Author
Markus Hiesmair

Submission
Institute of Telecooperation

Thesis Supervisor
Karin Anna Hummel

February 2018

ROBUST DRIVE-BY ROAD SIDE PARKING DETECTION ON MULTI-LANE STREETS USING AN OPTICAL DISTANCE SENSOR

Master's Thesis

to confer the academic degree of

Master of Science

in the Master's Program

Computer Science

Affidavit

I hereby declare that the following dissertation "Put your thesis title here" has been written only by the undersigned and without any assistance from third parties.

Furthermore, I confirm that no sources have been used in the preparation of this thesis other than those indicated in the thesis itself.

Linz, on November 18, 2017

Markus Hiesmair

Acknowledgment

Summary

Summary . . .

asdf

Abstract

Abstract . . .

Contents

Abbreviations

CNN Convolutional Neural Network

GPS Global Positioning System

LIDAR Light detection and ranging

List of Figures

List of Tables

Chapter 1

Introduction

Traffic congestion in urban areas became a bigger problem every year in the past decades. Increasing traffic causes several issues, for example high monetary and environmental costs by using gasoline and by emitting CO_2 to the environment. There are several strategies to reduce urban traffic to mitigate these problems like for example new investments in public transport infrastructure. However, the usage of private cars in cities won't stop in the next decades, but increase. According to a study by Hao et al. [6] car sales will continue to grow in the upcoming decades and therefore also traffic will continue to become worse in big cities. Thus, it is important to find ways to reduce traffic in urban areas.

With more vehicles driving in urban areas, there also comes the need for a sufficient number of parking spaces. Finding parking spaces in urban areas can be a really difficult, frustrating and time consuming task for drivers. There often exists some information about the availability of parking spaces in parking garages, but in most cities the situation of road side parking is rather non-transparent. This not only leads to frustrated drivers, who are searching for parking spaces a long time, but again contributes to urban traffic congestion as many cars have to drive around close to their destination while searching for free parking spaces.

There are many studies, which show that the searching for parking spaces adds a lot of traffic. In 2013 a study by Nawaz et al. [14] showed that about 30% of traffic congestion is created by drivers looking for free parking spaces. Another study [7] found that alone in 2007 searching for parking spaces caused costs of about 78 billion US dollars by using 2.9 billion gallons of wasted gasoline and 4.2 billion lost hours only in the United States. Furthermore, this obviously causes a lot of CO_2 emissions which is not only bad for the environment and contributes to climate change, but also lowers the quality of living in big cities through the significant amount of air pollution.

One of the most important contributors to high search times for parking spaces is not only the lack of vacant parking spaces, but also the lack of information, if and where free parking spaces are available. Therefore, one way to mitigate many of the above stated problems is to determine the current parking space situation in the city and make it accessible to the public (e.g. via web application), so that drivers can efficiently navigate to a vacant parking space, or even decide if they want to go by car or use public transportation, depending on the number of parking spaces available close to their destination.

However, detection of road side parking spaces and their states is a challenging task. Of course an obvious approach to the problem would be to put stationary sensors to every parking space in the city, which check, if the corresponding parking space is occupied or vacant. This, however, has the drawback to be very expensive as, for big cities, thousands of sensors would have to be bought, installed and maintained. Furthermore, because the state of parking parking spaces does not often change, the high frequency of sensing with such a system would be rather inefficient.

1.1 Drive-By park sensing

A promising novel approach to sense a city's parking situation is the use of mobile sensors instead of static ones. Crowd sensing has the advantage to be usually more cost effective and can provide sufficient accuracy for the purpose of providing parking space availability maps.

There are several approaches to mobile parking availability sensing, which will be discussed in chapter 2. This thesis will focus on "drive-by park sensing". The idea of drive-by sensing approaches is that there are sensing vehicles which are driving through the city and collecting data of their environment through mounted sensors. Using the collected data parking cars and vacant parking spaces should be detected. There already exists a prototype implementation of such a system. In 2013, Mathur et al. [12] presented their system, called ParkNet, which continuously measured the distance to the nearest obstacle on the right side of the road, as well as the location of the sensing vehicle through a GPS sensor. Using these data, they used thresholding to detect parking cars and vacant parking spaces. A more detailed description of their work is being described in section **??**.

Figure 1.1 shows a standard drive-by scenario of a sensing vehicle which passes two parallel parked cars and a vacant parking space in between them. The distance mea-

Figure 1.1: The sensing vehicle passes two parked cars and should identify a vacant spot in between using distance and location measurements. **To-Do:** Hier Referenz zu Parknet-Paper??? Ähnliche Grafik...

surements while passing the parked cars will be much shorter than the measurements taken while passing the vacant parking space. This should allow a basic algorithm to recognize parking cars and vacant parking spaces.

However, the situation which is shown in figure 1.1 is an idealistic one. In real life traffic, there will be much more complex situations to face, which are not as easily detectable and which might influence the success of the detection. For instance, the sensing vehicle might not drive in the right-most lane, therefore the measured distances will be much longer. Another possible issue are other driving cars, motorcycles or bicycles which the sensing car overtakes. There are many of such distractions which have to be filtered out to ensure that there are as few false detections as possible. The existing ParkNet prototype does not take any of such situations into account, because it was only evaluated on parallel parking cars on single lane roads.

The high complexity of urban traffic and the many distractions during sensing make it nearly impossible to create a rule set based on the sensor measurements which would be able to detect parking situations at a sufficient accuracy. Therefore, simple thresholding as applied in the ParkNet prototype will not work in real life traffic scenarios. Furthermore, such rule sets would have to be created for each city individually because of the different nature of the roads and the parking spaces. For instance, the distance

between roads and parking spaces can vary highly in two different cities. Thus, to be able to detect parking situations, new methods have to be found, which should be more flexible to distractions and changes in the environment.

