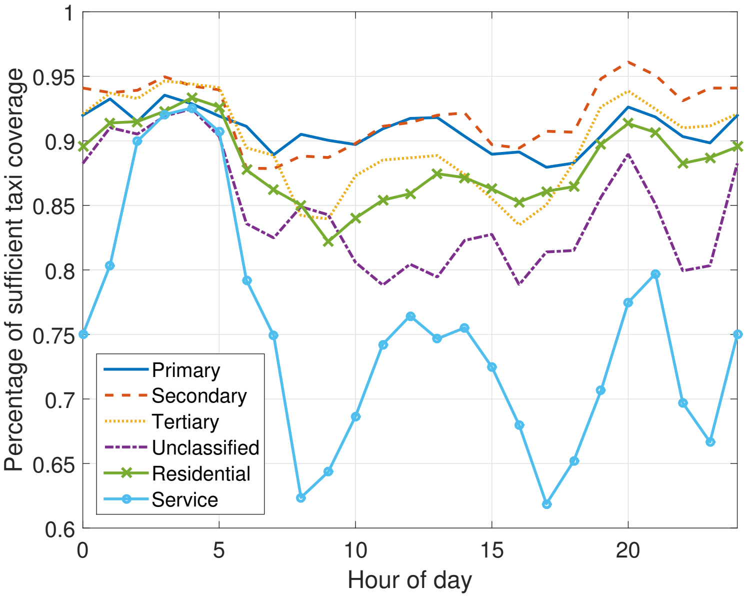


Figure 7.5: Temporal dependence of the percentage of sufficient road coverage (difference is 0 or ± 1) per road class as specified in OpenStreetMap for the pessimistic case.

7.3 Estimation and prediction of parking availability based on binary classification

To improve the parking availability estimation, machine learning methods can be applied. In this section, the estimation was modeled as a binary classification, since most drivers are interested in knowing whether at least one parking space is available at the destination. In addition, drivers benefit from an estimation of the reliability of this information. The estimation method can also be used to predict parking availability in the near future. In this approach, it is taken into account that crowdsensed parking data does not have a regular frequency, but parking is observed at highly irregular intervals. Thus, established time series approaches could not be directly applied to this setting. In the following, the term *estimation* refers to the parking availability at the present time, using crowdsensed data up to this point in time, while the *prediction* has the goal of computing the parking availability in the near future.

7.3.1 Binary classification approach

For the classification task, a random forest classifier was used (see Section 2.2.3). The parking availability was estimated for the present time and predicted for a time horizon of up to 55 minutes in the future. For the prediction, this means that there is the prediction horizon plus a time gap between the last observation of the parking segment and the present time.

Four types of features were defined for the classification: recent observations of the availability in the parking segment, observations of the availability in the adjacent parking segments, time information, and road information:

- **Recent observations** were defined as the last n observations ($n = 5$ in this case) before the present time, with a count and ratio of empty parking spaces. In addition, the time difference between observation and the present time was used, as crowdsensed observations are not uniformly sampled and so older information might be less relevant. If parking spaces were not provided at that time, e.g. because of temporal parking restrictions, negative default values are set (availability ratio to -0.5, number of empty parking spaces to -1) to distinguish this case from the case of fully parked streets and therefore also zero empty parking spaces.
- **Observations in adjacent roads** were determined in predefined radii in beeline distance (100 m, 200 m, 400 m, and 800 m) and time intervals (0 - 5 min, 10 - 15 min, 20 - 30 min, and 35 - 60 min before investigated time). For each combination of these parameters, the total number of empty parking spaces and the rate of empty parking spaces were calculated.
- **Time information** consisted of the minute of the day and the day of the week. These features were included to consider the typical temporal patterns of parking availability, as presented in Section 6.2.
- **Road information**, containing the average geographic location (midpoint) of the parking segment in UTM coordinates and the number of parking spaces provided (parking capacity) at that time instant. The number of parking spaces provided was included to reduce the problem to a pure estimation/prediction of parking space availability and to avoid a combined learning of availability and capacity changes (e.g. periodic parking prohibition times).

As a baseline, the estimation was computed based on the last observation of the crowdsensing taxis (persistence method). The baseline can be computed using either the availability rate or the number of available parking spaces. If the availability rate r or the number of available parking spaces p was smaller or equal to a threshold t, the classification result was ‘full’. The different precision and recall values were obtained by variation of the threshold t. Intuitively,

$t = 0$ represents the simplest solution: if a parking segment was ‘full’ in the past observation, the estimation is also ‘full’.

7.3.2 Results of binary classification estimation and prediction

In the following, results for the estimation and prediction based on taxi crowdsensing are compared with the baseline solution, as well as with the scenario of static sensors, and the impact of the taxi fleet size is evaluated. In addition, an evaluation of the feature importance is presented. Finally, the quality of the classification score as a probability estimation is described. For the following evaluations, the SFpark dataset of six weeks was split into a training dataset of three weeks (June 13th - July 3rd), except for the evaluation of the training period length in Figure 7.6b, and a test dataset of two weeks (July 11th - July 24th). The week between training and test period was omitted to prevent interference and to avoid the national holiday on July 4th (Independence Day).

Results of availability estimation for taxi crowdsensing

The results of the parking availability estimation for 486 taxis (pessimistic case) using a random forest classifier with 200 trees are presented in the confusion matrix in Table 7.1 and the Precision-Recall curve in Figure 7.6a. In total, there were 1,528,582 cases considered, which were all time instants during the two-week test period in July on any parking segment of the evaluation area, when parking was allowed. Regarding the confusion matrix, the unweighted random forest classifier estimated that there were 112,195 cases of ‘full’ parking segments (positive class), of which 94,836 cases were correct (true positives) and 17,359 cases were incorrect (false positives). The existence of at least one empty parking space (‘not full’, negative class) was correctly estimated in 1,389,349 cases (true negatives) and wrongly predicted in 27,038 cases (false negatives). These numbers led to a precision of 84.5 %, a recall of 77.8 %, and an F-measure of 81.0 %. These values correspond to the red cross in Figure 7.6a. In this figure, the Precision-Recall curve is also illustrated, in blue. The optimal F-measure value on this curve was 81.5 % (see also Table 7.2).

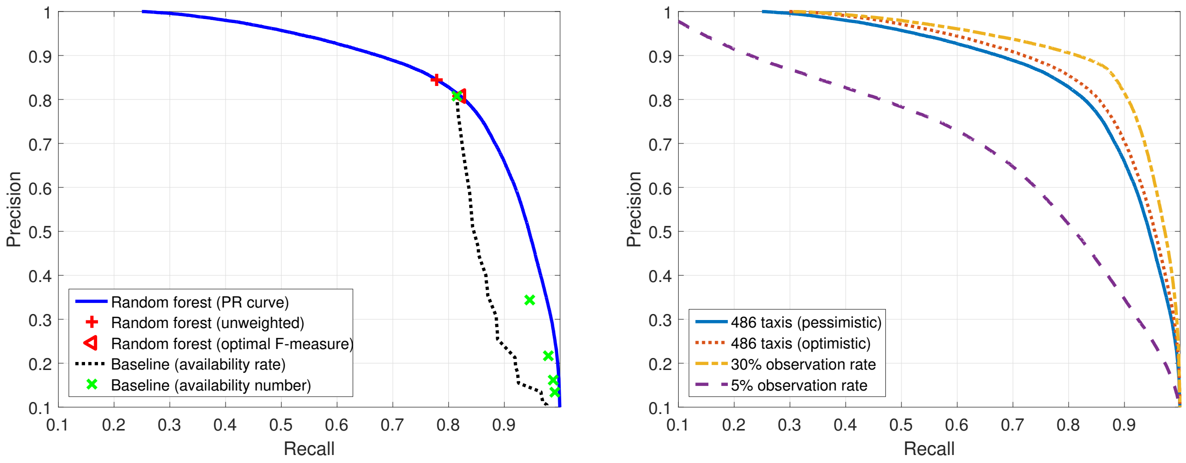
Table 7.1: Confusion matrix of results for binary classification with 486 probe vehicles in the pessimistic case.

		Actual Parking State	
Predicted		Full	Not full
Parking	Full	94,836 (TP)	17,359 (FP)
State	Not full	27,038 (FN)	1,389,349 (TN)

Table 7.2: Evaluation metrics for result with 486 probe vehicles in the pessimistic case.

Method	**Precision**	**Recall**	**F-measure**
Random forest (unweighted)	84.5 %	77.8 %	81.0 %
Random forest (optimal F-measure)	80.8 %	82.2 %	81.5 %
Baseline (optimal F-measure)	80.9 %	81.5 %	81.2 %

The baseline estimations, based on the availability number (green crosses) and on the availability rate (black dotted line), are also shown in Figure 7.6a. The optimal F-measure (81.1 %) was reached for the case in which ‘full’ is predicted only if it was also ‘full’ in the last observation. This result was very similar to the estimation based on the random forest classifier. If good recall was



(a) Classification and baseline result for a fleet of 486 taxis.

(b) Results for 486 taxis and constant observation rates.

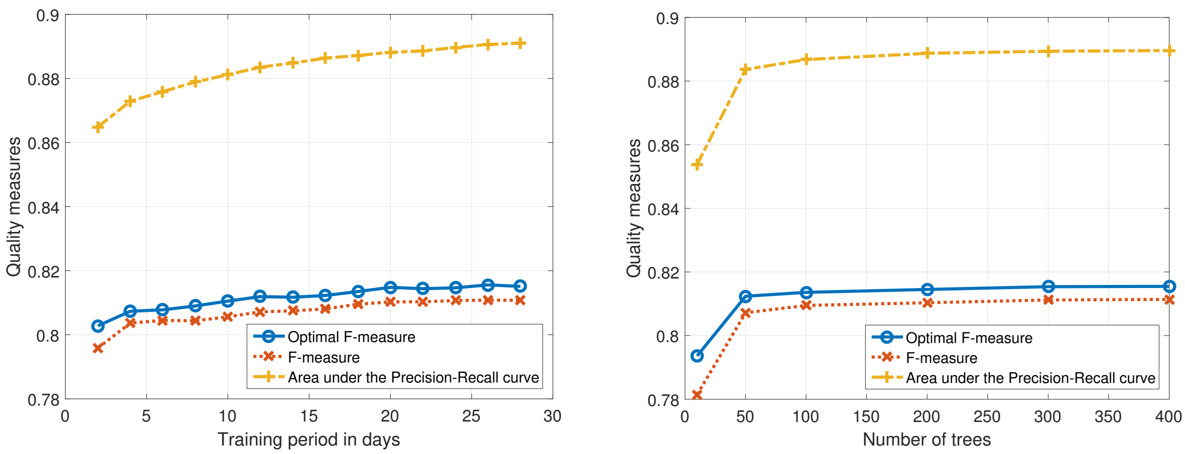
Figure 7.6: Precision-Recall curves for parking availability estimation: (a) availability estimation for 486 taxis and comparison with the baseline (persistence method) and (b) availability estimation for 486 taxis (pessimistic and optimistic) and constant observation rates of 5 % and 30 %.

more important than precision, the random forest classifier showed better results than the baseline estimations. In the opposite case, the baseline prediction was not able to generate high precision values except when 'not full' was predicted for all cases. This is the case since a threshold lower than 0 for the availability rate in the last observation has no effect. In Figure 7.6b and Table 7.3, the results for 486 taxis are compared with the models with constant observation rates. Similarly to the comparison of the spatio-temporal coverage in Subsection 6.6.3, a fleet of 486 taxis is somewhat worse than the 30 % observation rate, but clearly better than the 5 % observation rate.

Table 7.3: Evaluation metrics for the results with 486 probe vehicles as well as constant observation rates of 5 % and 30 %.

Observation scheme	**Optimal F-measure**	**Unweighted F-measure**	**Area under the PR curve**
486 probe vehicles (pessimistic)	81.5 %	81.0 %	88.9 %
486 probe vehicles (optimistic)	82.9 %	82.6 %	90.4 %
30 % observation rate	86.9 %	86.8 %	93.0 %
5 % observation rate	67.3 %	63.5 %	72.0 %

The curves in Figure 7.7 support the parameter choices in the experiments. The length of the training period was varied between two and 28 days. The results (Figure 7.7a) show a continuous increase of the quality measures with increasing length of the training period. However, for values larger than 20 days, small improvements were observed in the area under the Precision-Recall curve (AUPR), but the F-measure and the optimal F-measure were almost constant. For all other experiments, 21 days were chosen as the training period. Figure 7.7b illustrates the quality measures for different numbers of trees in the random forest classifier. A large improvement was observed only for the increase from 10 to 50 trees, while the quality values were nearly constant for more than 50 trees. Thus, the choice of 200 trees for all experiments is rather conservative.



(a) *(b)*

Figure 7.7: Learning curves for the variation of (a) the training period and (b) the number of trees in the random forest classifier.

Quality of crowdsensing in comparison to static sensors

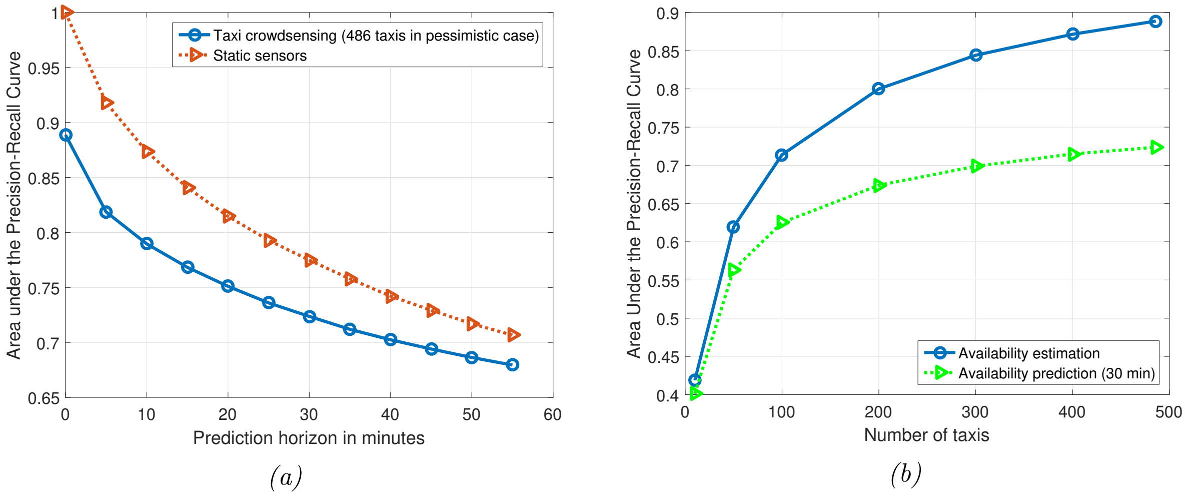
A comparison of the estimation and prediction quality between the crowdsensing solution with 486 taxis (pessimistic case) and the continuous measurement with static sensors is shown in Figure 7.8a. As might be expected, the static sensors always achieved a better result than the crowdsensing. However, the larger the prediction horizon, the smaller the difference became between both solutions. For a zero minute horizon, the static sensors knew the complete situation (assuming correct sensors) and thus had an AUPR value of 100 %. With an increasing prediction horizon, the quality decreased for both cases. This allows a direct comparison between both curves: for example, the quality of taxi crowdsensing for a prediction horizon of five minutes is about the same as the quality of static sensors with a horizon of 20 minutes. To better interpret differences in the quality, a typical parking search scenario is described in Chapter 8.