1.2 Research goals and methodology

The overall goal of this thesis is to determine which accuracy can be achieved for detecting the parking space situation in urban real world traffic scenarios using drive-by park sensing. Furthermore, distracting situations should be identified and strategies to cope with them should be found to be able to keep the detection accuracy high. In contrast to existing approaches in this thesis machine learning will be applied to classify different road traffic and parking scenarios to be able to better cope with the high variety of situations possible. As a first step, sensor data has to be collected in real world traffic scenarios including distance sensor measurements and GPS locations of the sensing vehicle. To be able to apply supervised learning all sensor measurements have to get assigned their ground truth manually. As next step different machine learning algorithms will be applied, evaluated and compared to each other in terms of accuracy for the different classes. The detailed methodology which will be followed includes the following steps:

Building a test bed As a first step a test bed has to be built, which is able to access the required sensors to record all the necessary data. A Raspberry Pi will be used as processing device because of its popularity and the many compatible sensors which work with this platform. It is connected to a LIDAR-Lite v3 sensor which continuously measures the distance to the nearest obstacle on the right side of the road. A GPS receiver will track the location of the sensing vehicle and a camera will be used to record images of the ground truth for evaluation purposes only. In section 3.1 the complete setup of the test bed is described as well as all the specific hard ware parts and their abilities.

Acquiring a dataset As soon as the test bed is ready, the sensors should be mounted on the prototype car to be able to start recording the dataset. Test drives should be done in some selected streets in Linz, Austria with the focus of variety of the recorded situations. The test scenes should include single lane as well as multi lane streets and measurements in all streets should be done several times. Furthermore, the car should be driving as it would in regular traffic (not only in the right most lane, etc...) and the scenes should also include high and low traffic scenarios to have a high amount of diverse data. All measured distances, GPS

locations and ground truth images have to be saved to files with the according timestamps to be able to evaluate the results of different approaches later on. Furthermore, using the images taken by the camera, ground truth values will be manually labelled in different classes (Parked car, free space, overtaken car, ...). A detailed description about the dataset and the ground truth tagging can be found in section **To-Do:** ref **??**.

Data processing and segmentation As next step the measured sensor values have to be preprocessed and filtered in order for the following algorithms to work. Sensing and overflow errors as well as outliers in the measurements should be identified and removed before further processing. After the raw data has been filtered, the sensor data has to be segmented. As parking cars and other cases which should be classified consist of several sensor measurements, the corresponding sensor measurements should be grouped together and merged to segments which will be later classified. All preprocessing steps and the segmentation process are described in more detail in section **To-Do:** ref 3.3.

Classification using basic machine learning techniques Features on the created segments are calculated (for instance length, average distance and variation of the distances) and are being used to train and evaluate several machine learning algorithms which will be compared on their performance. Furthermore, some deep learning models will also be evaluated on the raw sensor data of the segments and will be compared to common machine learning results. The results of all experiments can be found in section **To-Do:** ref **??**.

Further improvements **To-Do:** todo

Chapter 2

Related Work

This chapter will cover work related to park sensing and machine learning. Section 2.1 will discuss different approaches to sensing current parking situations in cities. Furthermore, a comparison of them will be given as well as advantages and disadvantages of the specific approaches. In section 2.2 the need of parking space maps will be discussed as well as how they can be obtained using several sensing vehicles. **To-Do:** Machine Learning...

2.1 Approaches to park sensing

There already exist numerous approaches for detecting the states of parking spaces as well as the parking situation in a city. In this section parking detection approaches will be categorized in six different categories and in the following subsections several reference papers for all categories will be discussed. The first section about stationary park sensing (section 2.1.1) will reference approaches where sensors are stationary deployed per parking space. Section 2.1.2 will cover camera-based systems which detect parking spaces in a dedicated parking area. Counting in- and outgoing vehicles to coarsely estimate parking space counts is discussed in section 2.1.3. Another approach is to detect certain events (for instance parking and unparking) to estimate parking space counts. Section 2.1.4 will discuss several related papers in this area. Drive-by sensing will be discussed in sections 2.1.5 and 2.1.6. Using cameras to detect parking spaces while driving by will be referenced in section 2.1.5, while section 2.1.6 will cover using distance sensors. Limitations of both drive-by sensing approaches will be discussed in section 2.1.7 and an overall comparison of all discussed approaches will be given in section 2.1.8.

2.1.1 Stationary park sensing

The most obvious and technically simple solution to park sensing is to use stationary sensors to determine the state of parking spaces. Usually one sensor per parking space is used which determines its state (occupied or vacant) and sends it to a central server. Several reference projects already exist [16, 15] implementing this technique. In this section as a representation of all the similar systems, the SFpark project will be examined.

In San Francisco a first prototype of the SFpark project [16] has been implemented from 2008 to 2011. In specific down town areas in San Francisco with high amounts of traffic, wireless stationary sensors have been placed at about 8.200 road side parking spaces. These sensors are able to detect the state of one parking space in real time and send this information to a central server. Furthermore, parking garages also count the in- and outgoing vehicles and shared this information, so that the parking situation in the areas with park sensing can be derived. The gained information is being shared as open data, so third party developers and researchers can also use the dataset for all kind of projects and purposes. Furthermore, there exists an App from SFpark itself to help drivers find available parking spaces nearby, navigate to them and pay as they go with their phones.

A top priority goal of the SFpark project is to increase the availability of parking spaces in every block throughout the city. To achieve this goal, they are using demand responsive pricing. If (almost) all parking spaces in a neighbourhood are occupied for a long time, they raise the price in this specific area and vice versa if no parking spaces are occupied they lower prices. This leads to high overall parking space availabilities (20 - 40%) and also to lower traffic congestion and lower greenhouse gas emissions. However, besides the advantages of high accuracy and being a real time system, there also are disadvantages. First of all, only metered parking spaces can be tracked, as sensors have to be installed per parking space, so areas where parking is allowed but there are no clearly marked parking spaces cannot be sensed with a reasonable accuracy. Furthermore, another big drawback are the high overall costs. The about 8.200 stationary sensors have to be bought, installed and maintained, which obviously causes high costs while only covering a tiny fraction of the overall San Francisco down town. So if the system should be available in the whole down town, costs would increase dramatically.

2.1.2 Stationary park sensing using cameras

Another approach while using stationary sensors is to use fixed deployed cameras which continuously record images of parking areas and analyze them for vacant parking spaces. Cameras can monitor up to one hundred parking spaces simultaneously with an accuracy up to 96%. Challenges of image detection are of course different lightning and weather conditions as well as occlusions depending on the angle which the camera records the parking scene. Detection algorithms using standard digital image processing will be discussed in this section as well as approaches using deep learning and convolutional neural networks (CNNs).

Parking detection using digital image processing

There exist a few common approaches using digital image processing to detect the state of parking spaces using a captured image of a parking area. First of all, edge detection is often used for parking space classification. Common edge detectors such as Canny Edge Detector or Sobel can be used to derive the edge pixels of an image. As next step the edges or edge pixels are counted and if they are above a certain threshold, the space will be detected as occupied. The assumption behind this approach is, that usually a vacant parking space has a plain surface and therefore a low amount of edges whereas a image of a parking car should have a lot of edges. Blumer et al. [2] and Liu et al. [10] both used this technique as part of their algorithms.

Often a slightly different yet related approach is also used, namely object counting [10]. The edges of the image segment of a parking space are analyzed and closed contours (treated as objects) are detected and counted. Then depending on a threshold which has to be set first, a parking space will be classified as either vacant or occupied depending on the object count.

Another common image processing technique is to use foreground/background information of the images. Using this approach the main background color is being identified and compared to the whole image. The background of a parking space can either be defined via extracting a certain part of an image which should always represent the main background color of a parking space's pavement (done by Blumer et al. [2]) or via an histogram of the image assuming that the background uses the most pixels in a recorded image (done by Liu et al. [10]). After the background color is available it is being subtracted from the original image and using thresholding foreground and background pixels are identified and counted. Depending on the count of the respective pixels, the parking space is then classified as vacant or occupied.

Liu et al. [10] used all of the above mentioned techniques in combination to build a more stable prototype. They only tested their prototype indoors which is why weather and lightning conditions were no problems. With sensing only a maximum of seven cars, it provided an ensemble technique which should be more reliable. However, they did not include a statistic of how well their algorithm performed on a bigger dataset. Another ensemble method was developed by Blumer [2]. They used edge counts and background/foreground information as input for different machine learning techniques, which should then classify a parking space as vacant or occupied. Their plain algorithm achieved an accuracy of about 77.8%. However, they improved it using frames of preceding and following images to identify parking/unparking events and then achieved an accuracy of about 88.8%.

Parking detection using deep learning and CNNs

In recent years deep learning, which is a specific variant of neural networks, gained a lot of significance in the field of machine learning. For example in image recognition, convolutional neural networks (CNNs) preform nowadays as well as humans in classifying everyday objects in digital images. A CNN is a neural network with a possibly large amount of hidden layers of which some of them are convolutional layers, which take into account the spatial relationships between neighbouring pixels (in the case of image recognition). This rise in learning power also leads to projects which try to train CNNs in classifying the states of parking spaces while using camera images of a parking area.

In 2016 Amato et al. [1] trained two different CNNs on classifying parking space states and compared their performances afterwards. They used the publicly available PKLot dataset as well as a dataset collected by themselves to train and evaluate their neural networks. In total they had over 700.000 images in different weather conditions and with different levels of quality (in some situations parking lots were almost fully occluded by trees or lamp posts, etc.). Results of the CNN's classifications were promising. Despite the fact that some of the images did not exactly match the parking space and there were a lot of occlusions, the best performing deep neural network gained an accuracy of above 91% on all subsets of the datasets.

Another work in the field of deep learning has been done by Di Mauro also in 2016 [4]. They also used CNNs and the PKLot dataset as well as a self acquired dataset. However, they had a slightly different CNN as they also used a technique called pseudo-labelling of the data which can be seen as semi-supervised approach. When using pseudo-labelling both labelled and unlabelled data are used at the same time to train the network. For

unlabelled data the label which was computed by the CNN in the forward pass is used, that is also why it is called pseudo-labelling. Furthermore, a different loss function has to be used as the pseudo labelled data has much less significance than the ground truth labelled data. Di Mauro et al. trained the CNN with about 5% of the data and achieved an accuracy of above 96% with a the fully supervised approach. The pseudo-labelling approach reached about the same level of accuracy, however, the dataset had to be balanced for it to work properly.

Advantages and disadvantages of using stationary cameras

As described in this section, there exists already a lot of research how cameras can be used to classify the states of parking spaces. Advantages of cameras are the high level of accuracy they reach while covering a lot of cars at once. However, for all of the above discussed approaches to work cameras would have to manually calibrated before usage to know where parking spaces are. Furthermore, most of the research focused on the use of cameras for detection the parking space situation of a single parking lot with many cars rather than for road side parking spaces. For this approach to work at a city wide scope, cameras would have to be mounted at least at every block. Similar to the dedicated sensors per parking space (section 2.1.1) this would cause high costs not only in the form of hardware but also in installation and maintenance costs. Furthermore, mounting cameras throughout the city obviously brings up privacy issues.

2.1.3 Counting in- and outgoing vehicles

A rather simple approach of estimating the number of free parking spaces is to count in- and outgoing vehicles at parking zones. Zadeh et al. [19] developed a prototype which counts all in- and outgoing vehicles using a Raspberry Pi and two ultrasonic range finders at each gate. Ultrasonic sensors measure the distance from the side of the road to a potentially passing vehicle. Two ultrasonic sensors are needed to detect if a vehicle is going in or out. According to the order in which the ultrasonic sensors detect a passing vehicle it is decided whether a vehicle is going in or out. Furthermore, the two sensors are also used to differentiate a passing person from a vehicle. Both sensors are as far apart as they can detect a vehicle at the same time but not a person. Misleading detections of passengers are filtered using this technique.