Evaluation for different numbers of taxis

A very interesting piece of information for practitioners is the size necessary for a taxi fleet to achieve a reasonable crowdsensing quality. In Figure 7.8b, the quality in terms of AUPR was computed for different fleet sizes and for both the parking availability estimation and the prediction for a 30-minute horizon. In both cases, the quality clearly increased with an increasing number of taxis. It can be assumed that the curves asymptotically converge to the values of the static sensors for very high numbers of taxis, and consequently continuous observations. A fleet of 300 taxis achieved almost the same quality as the 486 taxis in the complete dataset. Thus, this size might be a good trade-off between quality and costs.

Feature importance

To investigate the importance of different features, a leave-one-out feature analysis was computed. In this evaluation, the estimation results were compared with the results when leaving out one type of feature. Only the last observation of the parking segment was kept for all cases, as it was shown already in this section and in Section 6.2 that this value had a very high predictive quality. The results are shown in Figure 7.9. All features except the adjacent parking segments led to improvements in all cases. The addition of further recent observations (second to fifth last observation) led to the highest improvement for the prediction with static sensors. For the crowdsensing estimation, the number of parking spaces in a parking lane (parking capacity) was the most important feature, but with just a small relative improvement of less than 0.5 %. The features



(a) *(b)*

Figure 7.8: Quality evaluation of the estimation and prediction for (a) different prediction horizons and (b) different numbers of taxis for the pessimistic case.

computed from parking availability in adjacent parking segments resulted in a large improvement for the crowdsensing prediction, while there was a clear decrease of about 1 % for the static sensor prediction.

Classification score as probability estimation

The classification score, resulting from the voting of trees in the random forest (as described in Section 2.2.3), gives an indication of the reliability of the classification result. Its value is in the range between 0 and 1. A value close to the limits means that the classifier is very certain (0: at least one parking space is available, 1: no parking), while a value around 0.5 means that the classifier is unsure. In Figure 7.10a (complete view) and Figure 7.10b (enlarged view), the distribution of the classification score is shown for the availability estimation in the test period. In many cases, the classification score was close to 0 and there was also parking available. For all cases with parking available (‘not full’), the count of cases decreased homogeneously with an increasing classification score. For the ‘full’ cases, the count increased with an increasing classification score and a high count of 30,457 cases (27.2 % of all ‘full’ cases) had the value 1. This distribution shows that only a few cases were wrongly classified if the classification threshold was set to 0.5.

An attribute often wanted is a probability estimation of the classification result. In the case of parking search, it is desirable that parking routing decisions are based not only on the estimated classes ‘full’ or ‘not full’, but also on a probability estimation, to provide a more reliable recommendation. For random forests, the classification score already provides a good estimation. Thus, the ratio of an actual class, in this case the class ‘full’ for all cases with a specific classification score, was directly compared with the value of the classification score as a probability estimation. In Figure 7.11, the classification score for both the parking estimation and the parking prediction shows a continuously increasing behavior. For the parking estimation in particular, the behavior came very close to the ideal distribution. Thus, the classification score represented a good probability estimation.

7.4 Concluding remarks

Comparison of approaches for parking availability estimation

In this chapter, the parking availability was estimated using three different approaches: spatial

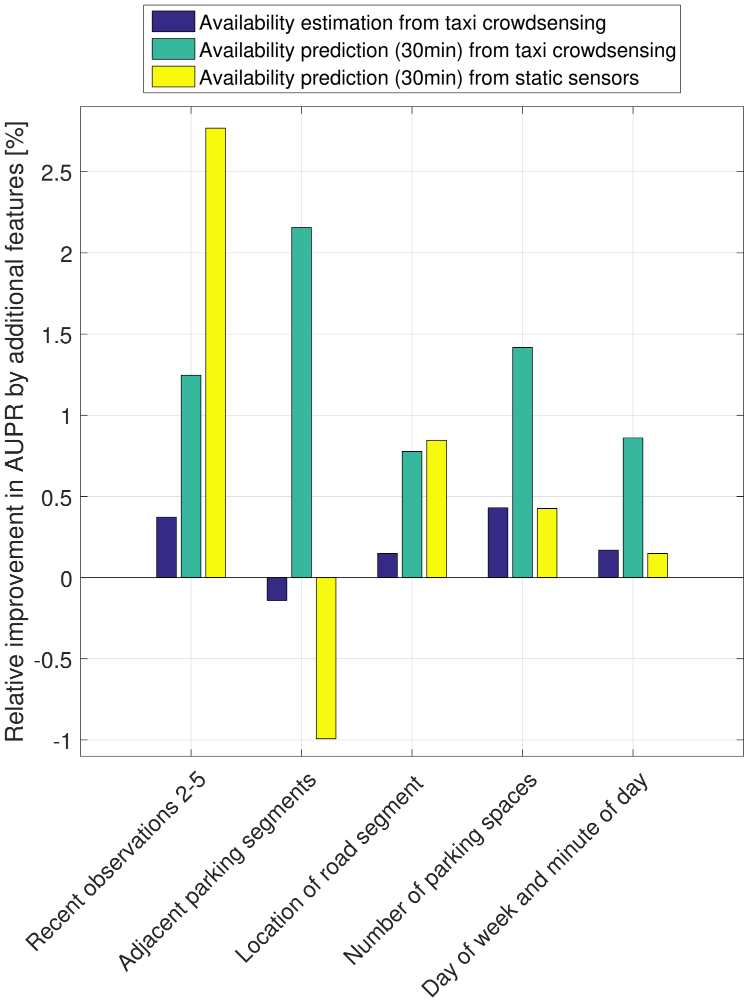
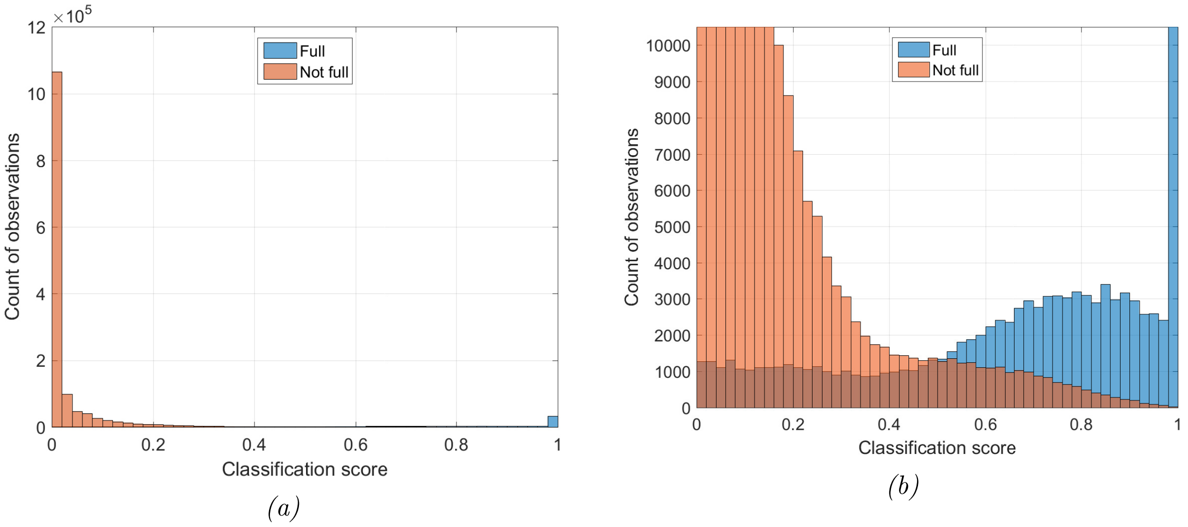


Figure 7.9: Feature importance evaluation by a leave-one-out analysis for the availability estimation and prediction of 486 taxis in the pessimistic case and the prediction with static sensors. The quality was computed with and without the specific feature type and the relative improvement in terms of AUPR was determined.



(a) *(b)*

Figure 7.10: Histogram of the classification score (486 taxis in the pessimistic case) for all cases in the test period shown in (a) a complete view and (b) an enlarged view.

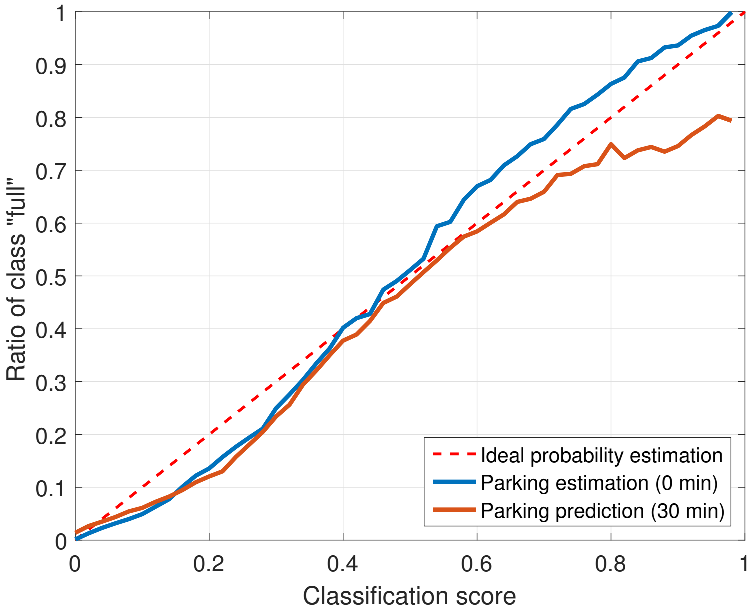


Figure 7.11: Ratio of the class 'full' for different classification scores in comparison with an ideal probability estimation for 486 taxis in the pessimistic case.

interpolation, persistence of the last observation, and binary classification with random forests. The spatial interpolation clearly achieved the worst results for any taxi fleet size. For a fleet size of 486 taxis, the Mean Absolute Error (MAE) values of the availability rate was about 0.18, but only 0.08 for the persistence method. In Section 7.3, it was shown that the persistence method as a baseline estimation gives about as good results as the classification approach, if false positives and false negatives have the same relevance.[2] The more sophisticated method might not achieve superior results, because arrival and leaving of vehicles might be highly random and thus hard to estimate for short periods.

The main benefits of the classification approach, compared with the persistence method, are threefold: firstly, an expressive score representing a probability estimation of the result is only available for the classification approach. This information can be very useful when computing a route for finding parking (Jossé et al., 2013; Guo and Wolfson, 2016). Secondly, the classification approach can easily be extended to predict the parking availability in the near future. Thirdly, a preference for a low number of false positive or false negatives can easily be implemented in the classification approach by defining weights. In such unbalanced cases, the random forest classifier revealed better results than the persistence method. In the case of a parking guidance system that leads a user to the next available parking space, the system might be perceived as more reliable if the number of recommendations for an actually full parking segment is low (false negatives). On the other hand, available parking in not-recommended road segments (false positives) may be not perceived by the user.

Relevance of spatial relations in on-street parking availability data

The spatial relations in on-street parking availability data were addressed in multiple sections. In Section 6.4, the spatial correlation was analyzed and only a weak correlation was found. The spatial interpolation of parking measurements was evaluated in Section 7.1 and only a small improvement

---

[2] Note that a direct comparison between interpolation and the binary classification method is not possible, because the interpolation estimates the parking availability rate (regression problem) and the classification distinguishes only between full and not full. The persistence method can easily be applied both for regression and classification.

over a simple averaging was found for the Inverse Distance Weighting method. Finally, spatial features were included in the binary classification in Section 7.3. These features were beneficial in only some cases. Thus, it can be concluded that parking is a rather local phenomenon in the parking dataset investigated. Therefore, spatial relations have a minor relevance for parking availability estimations and predictions, in contrast with the work by Rajabioun and Ioannou (2015), who achieved a strong improvement by including the spatial context.

Parking crowdsensing with taxis as probe vehicles

The evaluation of taxi trajectories in the investigation area in San Francisco (see Section 6.6) showed that taxi coverage is spread over all investigated parking segments, and they are active throughout the whole day. Sufficient coverage was observed in about 90 % of the cases for the persistence method (Section 7.2) and the AUPR quality measure for the classification approach came close to the values of static sensors (Section 7.3), which are much more expensive to install and maintain (Xu et al., 2013).

However, a strong dependence on the time of day and the road class was found. While the parking activities showed a similar daily pattern and thus compensated for the periods of lower taxi activity, small roads were seldom visited even when parking was changing frequently, leading to low quality values in those cases. Regarding the number of taxis, the quality clearly increases with more taxis in an asymptotic manner. Thus, only small improvements were reached for fleets with more than 300 taxis. Practitioners need to decide whether more probe vehicles are worth the additional effort. However, the results based on standard machine learning measures, such as the Area Under the Precision-Recall curve (AUPR), are hard to interpret for such a decision. Therefore, the relevance of different on-street parking information is evaluated in a typical search scenario in the following section.

The smartphone case, although optimistically modeled, clearly achieved worse results in all evaluations. Thus, a much larger user base than assumed (4 % of all drivers) is necessary to achieve good results and to compete with the parking crowdsensing solution. This result is in line with the work by Lendák and Farkas (2016). They evaluated the potential reasons for the failure of Google's parking app OpenSpot. They concluded that the information quality of the app was poor due to a user rate that was too low, so a very high user rate is necessary for a successful parking app. It might be particularly challenging for parking app start-ups to achieve satisfying information quality, at least for the approaches considered in this thesis and in the investigations by Lendák and Farkas (2016).

Limitations of evaluations

The evaluations are limited to some districts of San Francisco. Investigations are needed whether the relative performances of the methods presented are also valid for other parts of the city or further cities. However, the models are generic and do not include city-specific information. Thus, it is likely that the performance is similar in other cities, if training data from these cities are used. A challenge might arise for parking segments without markings for every stall. Since the maximal number of vehicles fitting in the parking segment depends on the length of the vehicles, the parking availability rate is harder to estimate. Also, when a parking gap is available for a small vehicle, the gap might be too small for a larger vehicle.

The limitations of the SFpark dataset also need to be considered, as discussed in Section 6.7. At the same time, the coverage of the taxis is somewhat underestimated. Several trips were not considered, because the map matching failed in these situations. If all trips were considered in the evaluation, the coverage might be even significantly higher.

8 Benefits of crowdsensed parking availability information

In the previous chapter, it was shown that the crowdsensing of parking availability with taxis as probe vehicles is a promising solution, as the results for the quality metrics come close to the results for static sensors. However, it is hard to interpret the remaining quality difference regarding the impact for the drivers. Thus, in this chapter, a typical scenario of parking search is evaluated regarding the benefits for drivers using different kinds of parking information.

In Section 8.1, the types of parking information are presented in the form of map layers based on the work in the previous chapters. The parking route choice at an intersection is modeled as an evaluation scenario in Section 8.2. Results are described in Section 8.3. Finally, the results and limitations of the evaluation are discussed in Section 8.4.

8.1 Types of information for on-street parking

Broadly speaking, there are two main map layers in a dynamic parking map in addition to the basic road network layer, as illustrated in Figure 8.1:

1. *Parking legality layer*, containing information about the location of on-street parking lanes, together with their vehicle capacity and optional temporal restrictions.
2. *Parking availability layer*, containing dynamic information about the parking availability information, from either static or mobile sensors. For the mobile sensors, the availability estimation is based on either the last observation (persistence method, see Section 7.2) or on a probability estimation of the binary classification (Section 7.3).

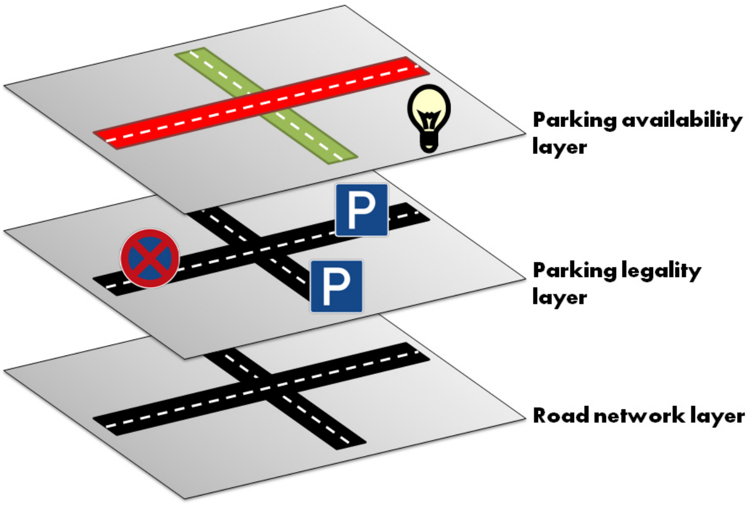


Figure 8.1: The basic road network layer plus the two layers of the dynamic parking map: the parking legality layer with the parking capacity and the parking availability layer with measurements from static or mobile sensors.

The relevance of these layers is evaluated in the following section, for different settings:

Parking legality layer with parking capacity information
The parking legality layer consists of the locations of parking lanes, their temporal restrictions, and their capacity. They are usually defined by the city authorities. However, as maps of parking lanes are often unavailable, manual recording or crowdsensing solutions are needed to map parking lanes (as described in Chapter 5). Parking capacity can change over time of day, both for regular reasons, such as peak-hour driving lanes and street cleaning, and on an irregular basis for construction sites, for example. If each parking space is equipped with a parking meter, the capacity of the lane is fixed. If not, the capacity depends on the length of the lane, the car size, and the distances between the vehicles. In this evaluation, using the SFpark dataset as presented in Chapter 6, all parking spaces were metered and the capacity was provided continuously via the API, along with the availability information for every parking segment.

Parking availability layer: availability estimation from last measurement of mobile sensors
The last measurement from the crowdsensing vehicles (persistence method) is a good indicator for the current parking availability in a parking segment, as evaluated in Section 7.2. The simulated parking observations based on taxi trajectories (Section 6.6) were used in this evaluation.

Parking availability layer: availability estimation based on binary classification for mobile sensors
To estimate the probability that at least one parking space is available in a parking segment, the binary classification model using a random forest classifier was used, as described in Section 7.3. The classification score of the random forest classifier was used as a probability estimation for finding an empty space in the parking segment. Also, for this setting, the simulated parking observations based on taxi trajectories were used.

Parking availability layer: real-time information from static sensors
The best information about the current parking availability can be obtained from static sensors. In this evaluation, the SFpark measurements were used as real-time information and also as ground truth, assuming that all measurements were correct.

8.2 Experimental setup

The relevance of different parking information layers for the drivers' search was evaluated in a typical parking search scenario. This scenario investigates the situation when a driver does not find a parking space in the parking segment at the destination and starts to roam. In this situation, the driver needs to decide which road to take next, as illustrated in Figure 8.2. Similarly to the parking search simulation by Millard-Ball et al. (2014), it is assumed that the driver considers all roads following their current one. In contrast to their approach, the driving directions and prohibitions, e.g. due to one-way roads, were considered in this evaluation. It is further assumed that drivers do not make U-turns. As discussed by Millard-Ball et al. (2014), some drivers carry out this maneuver, but it is illegal in the investigation area. The parking search is limited to only the next road segment, as the parking search can be seen as a series of en-route decisions at the intersections (Bonsall and Palmer, 2004). If there is no parking space available at the next road segment either, then the driver has to re-think which road segment to take next. Walking distances are neglected in this consideration.

8.2.1 Routing strategies

The routing strategies of a parking guidance system are based on the availability of the dynamic parking map layers. The following strategies were evaluated:

1. *Baseline strategy*: the drivers do not have access to any parking information and just randomly choose the subsequent road.
2. *Parking legality strategy*: a random road is chosen where parking is allowed at that time.
3. *Parking legality and capacity strategy*: the road with the highest parking capacity is chosen.
4. *Crowdsensed availability with persistence strategy*: the driver is guided to the road where the parking availability was highest based on the last observations of the crowdsensing vehicles.
5. *Crowdsensed availability with classification strategy*: the road is suggested that has the highest estimated probability of having at least one empty parking space.
6. *Optimal strategy*: the driver knows the real-time availability of all parking segments and therefore always makes the correct choice, if there is a subsequent road with an empty parking space.

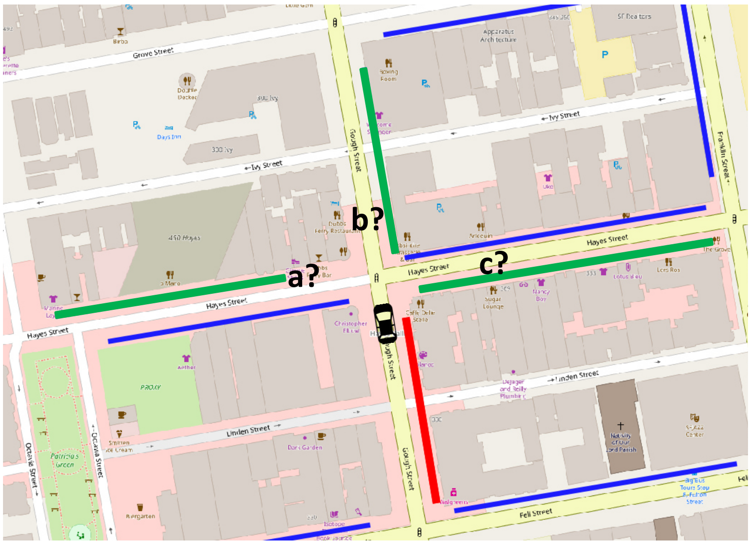


Figure 8.2: Example for the scenario where a driver does not find parking at the destination and needs to decide on the subsequent road. The red line represents the full parking segment at the destination. The green lines are parking segments reachable from the intersection without U-turn, the blue lines are further parking segments.

As an evaluation measure, the number of unsuccessful decisions (not finding parking in the next parking segment) was counted for the different strategies.

8.2.2 Data sources

For the experimental evaluation, the SFpark dataset (see Section 6.1) and the simulated crowdsensing dataset from taxi trajectories (for 486 taxis with the pessimistic case; see Section 6.6) were used. The period from July 11th until July 24th, 2013 was used for the evaluation, and the period from June 13th until July 3rd, 2013 for training of the availability estimation (analog to Section 7.3). In addition, the information about the topology and characteristics of the road network were extracted from OpenStreetMap. For the area of investigation considered, 420 road segments with parking information were available, of which about 40 % were one-way roads.

For all road segments with parking information, the route options at the intersection in the driving direction were determined from the road network of OpenStreetMap. However, there were intersections with fewer than two route choices with parking information (e.g. at the border of the SFpark pilot areas). In these cases, the road network was traversed until at least two road segments with parking information were reached, so that a route decision could be evaluated. This traversal was based on three steps: first, the shortest path routes were computed from the intersection considered to the entry point of all road segments with parking information, limited to a maximal distance of 500 meters to avoid an unrealistic route choice. Second, all routes were excluded that contained a road segment with parking information before reaching the destination road segment. Third, the routes were iteratively added to the final set of route choices with an increasing number of traversed road segments until at least two routes were contained in the set. For example, if no road segment with parking information can be reached directly from an intersection and there are three road segments with parking information that can be reached from the intersection via one other road segment, then no further routes were considered with more than one other road segment between the starting intersection and the destination. If fewer than two subsequent road segments were found by this method, the initial road segment was excluded from the evaluations (18 % of the road segments).

8.3 Evaluation of the impact of different parking information

The scenario was evaluated for all time instants per road segment when the corresponding parking segment was completely occupied. Parking segments were not considered as drivers' destinations during periods of parking prohibition. In each of the cases evaluated, it was assumed that a driver arrived at this road segment and had to decide on a subsequent road segment at the next intersection. The ratio of unsuccessful decisions was determined, i.e. the ratio of cases without empty parking spaces in the road chosen. In Subsection 8.3.1, all of these cases are evaluated. The comparison of different parking guidance strategies is more relevant in cases where there is a correct and a wrong choice (at least one empty and one full road segment). Consequently, these cases were further investigated in Subsection 8.3.2. In addition, results were analyzed for the hours of the day. The parking capacity was very different among the parking segments in the investigation area in San Francisco, as shown in Figure 6.3a (page 74). To assess the impact of the capacity, the results were evaluated for different levels of parking capacity similarity in Subsection 8.3.3.

8.3.1 Results for all decisions in the dataset

For the evaluation period of two weeks, all the spaces of a parking segment were occupied in 95,403 cases (time instants on a parking segment). An overview of the results for the different strategies is shown in the first column of Table 8.1. To be specific, if the subsequent road segment was chosen randomly, the driver did not find parking in the corresponding parking segment either in about 16 % of all cases. If the parking prohibitions in the subsequent road segments were considered, the ratio of unsuccessful cases was reduced to about 11.8 %. Using the parking guidance based on the complete capacity information in the parking legality layer reduced the ratio to only 5.6 %. Information from taxi crowdsensing achieved wrong decisions in 3.9 % of the cases for the persistence method and 3.5 % for the binary classification. These results were already very close to the optimal solution of 3.2 % for the real-time information from static sensors. The optimal solution was not zero, because there were some cases with no successful option. Concerning the

Table 8.1: Rate of unsuccessful choices for all route decisions (Subsection 8.3.1) and relevant decisions (Subsection 8.3.2).

	All decisions	**Relevant decisions**
Number of cases	95,403	28,744
Random choice	16.0 %	42.5 %
Parking legality	11.8 %	28.5 %
Parking legality and capacity	5.6 %	7.9 %
Crowdsensed availability with persistence	3.9 %	2.5 %
Crowdsensed availability with classification	3.5 %	1.1 %
Real-time information from static sensors	3.2 %	0 %

relative reduction, failure to find parking in the subsequent road segment was reduced by up to 80 % (=(16 %-3.2 %)/16 %) using the parking guidance based on static sensors, compared with the random choice.

8.3.2 Results for relevant decisions

Parking guidance is only able to help at intersections where there is at least one subsequent road segment with an empty parking space. Moreover, if there is parking available in all subsequent road segments, there is no need for a parking guidance system. Therefore, to compare the quality of parking guidance based on different layers of the dynamic parking map more effectively, only the cases with at least one full and one empty subsequent parking segment were considered. An overview of the results is presented in the second column of Table 8.1. The number of cases reduced from 95,403 to 28,744[1]. This means that parking guidance has the chance to help in about 30 % of all cases, if a driver does not find parking at the destination. In these cases, the result for the random choice increased to 42.5 %. If there were only two options at every intersection, 50 % was the theoretical result. But as there are situations with more than two choices, the result is lower than this theoretical value. Also, corresponding to expectations, the optimal result is 0 %. Compared with the result for the guidance based on parking legality and capacity with about 8 %, the ratio of unsuccessful decisions was reduced by a factor of three when using the persistence method with crowdsensing (2.5 %). The classification approach with crowdsensed availability data further reduced the ratio of wrong decisions to 1.1 %.

The results show that parking information can be used to significantly reduce the risk ending up in a full parking segment. However, for most types of parking information, the ratio of wrong decisions is on a low level. Thus, a decision based on the parking capacity is sufficient in most cases and there might be no need for parking availability information. Therefore, the influence of the parking capacity is investigated in more detail in the following subsection.

The impact of the time of day on the result for all relevant cases is illustrated in Figure 8.3. The ratio of unsuccessful decisions was almost constant over the day for the random choice and the taxi sensing strategies. A strong variation was observed for the strategies based on parking legality, with a small ratio of unsuccessful decisions during the night hours and the highest values around noon and in the evening. However, most of the parking search cases occurred only at these times, as shown in Figure 8.4a.