As described this approach is technically simple to develop and the hardware costs are also quite cheap. However, this system is only designed for parking lots which have a low number of gates where cars go in and out and not for a city wide deployment. For

closed parking areas it is easy to detect in- and outgoing vehicles whereas it is hard to do so in open traffic and street scenarios. Furthermore, the counting also will not work with ultrasonic sensors in open street scenarios. For example, there are often several lanes, which would make the sensing impossible as passing cars would possibly occlude each other.

2.1.4 Event detection based park sensing using smartphones

Detecting parking and unparking events using a driver's smartphone is another approach to estimate parking space availabilities, which is especially interesting because nowadays almost everybody has a personal smartphone. As soon as these parking related information is available for a high enough rate of drivers, a city's parking availability situation can be roughly estimated. There already exist a few research papers which discuss different approaches to this topic. Three of these will be discussed in this section.

In 2013 Nawaz et al. developed ParkSense [14] which is an application for Android phones to detect certain events which relate to the change of parking situations. Using the app a user could log his parking events and furthermore the costs of parking. The app was released to the Google Play Store and recorded 59 parking traces in four different cities at the time their paper was released. However, the app not only recorded the times when parking and unparking took place, but also recorded a detail profile of the Wifi access points while parking as well as Wifi profiles and GPS locations after unparking took place. Using this dataset, the goal was to evaluate how accurately unparking events could be detected using Wifi fingerprints. ParkSense takes several Wifi fingerprints when parking takes place and then continuously scans the Wifi signal and compares it to the signal while parking. Using this technique ParkSense could detect when users returned to their vehicle. However, it cannot be assumed that when a user returns close to his vehicle, that he will leave with his car, therefore Nawaz et al. also implemented an activity detection for driving using the change of Wifi signals over a specific timespan. The idea behind this approach is, that while driving, Wifi access points in range will change much more frequently than while walking or while staying at the same place. Out of 41 cases where unparking actually took place ParkSense could detected 38 user returns correctly and 35 times out of the 41 it could detect that the vehicle started driving. Thus, ParkSense reached an overall accuracy of about 85% in detecting unparking events.

A similar approach to the problem of parking related event detection has been done by Ma et al. [11] in 2014. The overall goal of this research paper was to robustly identify

parking and unparking events using different sensors of a typical smartphone. They identified in total nine indicators which could give hints whether a parking/unparking event occurred. For instance, they used several accelerometer measurements to identify the change of the user from walking to driving and vice versa. Other indicators used were the Bluetooth signal strength of the car, sounds of a started motor sensed via a smartphone's microphone, Wifi fingerprints and accessing parking payment apps. To decrease the amount of energy consumed by the application, certain low power sensors (e.g. accelerometer) measured periodically while other high consumption sensors (e.g. microphone) only confirmed or rejected events which were already sensed by low power sensors. Using all indicators, Ma et al. built a model based on conditional probabilities which should be able to show the probability of an event at a certain time. To evaluate their approach, they developed an Android application which they used to manually collect their ground truth events. They then used 40 parking samples to train their model and tested it on 60 other samples. Their best configuration achieved 93% recall and 90% precision on parking events and 81% recall and 93% precision on unparking events.

Estimating events by non-tracked drivers

All of the above mentioned approaches would only work if almost all drivers in a specific city would have had the respective smartphone app installed, so that all parking and unparking events could be detected. That way the states of all parking spaces would be known and parking availability could be derived. However, this is highly unlikely as people would have to know that such an app exists in the first place and furthermore there is no instant positive effect of having such an application installed for a single user. Such an app might also have a negative influence on the battery life of a user's smartphone even if all of the above applications are designed for a low power consumption.

Facing this problem in 2014 Nandugudi et al. proposed PocketParker [13], a smartphone application which should not only detect parking and unparking events but also estimate the amount of so called "hidden drivers" - drivers that do not have the application installed. PocketParker uses a simple model to detect parking and unparking events. It uses the Google Play Services activity recognition library to detect changes from driving to walking and vice versa and thus derives parking and unparking events. After this information is available, PocketParker tries to estimate the amount of drivers having the application installed, so that all detected events could be scaled to the right proportion and taken into the overall parking estimation process. The PocketParker app not only identifies the parking space location, but also the user's end target after

parking his vehicle. That way they want to identify parking spaces which would be nearer to the end destination of the user, but probably are occupied since the user parked further away and did not take the closest parking space. Using this approach PocketParker derives parking space counts at dedicated parking lots. The overall parking space count of these parking lots is known before. To evaluate its performance Nandugudi et al. performed an experiment at their university's parking lot. 105 users generated 10.827 events over 45 days, therefore an average of 241 events per day. They used cameras to identify the ground truth and then evaluated their results against it. With an estimated driver fraction (drivers which had the app installed) of about 20%, they achieved a parking space count accuracy of about 94% for the university's parking lot.

Advantages and disadvantages of using an event based approach

There are obviously a number of advantages using smartphones to detect parking availabilities. Nowadays almost everybody has at least one smartphone which he/she carries around all day. Therefore, smartphones are the optimal sensing devices for detecting a user's activities throughout the day. Also parking and unparking activities can be detected and parking availabilities can be derived. The whole detection process would cause no additional costs for hardware installations and maintenance for city authorities. Furthermore, the detection process could all be done without the user even noticing, because the detection process could theoretically run fully as a background process.