---

[1] Within the difference of 66,659 cases, all road segments had empty parking spaces in 63,638 cases and there was no road segment with an empty parking space in only 3,021 cases.

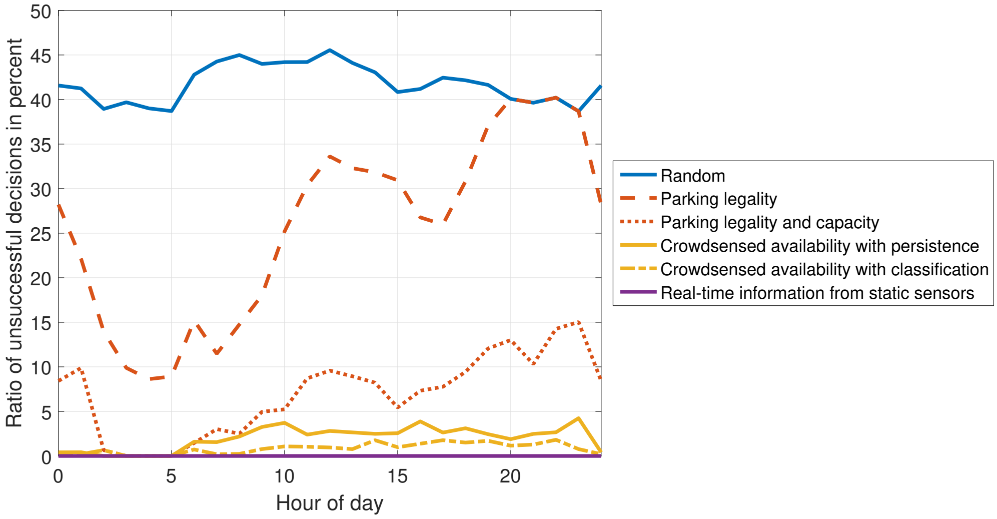
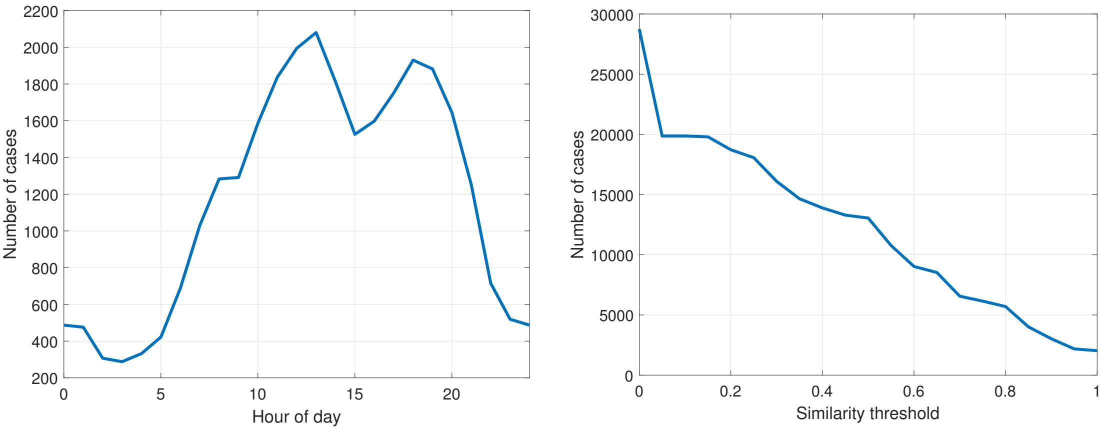


Figure 8.3: Ratio of unsuccessful decisions for different hours of the day and parking information strategies.



(a) *(b)*

Figure 8.4: Plot of the number of parking search cases for (a) different hours of the day and (b) different similarity thresholds of the parking capacity.

8.3.3 Similarity of capacity

The capacity of parking segments varied significantly in the investigation area in San Francisco (see also Figure 6.3a on page 74). Thus, the strategy based on the capacity information already achieved very good results. In this subsection, situations were evaluated in more detail for subsequent parking segments with similar capacities. A measure was defined for the capacity similarity $s(R)$ of a set of subsequent road segments R with capacities $c(R)$ in the corresponding parking segments as:

$$s(R) = 1 - \frac{\max(c(R)) - \min(c(R))}{\max(c(R)) + \min(c(R))} \tag{8.1}$$

This measure always has values between 0 (parking is not allowed in at least one subsequent parking segment) and 1 (the capacities are the same for all subsequent parking segments), except if all subsequent parking segments have a capacity of 0. Latter cases were excluded as they were not relevant anyway. At every end of a road segment, this similarity measure was computed for the set of subsequent road segments. If the similarity value was equal to or higher than a threshold, this road segment and the corresponding subsequent road segments were considered for the evaluation. However, if the similarity value was smaller than the threshold, it was determined whether a subset of subsequent road segments existed which fulfilled the similarity criterion. If there were at least two subsequent road segments with similar capacity, the intersection with these subsequent road segments was also considered for the evaluation.

In Figure 8.5, the results are shown for different similarity thresholds between 0 and 1, considering only the relevant cases. For a threshold of 0, the results correspond to the values in the second column of Table 8.1. With an increasing similarity threshold, only subsequent road segments with higher similarity were considered. The results for the strategy based on parking legality were equal to the random choice, because situations with parking prohibition have a similarity value of 0. When considering the parking capacity, the ratio of unsuccessful decisions increased monotonically until it reached values similar to the random choice at similarity thresholds of 0.75 and above[2]. The curves for crowdsensing also increase with a higher similarity threshold, but remain on a low level. The number of evaluated cases decreased continuously from 28,744 to 2,031 cases (see Figure 8.4b).

The strategy based on parking capacity is not better than the random guess for high similarity values, because there is no longer any relevant information gain from parking capacity information, as could be expected. Thus, parking availability information is much more relevant in situations when the parking capacity is similar in adjacent roads. This means that systems for parking availability information should focus on areas of the city where the parking capacity is similar.

8.4 Concluding remarks

Improvements of parking search using parking information

The achieved reduction of unsuccessful turn choices by up to 80 % is a considerably high percentage. This also means that parking search traffic could be reduced by up to 80 %, if it is assumed that drivers always make a new decision at every intersection. This assumption is atypical search behavior according to Bonsall and Palmer (2004) and is also used in parking search simulations,

---

[2] The slightly higher value for the capacity information compared with the random guess at a similarity threshold of 0.9 is likely to be an arbitrary variation caused by the small sample size.

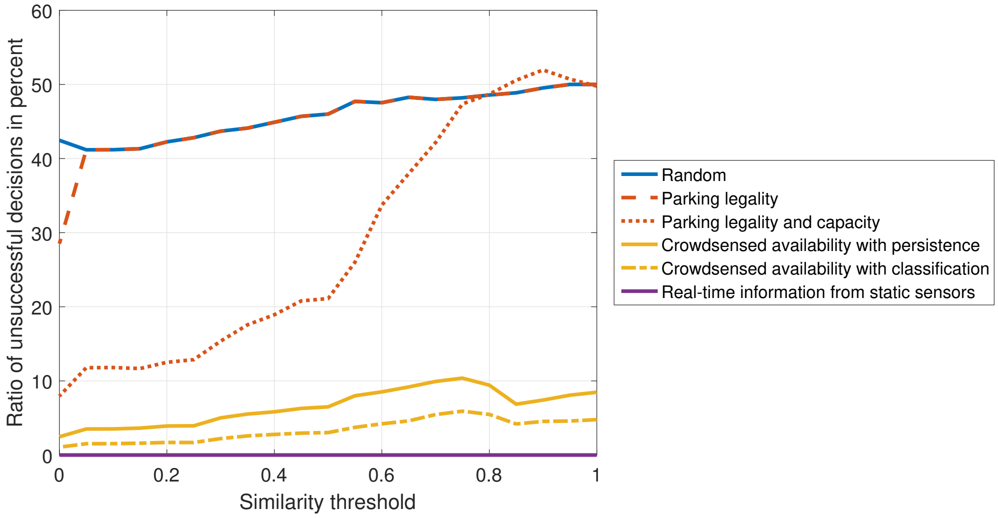


Figure 8.5: Ratio of unsuccessful decisions for different similarity thresholds and parking information strategies. Only the cases with similarity values higher than the threshold were considered.

such as the simulation by Benenson et al. (2008). However, it neglects the fact that drivers also consider the walking distance to the destination. In addition, drivers might not take every possible empty parking space they spot, due to individual preferences. Finally, in this scenario it is not considered that some drivers start to search for parking even before reaching the destination, for example as modeled by a probability distribution by Benenson et al. (2008). However, their parking simulation considered only parking search without any parking information.

Comparison of parking information

The guidance based on the capacity information strongly reduced wrong decisions by about 65 % for the full SFpark investigation area. This strong result implies that in such scenarios with high capacity heterogeneity, there might be no need for parking availability information systems. Nevertheless, for routing decisions with less heterogeneous capacity, the capacity information became irrelevant. All strategies with parking availability information still provided very reliable route suggestions, close to the optimum. Thus, the relevance of parking availability information, either from static sensors or probe vehicles, appears to be strongly dependent on the heterogeneity of the parking capacity in the city district.

The results for the strategies with parking availability information were similar in all cases investigated. In particular, the results from crowdsensing were nearly as good as the optimal result with static sensors. As the installation and maintenance of static sensors usually comes with high costs, traffic management authorities should also consider probe vehicles as a promising solution with lower costs, as proposed by Mathur et al. (2010). Since the classification approach is comparatively easy to apply, it is, despite only a small additional improvement, still recommended instead of the simpler persistence method.

Comparison with results from the literature

The potential reductions in parking search are higher in these results compared with the simulation study by Tasseron et al. (2016) where reductions were less than 50 % even in the most occupied settings. A main difference between their simulation environment and this real-world scenario is the heterogeneity in parking capacity and demand. Their model assumed a homogeneous parking demand, except for one city block, and a homogeneous capacity of the parking lanes throughout the network. They suggest in their study that higher heterogeneity might further reduce the parking search time using a parking guidance system. This assumption is supported by the results in this chapter. Another point is that only one vehicle on the search is considered in this chapter, while their simulation contained several vehicles. The number of informed vehicles searching at the same time presumably affects the benefits of a parking guidance system. In particular, if only the capacity information is used, all drivers would go to the same parking segments, regardless of the actual availability. This could lead to the inverse situation that the parking segments with the highest capacity would have the least availability in some cases. Meanwhile, a parking guidance system using parking availability information is able to adapt to this new situation and to direct drivers to other parking lanes.

Limitations of the evaluation and validity for other cities

It was assumed in this evaluation that there is one car arriving at every time instant when a parking lane is fully occupied. To achieve more precise results, the spatio-temporal distribution of the demand needs to be considered. Moreover, turn restrictions at intersections and (illegal) U-turns were not considered, which certainly affects the accuracy of the results. It can be assumed that the execution of U-turns shortens the parking search, as they are basically done when an empty parking space is spotted. In addition, the length of the road segments was not considered in the evaluation. The typical length of a road segment is between 150 and 200 meters and exceeds 200 meters in just a few cases. Thus, the travel time for length differences is in the same order[3] like the waiting times at intersections due to stop signs and traffic lights. Finally, only parking capacity data from the SFpark dataset is considered, but there might be also parking spaces without sensors, which would affect the chance to find parking in a road segment.

Several factors influence whether the results can be generalized to other cities, as parking behavior and taxi activities can be very different. Firstly, the parking demand and supply varies significantly among cities. In San Francisco, the low total number of fully parked parking segments indicates that parking search is not a dominant problem throughout the whole investigation area, but only at specific times in a few parts of the area. This result is in line with the results by Millard-Ball et al. (2014)[4]. They estimated the search by a simulation based on random walk using SFpark data from earlier years and found that there was an average cruising length of only 0.13 road segments. In cities with higher parking pressure, a parking guidance system should help in more situation. It is likely that the quality of the route recommendations is not affected by the level of parking pressure in situations with at least one successful option (*relevant cases* in this chapter). Secondly, the city layout with the capacity of parking lanes and their heterogeneity affects the parking guidance as shown in Subsection 8.3.3. Thirdly, for the quality of crowdsensing, the frequency of parking changes and the observation intervals of probe vehicles are essential (see also Subsection 6.6.4 on page 88). In general, the higher the frequency of parking changes, the more probe vehicles are needed to achieve a sufficient coverage. Both the frequency of parking changes and the sensing frequency of probe vehicles need to be investigated both spatially on road level and temporally for different hours of the day. In comparison between cities, the overall activity of probe vehicles can be different. For example, the annual mileage of taxis in New York City, US, is more than 110,000 kilometers (New York City Taxi&Limousine Commission, 2014), while taxis in a German city drive only about 60,000 kilometers per year (Linne+Krause, 2016). In conclusion, these factors need to be considered by traffic management authorities and companies, when considering the introduction of an on-street parking information system with either static or mobile sensor.

---

[3] At a velocity of $36\,\mathrm{km/h} = 10\,\mathrm{m/s}$, a road segment of 200 meters is traversed in 20 seconds.

[4] Note that Millard-Ball et al. (2014) modeled the parking search as a random walk (like the random guess in this evaluation), but they did not evaluate the use of parking information.

9 Conclusion and outlook

In this thesis, the approach of vehicular crowdsensing was investigated in depth, for the purpose of the generation of dynamic parking maps. In the following, the results and answers to the research questions are summarized and an overall conclusion is drawn (Section 9.1). In addition, further dynamic phenomena are presented and their main spatial and temporal characteristics, sensors for crowdsensed detection, and the possibility of estimation are listed in Section 9.2. Finally, future research directions are outlined in Section 9.3 based on the results presented in this thesis.

9.1 Research questions addressed and overall conclusion

In terms of the research questions described in the introduction, the following results were achieved:

- **How can on-street parking occupancy information be obtained with a LiDAR mobile mapping system?**
  Chapter 4 presented and carefully evaluated the processing pipeline to extract the locations of parked vehicles from records with a LiDAR mobile mapping system. This pipeline consisted of a preprocessing step to filter and enrich the 3D point clouds, a segmentation step to separate the point clouds of distinct objects, a classification step to identify parked vehicles in the point clouds, and an aggregation step to match the detected vehicles to the road network.

  The evaluation showed that the pipeline achieved a recall of 93.7 % at a precision of 97.4 %. While misclassifications were a minor error source, the segmentation step was the cause for most detection errors. The aggregation of the detection results showed that valuable parking occupancy information over the course of the day could be extracted. The aggregated parking data are very relevant for city planners to reduce parking shortages on specific roads. From many detections of parked vehicles, dynamic parking maps can be generated, as presented in this thesis.

- **How can parking legality maps, representing the legality of parking in the streets, be derived from crowdsensed parking observations?**
  The generation of parking legality maps from crowdsensed parking observations was modeled as a learning problem in Chapter 5. The legality was determined individually for each small subsegment of the road. Several features were proposed to describe the parking occupancy on the subsegment itself and in its vicinity. A random forest classifier and a clustering-based classification with k-means were proposed to learn the legality.

  The approach was evaluated based on the detections from nine measurement drives of the LiDAR mobile mapping system with the aforementioned detection method. The results showed that the random forest classifier achieved the best results, with an accuracy of more than 93 %. The clustering-based classification achieved results similar to the approach from the literature by Coric and Gruteser (2013), however without the need for any training data. Thus, both methods are suitable for the generation of parking legality maps from crowdsensed data in cities where those maps do not exist or they are outdated.

- **How can parking availability be estimated based on crowdsensed parking availability information?**
  Three methods to estimate the parking availability from crowdsensed data were presented in Chapter 7. In the persistence method, the last parking availability observation in a parking segment was repeated for subsequent time instants until the next observation. In a spatial approach, the parking availability was estimated by interpolation using the Inverse Distance Weighting method. Finally, a classification approach was presented that estimated whether the parking segment was full or at least one parking space was still available. In addition to the last observations in the parking segment, features were proposed based on the occupancy in the surroundings, as well as temporal and spatial features.

  The spatial interpolation approach was clearly better than a random guess of the parking availability rate. However, it did not achieve convincing results, as the Inverse Distance Weighting method was mostly similar to a simple average over the complete investigation area. The persistence approach achieved much better results than the interpolation. Even the more sophisticated random forest classification could not outperform it for balanced precision and recall, presumably because arrival and leaving of vehicles might be highly random and thus hard to estimate for short horizons. The random forest classifier was beneficial, if either precision or recall were more important. In addition, the random forest classifier can also be trained to predict the parking availability in the near future. Furthermore, it has the advantage that its classification score for every sample is a good probability estimate as to whether an empty parking space can be found in a parking segment. This property is very helpful for the guidance of drivers on the search.

- **What is the potential quality of parking crowdsensing with taxis as probe vehicles?**
  In Chapter 6, parking crowdsensing with taxis as probe vehicles was simulated, based on large real-world datasets from static parking sensors and taxi trajectories. The crowdsensing performance was evaluated in great detail and compared with the static parking sensors in Chapters 6 and 7, assuming that they were always correct. For example, the parking availability estimation was correct or had a deviation of only one parking space in more than 90 % of the time instants over all parking segments, using the persistence approach and a pessimistic lane choice model. Focusing on the full parking segments, the binary classification led to a precision of 84.5% at a recall of 77.8 %. For the prediction, the crowdsensing results were nearly as good as predictions with continuous measurements from static sensors, with decreasing difference for higher prediction horizons.

  To better interpret the quality differences between the taxi crowdsensing and the static sensors, the relevance of both parking data sources and of the parking legality map was evaluated in Chapter 8 for an exemplary parking search scenario. The results showed that even the information about the parking legality and capacity could be a great help for drivers on the search, if the capacity is heterogeneous in the parking segments, as in San Francisco, but not for similar capacities of the parking segments. Parking availability information from crowdsensing further reduced the ratio of wrong choices on the search, with only small differences to the results of real-time information from static parking sensors.

Overall conclusion

This thesis presented several approaches to generate the content of dynamic parking maps. The approaches for obtaining parking data, learning the parking legality, and estimating the parking availability (except for the spatial interpolation) achieved very convincing results. Since these approaches were modeled in a generic way without city-specific information, they can easily be applied to other cities as well. Moreover, detailed evaluations showed that dynamic parking maps can help many drivers to shorten their search and thus to reduce the overall parking search traffic. Even if there are insufficient resources for continuous parking availability sensing, knowledge about the parking capacities can still offer good support.

Vehicular crowdsensing is a very promising solution for obtaining parking information for drivers on the search. In particular, it was shown that taxis are an excellent choice for probe vehicles, as they cover the complete city and are active all times of the day with their regular trips. Therefore, city planners and traffic management authorities should consider parking crowdsensing with taxis as a serious alternative to static parking sensors, which usually come with much higher costs. To assess the suitability of parking crowdsensing for a particular city, parking surveys and analysis of GPS trajectories from potential probe vehicles can be performed with the methods presented in this thesis.

9.2 Applicability of dynamic map approaches to further dynamic phenomena

Crowdsensing may also be profitable for many further dynamic phenomena in a street environment, which could be represented in a dynamic map. Some examples for those phenomena are:

Road constructions: Road constructions may exist for a few hours, such as lane paint works, or for many days, such as road renovations. As they are a potential accident risk and affect the traffic flow, a dynamic map of road constructions is helpful to warn drivers and to support automatic driving functions in future self-driving cars.

Potholes: Potholes can be inconvenient for car drivers and even dangerous for bicyclists and motorcyclists. Since they do not usually appear all of a sudden, a low temporal sensing resolution in a dynamic map is sufficient. For example, Eriksson et al. (2008) used sensor-equipped taxis to detect potholes.

Oil on road: Oil makes roads slippery and thus dangerous for drivers, especially for motorcyclists. Thus, frequent monitoring of the road in the order of minutes can prevent accidents.

Traffic signs: The location and content of static traffic signs change only seldom. However, variable traffic signs and mobile traffic signs (e.g. at road constructions) also exist. Information about traffic signs can assist drivers and can be used for automatic vehicle functions in combination with detection from on-board sensors. First products are already on the market (Pfeiffer, 2017).

Traffic light signals: The signal from traffic lights is very dynamic, as it changes every couple of seconds. Thus, a very high number of mobile sensors is necessary for a dynamic map. However, in cases with a periodic traffic light circuit, even a long-term prediction is possible, and thus a dynamic traffic light map requires fewer sensors (Axer et al., 2015; Protschky et al., 2015). Information about the traffic light signal is beneficial for fuel efficiency applications and traffic flow improvements.

Bike sharing fill level: For users of bike sharing services, the current fill levels of the racks is of importance. Within minutes, several bikes might be taken and the next user comes away empty-handed. Similarly to parking occupancy, the fill levels usually follow a typical day pattern, so that a long-term prediction is also possible (Vogel et al., 2011).

Street illumination: A dynamic map of street illumination is helpful for pedestrians during the hours of darkness and for city authorities to plan and replace streetlights. As lights have a long lifetime and a broken streetlight is not usually a safety risk, an update every few days is sufficient.

Speed traps: Both stationary and mobile speed traps can be included in a dynamic map to warn drivers. As mobile speed traps are only installed for several hours, a higher sensing frequency is needed in this case.

Street lanes: From time to time, the street layout changes or additional lanes are built. A map on lane level is discussed for self-driving cars (Joshi and James, 2015). If the self-driving cars do not strictly depend on the lane level map, an update frequency of the order of days is sufficient.

Traffic flow: Traffic flow may quickly change in the case of accidents in a time scale of minutes. Traffic flow information has several similarities with parking information: both have a periodic day pattern and a spatially heterogeneous distribution, and are usually treated on a road segment level. However, traffic flow is easier to observe with induction loops in the pavement or with GPS sensors in phones or cars. Moreover, flow models (e.g. based on fluid dynamics) ease the estimation in the case of insufficient sensor data.

Pedestrians: Dynamic pedestrian maps can be very helpful for drivers to prevent accidents. They contain either the current location of pedestrians with an update interval of a few seconds or the typical pedestrian flow at a lower frequency (Schlichting and Brenner, 2016).

Public transportation: If public transportation vehicles do not provide their location to the public, crowdsensing can be used to provide their current location and their estimated time schedule. An update interval of several seconds can be beneficial, if a passenger is trying to catch a bus.

Emergency vehicles: Dynamic maps of emergency vehicles could significantly improve rescue missions, if drivers were more aware of approaching emergency vehicles. However, as a very high sensing frequency is necessary, the emergency vehicles should transmit their location in addition to crowdsensing.

Rain: Rain information with an update interval of several minutes is beneficial for many fields, such as agriculture, leisure activities, or events. The wiper frequency of cars can be used to generate dynamic maps of precipitation (Fitzner et al., 2013).

Advertisements: For marketing purposes, the location and duration of billboard advertisements can be very relevant. For example, Matzen and Snavely (2014) presented a computer vision method to restore the time line of advertisements in New York based on online photo collections such as Flickr[1]. A low update interval of several days is sufficient.

The dynamic phenomena can be categorized according to the required location accuracy, the temporal dynamics (typical duration of validity), the temporal resolution necessary for sensing, the possible sensors for crowdsensed detection, and the possibility for short-term (missing data for a period in the order of the temporal resolution) and longer-term (missing data for longer periods) estimation of changes. Note that the two temporal scales can be very different, e.g. a traffic sign might change only after years, but when it is changed, the dynamic map should be updated within hours or days. Moreover, all ways of detection need a localization unit, such as a GPS receiver, in addition to the specified sensor. In Table 9.1, an overview of the phenomena for dynamic maps and their attributes is listed. In addition, Figure 9.1 illustrates the different orders of spatial and temporal resolution for a dynamic map of the phenomena presented. All determined values

---

[1] https://www.flickr.com/

are estimates and need further investigation before implementation of the corresponding dynamic map.

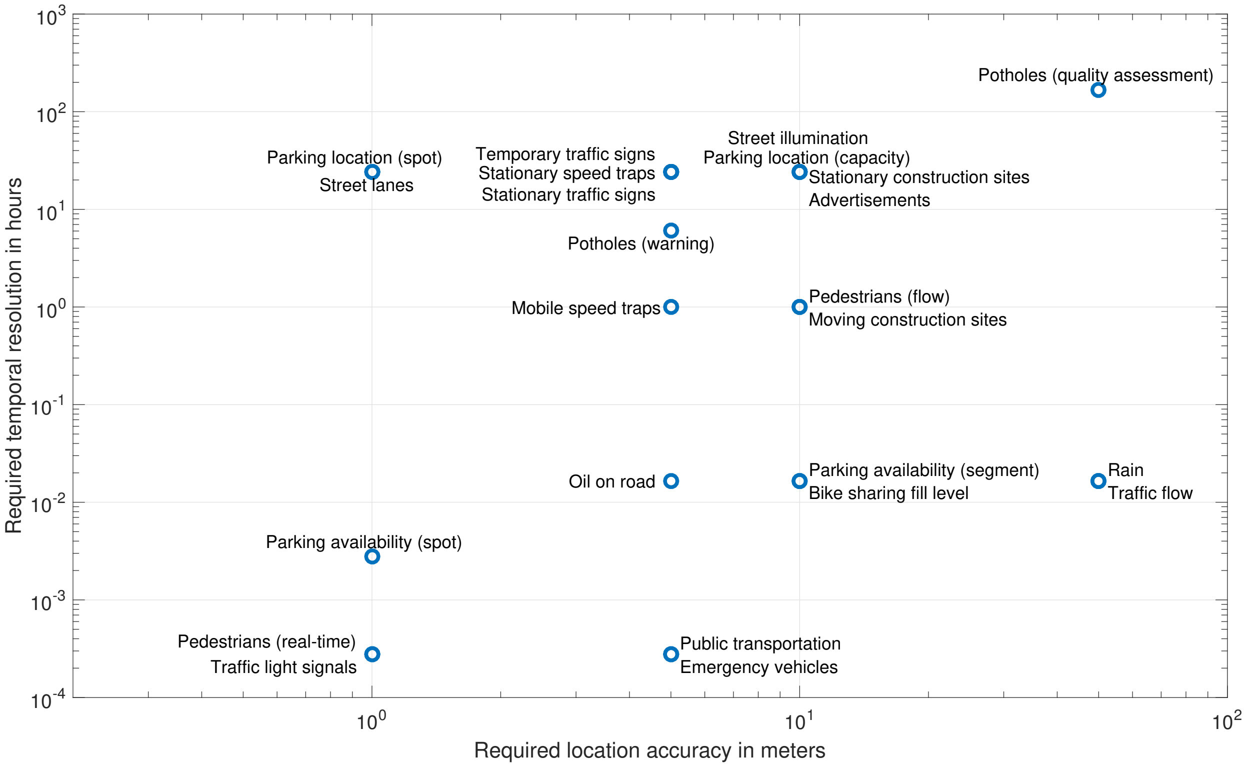


Figure 9.1: Estimated orders of temporal and spatial resolution for dynamic maps of different phenomena.

9.3 Future research directions

Regarding the detection of parked vehicles, more effort can be made to improve the segmentation step and the investigation of segmentation-free approaches. For example, Active Shape Models could be used to identify the vehicle shapes in the complete point clouds. Alternatively, Deep Learning methods are able to find structures from 3D point clouds without previous segmentation. Finally, since the LiDAR mobile mapping system is too expensive to be used on many probe vehicles, further work could be done on the usage of other sensors to detect parked vehicles or gaps in parking lanes, such as ultrasonic sensors or cameras.