However, a big concern of mobile applications is power usage. Activity detection requires certain sensors to run periodically which prevents the phone from being idle and therefore uses more battery. Even if all of the discussed approaches intentionally use high power consuming sensors like GPS sensors or microphones only in rare cases, the immanent sensing of low power consuming sensors will also cause decreased battery hours. Another disadvantage is that a lot of users would have to install the app for accurate parking availability estimations. Even if Nandugudi et al. [13] showed that the so called "hidden drivers" can be estimated to a good degree, it will still be challenging to get a user base of even 20% as they proposed in their paper. The first obstacle would be to let users even know that there exists such an app, so that they can install it. The following problem would be the cold start problem. After the app is released, usually only a few people will have it installed and they will probably not see an advantage of such an app because the availability estimates will not be accurate if there is only a really small user base.

2.1.5 Drive-by park sensing using cameras

In contrast to approaches with stationary sensors which were discussed in sections 2.1.1 and 2.1.2, drive-by sensing approaches use sensors which are mounted on vehicles driving through the city to sense vacant and occupied parking spaces. Using such mobile sensors has the goal of achieving a high enough accuracy at a fraction of the costs of stationary solutions. In this section the usage of cameras mounted on driving vehicles to detect parking spaces is discussed.

There exist a lot of different approaches in detecting parking spaces from a picture captured by a camera which is mounted on a vehicle. Most of them were designed for the purpose of park assistance, so that parking is being made easier for drivers or that the vehicle can even park itself. However, almost all of such approaches could also be applied for camera-based drive by sensing.

The first discussed approach is to use parking markings. Xu et al. [18] implemented a color vision based parking space detection algorithm which tries to identify parking markings on the street. They tried to use neural networks to learn to distinguish parking marking pixels from others and they used stereo vision to identify obstacles on parking spaces. A similar approach has been done by Jung et al. [8] who implemented a hough transform on the captured images to detect lines (parking markings) on the images. However, the accuracy of color vision based approaches, as just discussed can vary highly in different lightning conditions, conclusions or shadows. Furthermore, this approach obviously only works with parking spaces where markings are present.

Another approach is to use two or more images to create depth maps to estimate vacant parking spaces. Kaempchen et al. [9] developed a stereo camera based approach which estimates depth maps based on two images of calibrated cameras placed on the vehicle next to each other. By finding features in both images and correlating them, the disparity and thus the depth of points can be estimated and vehicles as well as free spaces can be detected using the 3D information. However, a stereo camera solution has the drawbacks of relying on sensitively calibrated cameras which should have a reasonable resolution to be able to match the features in both images.

Depth maps can also be generated using a single camera with motion stereo-based 3D reconstruction. In 2010 Suhr et al. [17] developed an approach using a fish eye camera mounted on a vehicle, which is facing backwards. It takes multiple images in a sequence and searches for point correspondences in adjacent images to reconstruct 3D structures and to estimate depth maps. This eliminates the need of a second camera while still

providing similar results. Suhr et al. reported a 90% success rate of their approach in detecting vacant spaces.

All of the above discussed camera based drive-by approaches were not designed to detect parking spaces while driving through the city and would have to get evaluated to know if they are suited for the task. However, there exists a recent work of Grassi et al. [5] about drive-by sensing to detect parking space availability at a road segment accuracy level. They developed a smartphone application, called "ParkMaster". The user has to mount his smartphone behind the wind shield of his vehicle so that the smartphone camera is able to capture images of the area in front of the vehicle. ParkMaster tries to detect parking cars using the camera's video stream and counts them. In combination with parking space counts at a road segments level they then estimate the number of vacant parking spaces. For the detection of the parking cars they use the video stream of the smartphone camera as input for a Viola-Jones feature-based cascade classifier, which is a machine learning technique to detect complex objects within images. Several positive and negative examples in different lightning conditions were used to train the classifier in an offline process. While driving the smartphone searches for the learned features in the captured images and the classifier reports the bounding boxes of detected vehicles. However, as the vehicle is driving, it might detect the same vehicle in multiple subsequent frames of the captured video sequence. ParkMaster handles this problem by estimating the GPS coordinates of detected parked vehicles and then compares their position to identify multiple detections of the same vehicle. For the calculation of the sensed vehicle's GPS coordinates, ParkMaster uses the coordinates of the bounding box in the image in combination with self calibrated parameters of the smartphone camera.

Grassi et al. evaluated their prototype with real world experiments in Paris, Los Angeles and a small village in Italy [1]. During their test drives, they recorded a dataset containing 5.896 parking cars and 2.280 vacant parking spaces. They manually assigned ground truth values, optimized the variables of their algorithm and tested their approach on the dataset which produced an overall accuracy of close to 90%.

Advantages and disadvantages of camera based drive-by park sensing

As only the last approach discussed in this section, is designed for parking space availability estimation, this paragraph will be focused only on ParkMaster's advantages and disadvantages. The main arguments, Grassi et al. give to promote their approach is the high enough accuracy and that high hardware costs can be avoided, because of the use

---

[1] Sant' Angelo in Vado

of smartphones of end users as processing devices. If smartphones are used, costs would be lower, however, as already discussed in section 2.1.4, smartphone apps would need to have a high enough distribution for enough drivers to sense available parking space. Furthermore, drivers would have to mount their phone each time they drive through the city and this obviously also costs a lot of battery hours. Therefore, it is not sure if a smartphone based system would give a good enough coverage. Of course dedicated cameras could be used as part of a hardware installation in a car, but this would of course introduce new costs, while still keeping costs lower than with stationary sensor deployments. Further limitations of drive-by park sensing in general will be discussed section 2.1.7.

2.1.6 Drive-by park sensing using distance sensors

Another approach of drive-by sensing is to use a distance sensor instead of a camera as discussed in the previous section. Mathur et al. [12] developed such a prototype, called ParkNet, in 2010. It consists of an ultrasonic range finder which measures the distance to the nearest obstacle on the right side of the road and a GPS receiver which records the corresponding locations. Furthermore, a camera was used for obtaining ground truth information through manual tagging of the captured images. Distance measurements are taken continuously at an interval of about 50 milliseconds (20 measurements per second). Mathur et al. deployed their system on a standard car, which collected test data in drives during daily commuting throughout 2 months in selected areas in Highland Park, New Jersey. Altogether, about 500 miles of street parking scenes were collected during their test period.

ParkNet's detection algorithm is based on thresholding. It tries to detect so called "dips", which are parts of the sensor signal caused by parking cars or other objects. Sensor readings are separated from each other by overflow measures which show that at that point the distance is higher than the range of the ultrasonic sensor. In a second step, these dips are compared to two thresholds for the distance and the length of the dip. These thresholds have been derived by a subset of the data and produced an overall error rate of 12.4%. ParkNet assumes that detailed parking space maps are available. They distinguish between a slotted and unslotted model for their parking space detection. In the slotted model, they assign detected parking cars to single parking spaces using the GPS position. However, GPS positions are not always as accurate as they need to be to correctly identify a parking space. Therefore, Mathur et al. developed an environmental fingerprinting algorithm which corrected GPS positions using multiple test runs on the same street. They identified static objects which were there all the time and then clustered their positions. They used the first cluster center

as true position and corrected the surrounding GPS measurements. In the unsotted model, they estimate the number of free parking spaces by checking for all free spaces if the length is high enough for a car to park in between two objects.

Mathur et al. used their acquired dataset to evaluate their algorithm. On the unslotted model, they tried to estimate the number of vacant parking spaces in a specific street. Their algorithm turned out to have an overall accuracy of about 95%. For the slotted model they reached an accuracy of 91% using their environmental fingerprinting approach to correctly assign detected parked cars to parking spaces.

To show that their system is more cost effective than stationary sensor deployments, Mathur et al. also presented a mobility study which was done in the city of San Francisco, California. They tried to estimate the interval in which a sensing vehicle would visit a street. The GPS locations of 536 taxi cabs which were collected during one month were used and the analysis of these data showed that while in the outer San Franciso areas visiting intervals can be as high as hundreds of minutes, in the down town areas 80% of the streets are visited every 10 minutes. Furthermore, they estimate the costs of running their system to be more than 12 times cheaper than with the use of stationary sensors at each parking space.

Advantages and disadvantages using distance sensor based drive-by park sensing

As already discussed ParkNet is much more cost effective than stationary sensor deployments while providing a sufficient accuracy in down town areas with enough sensing vehicles. In contrast to camera based drive-by sensing a higher accuracy can be achieved, due to the fact that distance information is available and therefore free spaces can be identified. Camera based drive-by sensing can only detect cars and then estimate the vacant parking spaces using an overall count of parking spaces per road segment. Furthermore, ParkNet is also able to assign parked cars to single parking spaces. This enables better parking analytics and may be also used by parking enforcement to spot illegally parked cars. Limitations of drive-by park sensing in general will be discussed in the next section.

2.1.7 Limitations of current drive-by park sensing approaches

In this section limitations which apply to both camera based- and distance sensor based drive-by park sensing will be discussed. The first limitation is based on the cars

(a) Parallel parking cars   (b) Perpendicular parking cars   (c) Angular parking car
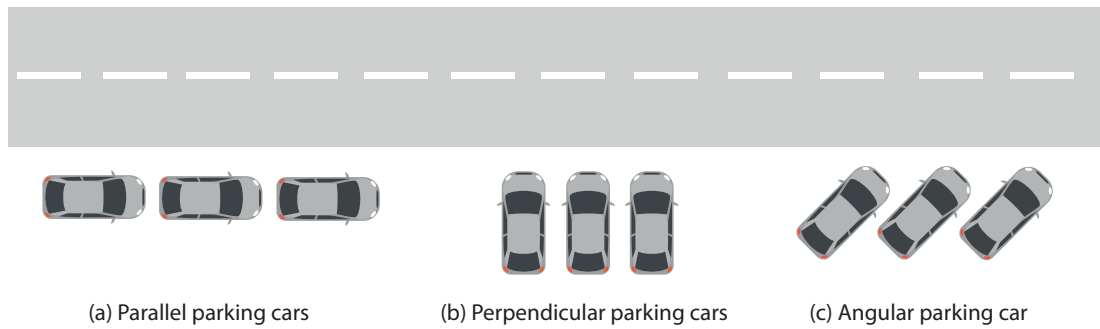
Figure 2.1: Different types of parking cars

which are detectable. Drive-by sensing approaches currently only try to detect parallel parking cars. Even if in most cities parallel parking spaces are the most prominent type of parking spaces, obviously there exist a lot of other parking spaces (the different types of parking spaces are shown in figure 2.1). Camera-based drive-by sensing would have to also learn to detect images of such parking cars and distance sensor based drive-by sensing would have to be much more variable in terms of length of detected cars as well as the patterns of the detected distance to the obstacle. For instance, in the case of angular parking spaces, the discussed approach of distance sensor based drive-by sensing would not work as they only apply thresholding and therefore it is likely that such parking spaces would not be detected correctly.

Another limitation of drive-by sensing are multi-lane roads. All of the discussed drive-by sensing approaches only work when the sensing vehicle drives in the right-most lane. In modern cities, there are often multiple lanes and GPS is too inaccurate to estimate the lane, where a vehicle is going. Therefore, to work properly lane detection would have to performed to be able to work on multi lane roads. However, it would be preferable to have a algorithm which also works in other lanes than the right most as then the coverage of detections would be greater. For algorithms to work on non-right lanes, they also have to face several distractions. Of course, the distances to parking cars will be greater if the sensing vehicle is going on a left lane. Additionally, when a sensing vehicle overtakes another car which is going slower, it will record the distances to this vehicle instead of the distance to parked cars. Furthermore, also distractions such as traffic lights or traffic jams have to be taken into account for a real life installation of such prototypes.