Additional sensor inputs could be used for the improvement of the parking legality map generation. In particular, the fusion of information from detections of parking road signs, lowered curbs, and road markings could significantly improve the quality of the map. While lowered curbs and road markings can also be extracted from the LiDAR 3D point clouds, cameras are necessary to interpret the content of road signs. For the present approach to learning parking maps, further classification and clustering algorithms could also be evaluated, to find out whether they improve the map quality. Moreover, although a high accuracy was achieved with the proposed methods, some drivers would be guided to spots where parking is illegal, if the crowdsensed parking legality map was used. To reduce these cases, a reinforcement approach could be developed that considers correction messages by drivers and continuously improves the map quality. In addition, an extension of the approaches for more parking classes, such as special parking legislation (e.g. parking for people with disabilities) is possible. Since the occupancy characteristics are less distinct in this case, the supervised approach is likely to be superior over the other approaches described. Finally, if

Dynamic phenomenon	**Location accuracy**	**Temporal dynamics**	**Temporal resolution**	**Ways of Detection**	**Estimation**
Construction sites					
- moving	±10 m	hours	hours	Camera, LiDAR	S
- stationary	±10 m	weeks-months	days	Camera, LiDAR	S
Potholes					
- for warning	±5 m	weeks	hours - days	Camera, LiDAR, Accelerometer	S
- for quality assessment	±50 m	weeks	weeks	Camera, LiDAR, Accelerometer	S
Oil on road	±5 m	months	minutes	Camera	-
Traffic signs					
- temporary	±5 m	days	days	Camera, (LiDAR)[a]	-
- stationary	±5 m	years	days	Camera, (LiDAR)[a]	-
Traffic light signal	±1 m	seconds	seconds	Camera, (LiDAR)[a]	S & L
Bike sharing fill level	±10 m	minutes	minutes	Camera, LiDAR	S & L
Street illumination	±10 m	weeks	days	Camera	-
Speed traps					
- stationary	±5 m	years	days	Camera, LiDAR	-
- mobile	±5 m	hours	hours	Camera, LiDAR	L[b]
Street lanes	±5 dm - ±5 m	years	days	Camera, LiDAR, GPS	-
Traffic flow	±50 m	minutes	minutes	Camera, LiDAR, GPS	S & L
Pedestrians					
- flow	±10 m	hours	hours	Camera, LiDAR	S & L
- real-time	±1 m	seconds	seconds	Camera, LiDAR	S
Public transportation	±5 m	seconds	seconds	Camera, LiDAR	S & L
Emergency vehicles	±5 m	seconds	seconds	Camera, LiDAR	S
Rain	±50 m	minutes	minutes	Camera, LiDAR	S & L
Advertisements	±10 m	weeks	days	Camera	S & L
Parking location					
- capacity	±10 m	years	days	Camera, Ultrasound, LiDAR, GPS	-
- per parking spot	±1 m	years	days	Camera, Ultrasound, LiDAR, GPS	-
Parking availability					
- per parking segment	±10 m	minutes	minutes	Camera, Ultrasound, LiDAR, GPS	S & L
- per parking spot	±1 m	minutes	seconds - minutes	Camera, Ultrasound, LiDAR, GPS	-

Table 9.1: Overview of phenomena for dynamic maps.

---

[a] for location, but not for content

[b] e.g. maps could indicate locations where mobile speed traps are installed frequently

parking occupancy data is recorded over a longer time period, changes to parking legislation must be considered in the model.

In this thesis, the case of taxis as probe vehicles is investigated in depth. Certainly, other vehicle types, such as delivery vehicles, buses, or even private vehicles, could also contribute to the detection of the current parking availability. It would be very interesting to compare their spatio-temporal coverages. Moreover, in the evaluations presented, the vehicle sensors were assumed to be correct. Of course, these sensors also fail in some cases. Thus, it would be very interesting to see how the quality of a parking map depends on the quality of the sensors. Furthermore, specific algorithms might be necessary to compensate sensor errors and to provide robust estimations.

The results showed that the taxis cover the complete city well and every parking segment was visited at least a few times. Still, there is a high imbalance in the visit frequency between the parking segments. Approaches to send the taxis on small detours to reach a better spatio-temporal coverage could be promising[2]. However, taxi drivers and passengers need to receive small incentives to participate. An economic study is necessary to evaluate whether the improved coverage is possible at reasonable costs.

The estimation and prediction of parking availability based on the crowdsensed data is an essential part for the creation of a dynamic parking map. While the approaches presented already show promising results, deeper investigations of further machine learning approaches such as the use of Long-Short Term Memories, a particular Recurrent Neural Network for the prediction of long time series, could improve the results. Recent methods to automatically select classification algorithms and tune their hyperparameters could further enhance the results (Thornton et al., 2013).

The approach to parking crowdsensing was evaluated for the city of San Francisco, because it is the only city where on-street parking and taxi trajectory data is freely available. As soon as both data types are available for other cities (or a representative number of GPS traces for other vehicle fleets), particularly in other city types and on other continents, a deeper evaluation of the generalizability of the results is important. A more detailed comparison with different smartphone approaches would also be very interesting, as many start-ups (e.g. Parknav[3], Parkbob[4]) are currently trying to provide successful parking apps.

Parking guidance and information (PGI) systems can use dynamic parking maps to guide drivers on the search. However, in very crowded situations, the best PGI systems fail, if all parking spaces are occupied in the surroundings. Therefore, PGI systems need to be extended by a multi-modal routing, suggesting parking in a remote place and using alternative travel modes, such as public transportation or ride sharing, for the last mile.

---

[2] First results of this idea were presented in a Master's thesis by Koetsier (2017), supervised by this author. Results showed that even an average detour of 3 % could lead to a coverage improvement of about 30 %.

[3] http://parknav.com/

[4] http://www.parkbob.com/

List of Figures

1.1 Concept of vehicular crowdsensing for parking availability. 14
1.2 Overview of the parts of this thesis. 15

2.1 Steps of the Knowledge Discovery in Databases (KDD) process. 23
2.2 Taxonomy of data mining methods. 25
2.3 Example for a decision tree for the question ‘Do I need to wear a jacket?’. 27
2.4 ROC curves for an example classifier, a random classifier, and a perfect classifier. 29
2.5 Illustration of the steps in the k-means algorithm. 32
2.6 Example for a dendrogram generated with hierarchical clustering. 33

4.1 Riegl VMX-250 mobile mapping system. 46
4.2 Map of measurement track. 46
4.3 Block diagram of processing steps to extract non-moving vehicles. 47
4.4 Examples for moving vehicles with colors indicating which scanner recorded the points. . . . 50
4.5 Examples for over- and under-segmentation. 51
4.6 Example for the result of the object segmentation. 52
4.7 Example for the classification result. 53
4.8 Precision-Recall curve of classification result. 54
4.9 Examples for misclassifications. 54
4.10 Example for parking occupancies over the course of a day. 55

5.1 Overview of the steps for the generation of the parking legality map. 58
5.2 Projection of a detected vehicle on the road subsegments. 58
5.3 Examples for input data of measurement drives. 62
5.4 Illustration of regional subsets of investigated road segments. 63
5.5 Comparison of Receiver Operating Characteristic (ROC) curves. 64
5.6 Comparison of different cross-validation subsets. 66
5.7 Example for the resulting parking map. 67
5.8 Examples for wrong classification results. 67
5.9 ROC curve for different feature sets with k-means. 68
5.10 Feature importance from random forest training. 69
5.11 Accuracy boxplots of different methods for different numbers of measurement drives. 69

6.1 Map of pilot areas of SFpark. 72
6.2 Example for the occupancy and capacity time series of a parking segment over three days. . . 73
6.3 Histograms with ratios of parking capacities and empty parking spaces 74
6.4 Analysis of the periodicity characteristics of the parking availability rate 75
6.5 Negative Davies-Bouldin index and Silhouette index for clustering results 76
6.6 Clustering results for normalized average parking occupancy with Pearson correlation 77
6.7 Spatial distribution of the clustering result for normalized average parking occupancy with Pearson correlation 78

6.8 Clustering results for average parking occupancy with Euclidean distance 79
6.9 Comparison of mean absolute difference (MAD) for spatial and temporal relation 80
6.10 Cumulative distribution plot of inter-sampling times for crowdsensing modelling by down-sampling. 81
6.11 Overview of the processing pipeline for crowdsensed parking dataset. 83
6.12 Example for the GPS trajectory of one taxi over the complete observation period. 84
6.13 Examples for regularly and irregularly sampled taxi visits 86
6.14 Cumulative distribution function of the time gap for the optimistic and pessimistic observation cases. 87
6.15 Temporal characteristics of time gaps for the optimistic case 87
6.16 Spatial characteristics of time gaps for the optimistic case 89
6.17 Map of the mean time gaps per parking segment. 90
6.18 Plot of the taxi activity compared to the mean parking changes. 91

7.1 Comparison of methods for parking availability interpolation for different taxi fleet sizes. . . . 94
7.2 Frequency of observation differences for the persistence method. 95
7.3 Frequency of differences between estimated and actual numbers of empty parking spaces for different taxi fleet sizes. 96
7.4 Mean Absolute Error (MAE) of the parking occupancy rate estimation for different fleet sizes in the pessimistic case comparing the persistence and interpolation (IDW) methods. 97
7.5 Temporal dependence of the percentage of sufficient road coverage. 97
7.6 Precision-Recall curves for parking availability estimation. 100
7.7 Learning curves for the variation of the training period and the number of trees in the random forest classifier. 101
7.8 Quality evaluation of the estimation and prediction for different horizons and numbers of taxis. 102
7.9 Feature importance evaluation by a leave-one-out analysis. 103
7.10 Histogram of the classification score for all cases in the test period. 103
7.11 Ratio of the class ‘full’ for different classification scores in comparison with an ideal probability estimation. 104

8.1 The basic road network layer plus the two layers of the dynamic parking map. 107
8.2 Example for the scenario where a driver does not find parking at the destination. 109
8.3 Ratio of unsuccessful decisions for different hours of the day and parking information strategies. 112
8.4 Plot of the number of parking search cases. 112
8.5 Ratio of unsuccessful decisions for different similarity thresholds and parking information strategies. 114

9.1 Estimated orders of temporal and spatial resolution for dynamic maps of different phenomena. 121

List of Tables

2.1 Example for a confusion matrix. 28

4.1 Object segmentation results. 53
4.2 Confusion matrix of the classification result. 53

5.1 Overview of all feature sets used in this evaluation. 59
5.2 Results for all methods with different choice of data subsets and parameters. 65

6.1 Road classes of the parking segments for the investigated districts 73

7.1 Confusion matrix of results for binary classification with 486 probe vehicles in the pessimistic case. 99
7.2 Evaluation metrics for result with 486 probe vehicles in the pessimistic case. 99
7.3 Evaluation metrics for the results with 486 probe vehicles as well as constant observation rates of 5 % and 30 %. 100

8.1 Rate of unsuccessful choices for all route decisions and relevant decisions. 111

9.1 Overview of phenomena for dynamic maps. 122

Bibliography

Ahmed, M., Karagiorgou, S., Pfoser, D., Wenk, C., 2015. A comparison and evaluation of map construction algorithms using vehicle tracking data. GeoInformatica 19 (3), 601–632.

Axer, S., Pascucci, F., Friedrich, B., 2015. Estimation of traffic signal timing data and total delay for urban intersections based on low-frequency floating car data. In: 7th Mobil.TUM - International Scientific Conference on Mobility and Transport.

Axhausen, K., Polak, J., Boltze, M., Puzicha, J., 1994. Effectiveness of the parking guidance information system in Frankfurt am Main. Traffic Engineering + Control 35 (5), 304–309.

Ayala, D., Wolfson, O., Xu, B., Dasgupta, B., Lin, J., 2011. Parking slot assignment games. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '11. ACM Press, New York, New York, USA, pp. 299–308.

Behrens, R., Kleine-Besten, T., Pöchmüller, W., Engelsberg, A., 2015. Digitale Karten im Navigation Data Standard Format. In: Handbuch Fahrerassistenzsysteme. Springer, pp. 513–523.

Benenson, I., Martens, K., Birfir, S., 2008. PARKAGENT: An agent-based model of parking in the city. Computers, Environment and Urban Systems 32 (6), 431–439.

Bessghaier, N., Zargayouna, M., Balbo, F., 2012. An agent-based community to manage urban parking. In: Advances on Practical Applications of Agents and Multi-Agent Systems: 10th International Conference on Practical Applications of Agents and Multi-Agent Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 17–22.

Bill, R., 2016. Grundlagen der Geo-Informationssysteme. Wichmann.

Bishop, C. M., 2006. Pattern recognition and machine learning. Springer.

Bock, F., Attanasio, Y., Di Martino, S., 2017a. Spatio-temporal road coverage of probe vehicles: A case study on crowd-sensing of parking availability with taxis. In: Societal Geo-innovation. Lecture Notes in Geoinformation and Cartography. Springer International Publishing, pp. 165–184.

Bock, F., Di Martino, S., 2017. How many probe vehicles do we need to collect on-street parking information? In: International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). IEEE, pp. 538–543.

Bock, F., Di Martino, S., Sester, M., 2016a. What are the potentialities of crowdsourcing for dynamic maps of on-street parking spaces? In: Proceedings of the 9th ACM SIGSPATIAL International Workshop on Computational Transportation Science. IWCTS '16. ACM, New York, NY, USA, pp. 19–24.

Bock, F., Eggert, D., Sester, M., 2015. On-street parking statistics using LiDAR mobile mapping. In: IEEE 18th International Conference on Intelligent Transportation Systems. pp. 2812–2818.

Bock, F., Liu, J., Sester, M., 2016b. Learning on-street parking maps from position information of parked vehicles. In: Geospatial Data in a Changing World. Lecture Notes in Geoinformation and Cartography. Springer International Publishing, pp. 297–314.

Bock, F., Sester, M., 2016. Improving parking availability maps using information from nearby roads. In: Transportation Research Procedia. Vol. 19. Elsevier, pp. 207–214.

Bock, F., Xia, K., Sester, M., 2017b. Mapping similarities in temporal parking occupancy behavior based on city-wide parking meter data. In: Proceedings of 28th International Cartographic Conference.

Bonsall, P., Palmer, I., 2004. Modelling drivers' car parking behaviour using data from a travel choice simulator. Transportation Research Part C: Emerging Technologies 12 (5), 321–347.

Breiman, L., 2001. Random forests. Machine learning 45 (1), 5–32.

Bush, K., Chavis, C., 2017. Safety analysis of on-street parking on an urban principal arterial. In: TRB 96th Annual Meeting Compendium of Papers. Transportation Research Board.

Caicedo, F., Robuste, F., Lopez-Pita, A., 2006. Parking management and modeling of car park patron behavior in underground facilities. Transportation Research Record: Journal of the Transportation Research Board (1956), 60–67.

Caliskan, M., Barthels, A., Scheuermann, B., Mauve, M., 2007. Predicting parking lot occupancy in vehicular ad hoc networks. In: IEEE 65th Vehicular Technology Conference - VTC2007-Spring. pp. 277–281.

Caruana, R., Niculescu-Mizil, A., 2006. An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd International Conference on Machine Learning. ACM, pp. 161–168.

Chen, X., Santos-Neto, E., Ripeanu, M., 2012. Crowdsourcing for on-street smart parking. In: Proceedings of the second ACM international symposium on design and analysis of intelligent vehicular networks and applications. ACM, pp. 1–8.

Cleverciti, 2017. Lösungen für Smart Parking. `https://www.cleverciti.com/de/technologie/parken-sensoren/`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5n8F2yT`].

Coric, V., Gruteser, M., 2013. Crowdsensing maps of on-street parking spaces. In: 2013 IEEE International Conference on Distributed Computing in Sensor Systems. IEEE, pp. 115–122.

Daimler AG, 2017. Pilotprojekt von Bosch und Daimler zu Community-based Parking: Ihr Mercedes-Benz wird zur Parkplatz-Suchmaschine. `http://media.daimler.com/marsMediaSite/ko/de/13521746`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5n99W7r`].

Davies, D. L., Bouldin, D. W., 1979. A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1 (2), 224–227.

Davis, J., Goadrich, M., 2006. The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning. ACM, pp. 233–240.

Delot, T., Ilarri, S., Lecomte, S., Cenerario, N., 2013. Sharing with caution: Managing parking spaces in vehicular networks. Mobile Information Systems 9 (1), 69–98.

Demantké, J., Mallet, C., David, N., Vallet, B., 2012. Dimensionality Based Scale Selection in 3D Lidar Point Clouds. In: ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVIII-5/. pp. 97–102.

Dubé, R., Hahn, M., Schütz, M., Dickmann, J., Gingras, D., 2014. Detection of parked vehicles from a radar based occupancy grid. In: Intelligent Vehicles Symposium Proceedings. IEEE, pp. 1415–1420.

Eggert, D., 2017. Effiziente Verarbeitung und Visualisierung von Mobile Mapping Daten. Vol. 808 of Deutsche Geodätische Kommission: C (Dissertationen). München.

Eggert, D., Sester, M., 2013. Multi-layer visualization of mobile mapping data. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. II-5/W2. pp. 73–78.

Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., Balakrishnan, H., 2008. The pothole patrol: Using a mobile sensor network for road surface monitoring. In: Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services. MobiSys '08. ACM, New York, NY, USA, pp. 29–39.

Evenepoel, S., Van Ooteghem, J., Verbrugge, S., Colle, D., Pickavet, M., 2014. On-street smart parking networks at a fraction of their cost: performance analysis of a sampling approach. Transactions on Emerging Telecommunications Technologies 25, 136–149.

Fabian, T., 2008. An algorithm for parking lot occupation detection. In: 7th Computer Information Systems and Industrial Management Applications. IEEE, pp. 165–170.

Farkas, K., Lendák, I., 2015. Simulation environment for investigating crowd-sensing based urban parking. In: 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). pp. 320–327.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., 1996. From data mining to knowledge discovery in databases. AI magazine 17 (3), 37.

Fitzner, D., Sester, M., Haberlandt, U., Rabiei, E., 2013. Rainfall estimation with a geosensor network of cars–theoretical considerations and first results. Photogrammetrie-Fernerkundung-Geoinformation 2013 (2), 93–103.

Ganti, R., Ye, F., Lei, H., November 2011. Mobile crowdsensing: current state and future challenges. IEEE Communications Magazine 49 (11), 32–39.

Ge, Y., Xue, W., Shu, Z., 2013. Improved System for ParkNet Mobile Network. In: Proceedings of the FISITA 2012 World Automotive Congress SE - 11. Vol. 200 of Lecture Notes in Electrical Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 131–145.

Geurts, P., 2005. Bias vs variance decomposition for regression and classification. In: Data Mining and Knowledge Discovery Handbook. Springer, pp. 749 – 763.

Guo, B., Yu, Z., Zhou, X., Zhang, D., 2014. From participatory sensing to mobile crowd sensing. In: International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops). IEEE, pp. 593–598.

Guo, Q., Wolfson, O., 2016. Probabilistic spatio-temporal resource search. GeoInformatica , 1–29.

Hampshire, R. C., Jordon, D., Akinbola, O., Richardson, K., Weinberger, R., Millard-Ball, A., Karlin-Resnik, J., 2016. Analysis of parking search behavior with video from naturalistic driving. Transportation Research Record: Journal of the Transportation Research Board 2543, 152–158.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learning. Springer.

He, H., Garcia, E. A., 2009. Learning from imbalanced data. IEEE Transactions on knowledge and data engineering 21 (9), 1263–1284.

Hofmann, S., 2017. Potential von LiDAR Mobile Mapping für hochgenaue Karten. Vol. 801 of Deutsche Geodätische Kommission: C (Dissertationen). München.

Horni, A., Montini, L., Waraich, R. A., Axhausen, K. W., 2013. An agent-based cellular automaton cruising-for-parking simulation. Transportation Letters 5 (4), 167–175.

Hu, Y., Kobourov, S. G., Veeramoni, S., 2012. Embedding, clustering and coloring for dynamic maps. In: Pacific Visualization Symposium (PacificVis). IEEE, pp. 33–40.

Ilarri, S., Wolfson, O., Delot, T., 2014. Collaborative sensing for urban transportation. IEEE Data Engineering Bulletin 37 (4), 3–14.

Joshi, A., James, M. R., 2015. Generation of accurate lane-level maps from coarse prior maps and lidar. IEEE Intelligent Transportation Systems Magazine 7 (1), 19–29.

Jossé, G., Schubert, M., Kriegel, H.-P., 2013. Probabilistic parking queries using aging functions. In: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, ACM Press, New York, New York, USA, pp. 452–455.

Klanner, F., Ruhhammer, C., 2015. Backendsysteme zur Erweiterung der Wahrnehmungsreichweite von Fahrerassistenzsystemen. In: Handbuch Fahrerassistenzsysteme. Springer, pp. 541–552.

Kleine-Besten, T., Kersken, U., Pöchmüller, W., Schepers, H., Mlasko, T., Behrens, R., Engelsberg, A., 2015. Navigation und Verkehrstelematik. In: Handbuch Fahrerassistenzsysteme. Springer, pp. 1047–1079.

Koetsier, C., 2017. Adaptives Rerouting von Taxis für Parkplatz-Crowd-Sensing mittels Apache Spark. Master's thesis, Leibniz Universität Hannover.

Kokolaki, E., Karaliopoulos, M., Stavrakakis, I., 2012. Opportunistically assisted parking service discovery: Now it helps, now it does not. Pervasive and Mobile Computing 8 (2), 210–227.

Kotb, A. O., Shen, Y.-C., Huang, Y., 2017. Smart parking guidance, monitoring and reservations: A review. IEEE Intelligent Transportation Systems Magazine 9 (2), 6–16.

Kuntzsch, C., Sester, M., Brenner, C., 2016. Generative models for road network reconstruction. International Journal of Geographical Information Science 30 (5), 1012–1039.

Lella, A., Lipsman, A., Martin, B., 2015. The 2015 U.S. Mobile App Report. `https://www.comscore.com/Insights/Presentations-and-Whitepapers/2015/The-2015-US-Mobile-App-Report`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5n6sDKP`].

Lendák, I., 2016. Mobile crowd-sensing in the smart city. In: European Handbook of Crowdsourced Geographic Information. London: Ubiquity Press, pp. 353–369.

Lendák, I., Farkas, K., 2016. How many drivers does it take to spot an OpenSpot? In: International Conference on Systems, Man, and Cybernetics (SMC). IEEE, pp. 401–406.

Li, X., Chuah, M. C., Bhattacharya, S., 2017. UAV assisted smart parking solution. In: International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, pp. 1006–1013.

Lin, T., Rivano, H., Mouël, F. L., 2017. A survey of smart parking solutions. IEEE Transactions on Intelligent Transportation Systems PP (99), 1–25.

Linne+Krause, 2016. Untersuchung zur Wirtschaftlichkeit des Taxigewerbes in der Bundeshauptstadt Berlin . `https://www.berlin.de/senuvk/verkehr/politik/taxi/download/untersuchung_wirtschaftlichkeit_taxi_berlin.pdf`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w63HW99y`].

Liu, R., Yang, Y., Kwak, D., Zhang, D., Iftode, L., Nath, B., 2017. Your search path tells others where to park: Towards fine-grained parking availability crowdsourcing using parking decision models. Proceedings of the ACM on Interactive, Mobile, Wearable Ubiquitous Technologies (IMWUT) 1 (3), 1–27.

Lookingbill, A., 2013. Predicting parking availability. US Patent 8,484,151.

Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., Huang, Y., 2009. Map-matching for low-sampling-rate GPS trajectories. In: Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems. ACM, pp. 352–361.

Ma, S., Wolfson, O., Xu, B., 2014. Updetector: sensing parking/unparking activities using smartphones. In: Proceedings of the 7th ACM SIGSPATIAL International Workshop on Computational Transportation Science. ACM, pp. 76–85.

Maimon, O., Rokach, L., 2005. Introduction to knowledge discovery and data mining. In: Data Mining and Knowledge Discovery Handbook. Springer, pp. 1–17.

Makridakis, S., Wheelwright, S. C., Hyndman, R. J., 2008. Forecasting methods and applications. John Wiley & Sons.

Maletic, J. I., Marcus, A., 2005. Data cleansing. In: Data Mining and Knowledge Discovery Handbook. Springer, pp. 21–36.

Margreiter, M., Mayer, P., 2015. A concept for crowdsourcing of in-vehicle data to improve urban on-street parking. In: 7th Mobil.TUM - International Scientific Conference on Mobility and Transport. pp. 1–8.

Masutani, O., 2015. A sensing coverage analysis of a route control method for vehicular crowd sensing. In: International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops). IEEE, pp. 396–401.

Mathur, S., Jin, T., Kasturirangan, N., Chandrasekaran, J., Xue, W., Gruteser, M., Trappe, W., 2010. Parknet: drive-by sensing of road-side parking statistics. In: Proceedings of the 8th international conference on Mobile systems, applications, and services. ACM, New York, NY, USA, pp. 123–136.

Matthaei, R., Reschka, A., Rieken, J., Dierkes, F., Ulbrich, S., Winkle, T., Maurer, M., 2015. Autonomes Fahren. In: Handbuch Fahrerassistenzsysteme. Springer, pp. 1139–1165.

Matzen, K., Snavely, N., 2014. Scene chronology. In: Proceedings of 13th European Conference on Computer Vision (ECCV), Part VII. Springer International Publishing, Cham, pp. 615–630.

Millard-Ball, A., Weinberger, R. R., Hampshire, R. C., 2014. Is the curb 80% full or 20% empty? Assessing the impacts of San Francisco's parking pricing experiment. Transportation Research Part A: Policy and Practice 63, 76–92.

Mitchell, T. M., 1997. Machine Learning, 1st Edition. McGraw-Hill, Inc., New York, NY, USA.

Monnier, F., Vallet, B., Soheilian, B., 2012. Trees detection from laser point clouds acquired in dense urban areas by a mobile mapping system. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. I-3. pp. 245–250.

Montini, L., Horni, A., Rieser-Schüssler, N., Axhausen, K. W., 2012. Searching for parking in GPS data. Working paper/Transport and Spatial Planning 780.

Navigation Data Standard (NDS) e.V., 2016. Navigation Data Standard - Format Specification - Version 2.5.0 beta. Tech. rep.

New York City Taxi&Limousine Commission, 2014. 2014 Taxicab Fact Book. `http://www.nyc.gov/html/tlc/downloads/pdf/2014_taxicab_fact_book.pdf`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5nroGx3`].

Ono, S., Kagesawa, M., Ikeuchi, K., 2002. A probe car for parking-vehicle detection by using laser range sensor. In: Intelligent Vehicles Symposium Proceedings. Vol. 2. IEEE, pp. 322–327.

OpenStreetMap, 2017a. Key:highway. `http://wiki.openstreetmap.org/wiki/Key:highway`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5qxUquq`].

OpenStreetMap, 2017b. Stats. `http://wiki.openstreetmap.org/wiki/Stats`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5rcWw8z`].

Park, W.-J., Kim, B.-S., Seo, D.-E., Kim, D.-S., Lee, K.-H., 2008. Parking space detection using ultrasonic sensor in parking assistance system. In: Intelligent Vehicles Symposium Proceedings. pp. 1039–1044.

Pfeiffer, H.-W., 2017. SCRUM bei Robert Bosch Car Multimedia? Erfahrungen nach drei Jahren in einem agilen Projekt. `http://archiv2.scrum-day.de/files/scrum-day/content/upload/c4p-vortraege-2015/SCRUMDAY2015-Pfeiffer-Robert-Bosch-Car-Multimdia-GmbH.compressed.pdf`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5lTXvvT`].

Pflügler, C., Köhn, T., Schreieck, M., Wiesche, M., Krcmar, H., 2016. Predicting the availability of parking spaces with publicly available data. In: GI-Jahrestagung. pp. 361–374.

Piorkowski, M., Sarafijanovic-Djukic, N., Grossglauser, M., 2009. CRAWDAD dataset epfl/mobility (v. 2009-02-24). Downloaded from `http://crawdad.org/epfl/mobility/20090224`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5m7yexy`].

Protschky, V., Ruhhammer, C., Feit, S., 2015. Learning traffic light parameters with floating car data. In: IEEE 18th International Conference on Intelligent Transportation Systems. pp. 2438–2443.

Rajabioun, T., Ioannou, P., 2015. On-street and off-street parking availability prediction using multivariate spatiotemporal models. IEEE Transactions on Intelligent Transportation Systems 16 (5), 2913–2924.

Richter, F., Di Martino, S., Mattfeld, D. C., 2014. Temporal and spatial clustering for a parking prediction service. In: 26th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, pp. 278–282.

Riegl LMS GmbH, 2012. Data Sheet - Riegl VMX-250. `http://www.riegl.com/uploads/tx_pxpriegldownloads/10_DataSheet_VMX-250_20-09-2012.pdf`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5nJs8d6`].

Robert Bosch GmbH, 2016. Bosch Community-based parking. `http://www.bosch-mobility-solutions.com/en/connected-mobility/community-based-parking/`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5mxBVSY`].

Rokach, L., 2005. Ensemble methods for classifiers. In: Data Mining and Knowledge Discovery Handbook. Springer, pp. 957 – 980.