	Accuracy	Costs	Applicable for parking lots or parking garages	Applicable for road side parking
Stationary Sensors	very high ●	very high ●	yes ●	yes ●
Stationary Cameras	high ●	high ●	yes ●	yes ●
Counting Vehicles	very high ●	low ●	yes ●	no ●
Event-based Sensing	high ●	low ●	yes ●	yes ●
Drive-by Sensing (Camera)	high ●	medium ●	no ●	yes ●
Drive-by Sensing (Distance sensor)	high ●	medium ●	no ●	yes ●

Table 2.1: Comparison of several park sensing approaches

2.1.8 Comparison of the existing park sensing approaches

Table 2.1 shows a comparison of the park sensing approaches which were discussed in this section. All of the above discussed approaches show high accuracy (>= 90%) but not all of them are suited for all kinds of park sensing. For instance, the drive-by approaches are not able to detect parking availabilities in dedicated parking lots or parking garages whereas vehicle counting cannot detect the parking situation of road side parking spaces and thus a city's overall parking situation.

The costs of the different approaches also varies highly. Approaches with stationary sensors usually have the highest accuracy, but also bring the highest costs. Event-based sensing operates on very low costs because it usually uses smartphones as sensing and processing devices. However, to reach a high accuracy a lot of people have to sense parking events. Furthermore, such apps will also drain the user's battery faster than usual.

Drive-by park sensing approaches are in the medium cost range because the vehicles which are sensing have to be equipped with the required sensors, but in contrast to stationary sensing, much fewer sensors can reach a sufficient degree of accuracy. However, there are certain limitation of the current approaches as described in the previous section. In this thesis, a prototype using distance sensor based drive-by park sensing will be developed and evaluated, as the ratio from cost to accuracy seems to be the best of the investigated approaches. To improve accuracy in real life scenarios, a prototype

which uses machine learning will be used to identify all types of parking cars (seen in figure 2.1) as well as different distractions such as overtaking while driving on multi lane roads.

2.2 Acquiring parking space maps

Parking space maps are of great importance when sensing a city's parking availability. Knowing where parking is allowed and how parking is possible is crucial information for park sensing approaches. Several of the in section 2.1 discussed approaches use some kind of parking space map. Coarsely grained parking maps in the form of parking space counts for road segments/parking lots are necessary for several approaches, like the counting all in- and outing vehicles approach discussed in section 2.1.3. Furthermore, such information is also necessary for event-based park sensing (section 2.1.4) as well as drive-by sensing (sections 2.1.5 and 2.1.6). Of course parking space maps with a higher granularity can be beneficial in most cases and are sometimes even necessary. For instance, the exact position and orientation of parking spaces can increase the performance of drive-by sensing.

Most previously discussed papers assume that parking space maps are already available, for example at city authorities. However, even if a city authority has a parking space map of their road side parking spaces, that does not mean that it is ready for processing or containing the necessary information. That is why Coric et al. [3] developed an algorithm to derive parking space maps from the data stream of an ultrasonic range finder. They used the same equipment as ParkNet [12] to record GPS and distance information to the right side of the road while driving through streets in their test zones. Using this test bed, they acquired two datasets in Highland Park, New Jersey and in Brooklyn, New York City. After obtaining the data, they try to predict areas where parking is legal and such where it is not. Their hypothesis is that spaces which are almost never filled will be most likely illegal spots, whereas spaces which are often occupied will be likely legal parking spaces.

Using sensor readings from several runs they try to aggregate this data to derive their results. They first assign the sensor readings to one meter cells and then try to classify these cells as legal or illegal parking spots. Using their algorithm, called "Weighted occupancy rate thresholding approach", they not only use occupancy rates per cell, but also take into account, that runs with almost only occupied cells provide better information than runs with a lot of vacant spaces, as usually only the illegal spaces stay vacant, when the parking availability is low in a certain street. They compute a

occupancy rate per cell and weight it using the overall occupancy of the street. Then they apply thresholding to the occupancy rates to classify whether parking in the range of a cell is legal or not. Coric et al. evaluated their approach using the obtained dataset and found that their acquired parking space maps have a false negative rate of 5.21% and a false positive rate of 5.89%. Thus, using the specified setup, parking space maps can be derived at a good accuracy and may be used for the approaches stated in section 2.1.

To-Do: short paragraph about deriving parking space map in this thesis?

2.3 Machine learning

Chapter 3

Prototype Implementation

As already discussed in chapter 1, sensing a city's parking availability and thus making the parking situation more transparent would be highly beneficial for reducing traffic congestion and greenhouse gas emissions as driver's looking for parking spaces in urban areas could navigate directly to vacant parking spaces close to their destinations. In this thesis a prototype of drive-by sensing using an optical distance sensor was designed, implemented and evaluated.

This chapter will describe the setup and implementation of the prototype. In particular, section 3.1 will discuss the overall envisioned system, the prototype car and the required sensors with their capabilities. In section 3.2 the experimental setup will be discussed as well as the acquiring of the test dataset through test drives in the city of Linz, Austria. Finally, section 3.3 will describe the preprocessing of the data, definition of the features and a short description of all used machine learning algorithms with their configurations. **To-Do:** update :)