Rokach, L., Maimon, O., 2005a. Clustering methods. In: Data Mining and Knowledge Discovery Handbook. Springer, pp. 321 – 352.

Rokach, L., Maimon, O., 2005b. Decision trees. In: Data Mining and Knowledge Discovery Handbook. Springer, pp. 165 – 192.

Rousseeuw, P. J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20, 53–65.

Rybarsch, M., Aschermann, M., Bock, F., Goralzik, A., Köster, F., Ringhand, M., Trifunović, A., 2017. Cooperative Parking Search: Reducing Travel Time by Information Exchange Among Searching Vehicles. In: IEEE 20th International Conference on Intelligent Transportation Systems.

San Francisco County Transportation Authority, 2009. On-street parking management and pricing study. `http://www.sfcta.org/sites/default/files/content/Planning/ParkingManagementStudy/pdfs/parking_study_final.pdf`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5mxoOOx`].

San Francisco Municipal Transportation Agency, 2014. SFpark: Putting theory into practice. Pilot project summary and lessons learned. `http://sfpark.org/resources/docs_pilotsummary/`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5otWgQf`].

Schlichting, A., Brenner, C., 2016. Generating a hazard map of dynamic objects using lidar mobile mapping. Photogrammetric Engineering & Remote Sensing 82 (12), 967–972.

Schmidt, M., Kwella, B., Pieth, N., Nov 2009. TPEG löst TMC ab Eine neue Generation der Navigation. ATZelektronik 4 (6), 30–35.

Shepard, D., 1968. A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of the 1968 23rd ACM national conference. ACM Press, New York, New York, USA, pp. 517–524.

Sherwin, I., 2011. Google labs' open spot: A useful application that no one uses. `http://www.androidauthority.com/google-labs-open-spot-a-useful-application-that-no-one-uses-15186/`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5m6dQia`].

Shoup, D., 2006. Cruising for parking. Transport Policy 13 (6), 479 – 486.

Shoup, D., 2007. Cruising for parking. Access 30 (30), 16–22.

Shumway, R. H., Stoffer, D. S., 2010. Time series analysis and its applications: with R examples. Springer Science & Business Media.

Siemens, 2017. Siemens startet in Berlin Pilotprojekt zur Parkplatzsuche per Radar. `https://www.siemens.com/press/pool/de/pressemitteilungen/2015/mobility/PR2015090340MODE.pdf`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5sS3n57`].

Sivaraman, S., Trivedi, M. M., 2013. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. IEEE Transactions on Intelligent Transportation Systems 14 (4), 1773–1795.

Stenneth, L., Wolfson, O., Xu, B., Yu, P. S., 2012. Phonepark: Street parking using mobile phones. In: 2012 IEEE 13th International Conference on Mobile Data Management. IEEE, pp. 278–279.

Tamrazian, A., Qian, Z., Rajagopal, R., 2015. Where is my parking spot? Online and offline prediction of time-varying parking occupancy. Transportation Research Record: Journal of the Transportation Research Board , 77–85.

Tasseron, G., Martens, K., 2017. Urban parking space reservation through bottom-up information provision: An agent-based analysis. Computers, Environment and Urban Systems 64 (Supplement C), 30 – 41.

Tasseron, G., Martens, K., Van der Heijden, R., 2015. The potential impact of vehicle-to-vehicle and sensor-to-vehicle communication in urban parking. Intelligent Transportation Systems Magazine, IEEE 7 (2), 22–33.

Tasseron, G., Martens, K., van der Heijden, R., 2016. The potential impact of vehicle-to-vehicle communication on on-street parking under heterogeneous conditions. IEEE Intelligent Transportation Systems Magazine 8 (2), 33–42.

Teodorović, D., Lučić, P., 2006. Intelligent parking systems. European Journal of Operational Research 175 (3), 1666–1681.

Thornton, C., Hutter, F., Hoos, H. H., Leyton-Brown, K., 2013. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 847–855.

Thornton, D. A., Redmill, K., Coifman, B., 2014. Automated parking surveys from a LIDAR equipped vehicle. Transportation Research Part C: Emerging Technologies 39, 23–35.

TomTom, 2017. Neueste Karten für Navigationsgeräte. `http://de.support.tomtom.com/app/answers/minor_detail/a_id/7908/`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5ltxkQK`].

Van Ommeren, J. N., Wentink, D., Rietveld, P., 2012. Empirical evidence on cruising for parking. Transportation Research Part A: Policy and Practice 46 (1), 123–130.

Vlahogianni, E. I., Kepaptsoglou, K., Tsetsos, V., Karlaftis, M. G., 2016. A real-time parking prediction system for smart cities. Journal of Intelligent Transportation Systems 20 (2), 192–204.

Vogel, P., Greiser, T., Mattfeld, D. C., 2011. Understanding bike-sharing systems using data mining: Exploring activity patterns. In: Procedia - Social and Behavioral Sciences. Vol. 20. pp. 514 – 523.

Weinberger, R., Millard-Ball, A., Hampshire, R. C., 2017. Parking search-caused congestion: Where's all the fuss? In: TRB 96th Annual Meeting Compendium of Paperss.

White, P., 2007. No Vacancy: Park Slope's Parking Problem and how to fix it. `https://www.transalt.org/sites/default/files/news/reports/2007/novacancy.pdf`, [Online; accessed: 2017-12-30; archived by WebCite® at `http://www.webcitation.org/6w5s6wMJc`].

Wolf, D. F., Sukhatme, G. S., 2005. Mobile robot simultaneous localization and mapping in dynamic environments. Autonomous Robots 19 (1), 53–65.

Wu, E.-K., Sahoo, J., Liu, C.-Y., Jin, M.-H., Lin, S.-H., 2014. Agile urban parking recommendation service for intelligent vehicular guiding system. IEEE Intelligent Transportation Syst. Mag. 6 (1), 35–49.

Xu, B., Wolfson, O., Yang, J., Stenneth, L., Yu, P. S., Nelson, P. C., 2013. Real-time street parking availability estimation. In: 14th International Conference on Mobile Data Management (MDM). Vol. 1. IEEE, pp. 16–25.

Zheng, Y., Rajasegarar, S., Leckie, C., 2015. Parking availability prediction for sensor-enabled car parks in smart cities. In: 10th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). pp. 1–6.

Zheng, Y., Rajasegarar, S., Leckie, C., Palaniswami, M., 2014. Smart car parking: Temporal clustering and anomaly detection in urban car parking. In: Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). IEEE, pp. 1–6.

Zucker, S. W., 1976. Region growing: Childhood and adolescence. Computer Graphics and Image Processing 5 (3), 382 – 399.

Acknowledgements

In these closing lines, I express my gratitude to those persons having contributed to the success of this work.

First of all, I thank Prof. Dr.-Ing. Monika Sester most sincerely for the great supervision and continuous support. In particular, I wish to thank for the helpful ideas and discussions on this work and for providing the setting for an inspirational and unbounded research atmosphere.

I thank the referees Prof. Dr.-Ing. Christian Heipke and Prof. Dr.-Ing. Bernhard Friedrich for the feedback and assessment of this work.

A special thanks goes to Prof. Dr. Sergio Di Martino for the successful collaboration, the regular exchange, and the fruitful research stay in Naples. Furthermore, I wish to thank apl. Prof. Dr.-Ing. Claus Brenner for the detailed conversations on methods and the joint supervision of several masters' theses.

The research for this work was conducted within the framework of the DFG-funded research training group SocialCars. I am very grateful for the introductory lectures, discussions, regular feedback, and last but not least for the financial support.

I really enjoyed working both at the Institute of Cartography and Geoinformatics and in the SocialCars project. I thank all my colleagues for making the working days very pleasant and enjoyable. Thank you for the great support whenever I needed it, the joint breaks, and the entertaining events. Among all, I wish to particularly thank Paul Czioska for helping me with geo-related questions, for the helpful feedback on this thesis, and the fun sport activities. Further thanks goes to Martin Werner for proofreading this work and for the extensive exchange on Big Geospatial Data. Moreover, I thank Yuri Attanasio, Steffen Axer, Daniel Eggert, Jason Liu, Antonio Origlia, and Karen Xia for the close collaboration, which contributed to parts of this work.

Finally, I wish to thank my family and friends for their support and their interest in my work. In particular, I thank my brother Julian Bock for all the discussions on driving assistance systems and machine learning and my brother Adrian Bock for proofreading this work. A special thanks goes to Rebecca Funken for going with me through these exciting years of research, traveling, and fun.

Curriculum vitae

Personal details

Name	Urs Fabian Bock
Date of birth	March 2, 1987
Place of birth	Lebach, Germany
Address	Vondelstr. 50 50677 Köln, Germany

Professional experience

06/2014 – 03/2018	**Research assistant at Leibniz University Hannover, Germany, Institute of Cartography and Geoinformatics** DFG Research Training Group SocialCars
05/2012 – 05/2014	**Research engineer at Robert Bosch GmbH in Hildesheim, Germany**

Eduction

06/2014 – 03/2018	**PhD student at Leibniz University Hannover, Germany**
09/2016 – 11/2016	Research stay at Università Federico II in Naples, Italy
10/2008 – 12/2011	**German Diploma in Physics at Heidelberg University, Germany**
08/2011 – 10/2011	Research stay at National Central University in Zhongli, Taiwan
07/2009 – 10/2009	Research internship at York University in Toronto, Canada
10/2006 – 09/2008	**Studies of Physics (Vordiplom) at Saarland University, Germany**
08/1997 – 07/2006	**Secondary education (Abitur) at Geschwister-Scholl-Gymnasium Lebach, Germany**
10/2004 – 09/2006	‘Juniorstudium’ in Engineering at Saarland University, Germany

Other work experience

02/2010 – 11/2010	Student assistant at the Interdisciplinary Center for Scientific Computing, Heidelberg University, Germany
07/2007 – 08/2007	Intern at CANway technology, Munich, Germany

Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover

(Eine vollständige Liste der Wiss. Arb. ist beim Geodätischen Institut, Nienburger Str. 1, 30167 Hannover erhältlich.)

Nr. 315	KERSTEN, Tobias:	Bestimmung von Codephasen-Variationen bei GNSS-Empfangsantennen und deren Einfluss auf Positionierung, Navigation und Zeitübertragung (Diss. 2014)
Nr. 316	BISKUPEK, Liliane:	Bestimmung der Erdorientierung mit Lunar Laser Ranging (Diss. 2015)
Nr. 317	STEINER, Christina:	Highspeed Stereo-Endoskopie für eng begrenzte Messvolumina (Diss. 2015)
Nr. 318	BANDIKOVA, Tamara:	The role of attitude determination for inter-satellite ranging (Diss. 2015)
Nr. 319	LIN, Miao:	Regional gravity field recovery using the point mass method (Diss. 2015)
Nr. 320	ZHANG, Lijuan:	Mining GPS-Trajectory Data for Map Refinement and Behavior Detection (Diss. 2015)
Nr. 321	ZADDACH, Sebastian:	Zum Beitrag Bayesscher Schätzverfahren in der Vergleichswertermittlung (Diss. 2016)
Nr. 322	SMYRNAIOS, Marios:	Carrier-phase Multipath in Satellite-based Positioning (Diss. 2016)
Nr. 323	MENZE, Moritz:	Object Scene Flow (Diss. 2016)
Nr. 324	WU, Hu:	Gravity field recovery from GOCE observations (Diss. 2016)
Nr. 325	XU, Xiangyang:	Terrestrial Laser Scanning for the Generation and Calibration of Finite Element Models (Diss. 2016)
Nr. 326	SAYYAD, Muhammad Naeem Shahzad:	Joint use and mutual control of terrestrial laser scans and digital images for accurate 3D measurements (Diss. 2016)
Nr. 327	SCHACK, Lukas:	Object-based matching of Persistent Scatterers to Optical Oblique Images (Diss. 2016)
Nr. 328	REICH, Martin:	Global Image Orientation from Pairwise Relative Orientations (Diss. 2016)
Nr. 329	KLINGER, Tobias:	Probabilistic multi-person localisation and tracking (Diss. 2016)
Nr. 330	SCHMIDT, Alena:	Markierte Punktprozesse für die automatische Extraktion von Liniennetzen in Rasterdaten (Diss. 2016)
Nr. 331	HOFMANN, Franz:	Lunar Laser Ranging - verbesserte Modellierung der Monddynamik und Schätzung relativistischer Parameter (Diss. 2017)
Nr. 332	BRIEDEN, Phillip:	Validierung von GOCE-Gravitationsgradienten in Kreuzungspunkten und Zukunftsperspektiven der Satellitengradiometrie (Diss. 2017)
Nr. 333	VON GÖSSELN, Ilka:	Simulationsbasierte Effizienzoptimierung von Messprozessen am Beispiel der tachymetrischen Netzmessung (Diss. 2017)
Nr. 334	HOFMANN, Sabine:	Potential von LiDAR Mobile Mapping für hochgenaue Karten (Diss. 2017)
Nr. 335	ALBERT, Lena:	Simultane Klassifikation der Bodenbedeckung und Landnutzung unter Verwendung von Conditional Random Fields (Diss. 2017)
Nr. 336	NIEMEYER, Joachim:	Verwendung von Kontext zur Klassifikation luftgestützter Laserdaten urbaner Gebiete (Diss. 2017)
Nr. 337	EGGERT, Daniel:	Effiziente Verarbeitung und Visualisierung von Mobile Mapping Daten (Diss. 2017)
Nr. 338	FITZNER, Daniel:	Estimation of Spatio-Temporal Moving Fields at High Resolution (Diss. 2017)
Nr. 339	CZIOSKA, Paul:	Meeting point locations for shared rides (Diss. 2018)
Nr. 340	FEUERHAKE, Udo:	Erfassung von Trajektorien und Erkennung von Bewegungsmustern (Diss. 2018)
Nr. 341	HUANG, Hai:	Bayesian Models for Pattern Recognition in Spatial Data (Habil. 2018)
Nr. 342	BOCK, Fabian:	Dynamic Parking Maps from Vehicular Crowdsensing (Diss. 2018)

Die Arbeiten werden im Rahmen des wissenschaftlichen Schriftenaustausches verteilt und sind nicht im Buchhandel erhältlich. Der Erwerb ist zu einem Stückpreis von € 25,00 bei den herausgebenden Instituten möglich.