3.1 System design

This section will describe the envisioned system for the urban parking availability detection system. The proposed system is a mobile sensing system where sensing vehicles try to detect parking cars while driving through the city. The sensing vehicles are equipped with several sensors. Most importantly an optical distance sensor measures the distance to the nearest obstacle on the right side of the road continuously while the vehicle is driving through the city. Furthermore, a GPS receiver determines the corresponding positions.
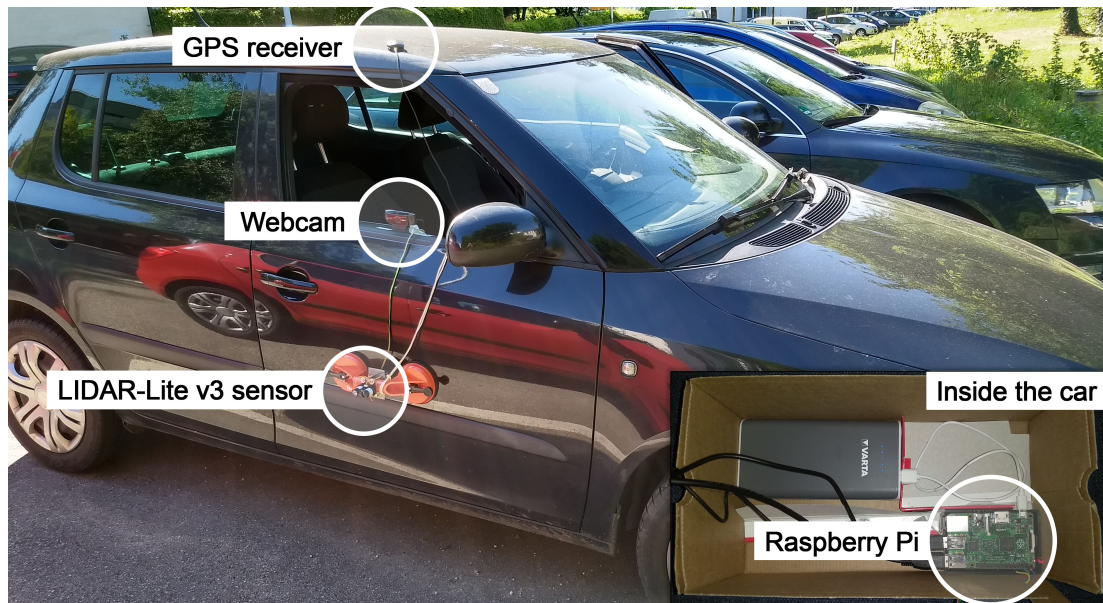
Figure 3.1: Prototype of the sensing car, which is composed of a LIDAR Lite v3 optical distance sensor, a GPS receiver, a camera for ground truth collection and a Raspberry Pi as processing device

	Sampling Frequency	Costs per Sensor
Lidar Lite v3	200 measurements/s	€169,00
Navilock USB GPS receiver	1 measurement/s	€74,90

Table 3.1: The used sensors

- System description (type of sensors, requirements); overall envisioned system of cars detecting free spaces and communicating the information

3.1.1 Test bed description

- Testbed description (concrete HW and technical capabilities)

Figure 3.1 shows the sensing car and its mounted sensors.

3.2 Experiment description and data collection

- Description of experiments (scenarios we are interested in: parking car, etc.) - Data set derived (raw data, size, etc.) - Map data and camera ground truth data

3.3 Data processing

- Preprocessing, feature definition ... - Traditional ML Algorithms (incl. a short description of each algorithm and configuration options) - Deep Learning

3.4 Experimental results

- Results of the different ML approaches, important features, difficult cases, etc. - Comparison of traditional ML and Deep Learning (cf. our demo)

Chapter 4

Results and Discussion

Chapter 5

Conclusions and Future Work

Bibliography

[1] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo. Car parking occupancy detection using smart camera networks and deep learning. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 1212–1217, June 2016.

[2] Katy Blumer, Hala R. Halaseh, Mian Umair Ahsan, Haiwei Dong, and Nikolaos Mavridis. *Cost-Effective Single-Camera Multi-Car Parking Monitoring and Vacancy Detection towards Real-World Parking Statistics and Real-Time Reporting*, pages 506–515. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[3] V. Coric and M. Gruteser. Crowdsensing maps of on-street parking spaces. In *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pages 115–122, May 2013.

[4] Daniele Di Mauro, Sebastiano Battiato, Giuseppe Patanè, Marco Leotta, Daniele Maio, and Giovanni M. Farinella. *Learning Approaches for Parking Lots Classification*, pages 410–418. Springer International Publishing, Cham, 2016.

[5] Giulio Grassi, Kyle Jamieson, Paramvir Bahl, and Giovanni Pau. Parkmaster: An in-vehicle, edge-based video analytics service for detecting open parking spaces in urban environments. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, SEC '17, pages 16:1–16:14, New York, NY, USA, 2017. ACM.

[6] Han Hao, Yong Geng, and Joseph Sarkis. Carbon footprint of global passenger cars: Scenarios through 2050. *Energy*, 101(Supplement C):121 – 131, 2016.

[7] Texas Transportation Institute. Urban mobility report. 2007.

[8] Ho Gi Jung, Dong Suk Kim, Pal Joo Yoon, and Jaihie Kim. Parking slot markings recognition for automatic parking assist system. In *2006 IEEE Intelligent Vehicles Symposium*, pages 106–113, 2006.

[9] N. Kaempchen, U. Franke, and R. Ott. Stereo vision based pose estimation of parking lots using 3d vehicle models. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 2, pages 459–464 vol.2, June 2002.

[10] J. Liu, M. Mohandes, and M. Deriche. A multi-classifier image based vacant parking detection system. In *2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*, pages 933–936, Dec 2013.

[11] Shuo Ma, Ouri Wolfson, and Bo Xu. Updetector: Sensing parking/unparking activities using smartphones. In *Proceedings of the 7th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, IWCTS '14, pages 76–85, New York, NY, USA, 2014. ACM.

[12] Suhas Mathur, Tong Jin, Nikhil Kasturirangan, Janani Chandrasekaran, Wenzhi Xue, Marco Gruteser, and Wade Trappe. Parknet: Drive-by sensing of road-side parking statistics. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 123–136, New York, NY, USA, 2010. ACM.

[13] Anandatirtha Nandugudi, Taeyeon Ki, Carl Nuessle, and Geoffrey Challen. Pocketparker: Pocketsourcing parking lot availability. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 963–973, New York, NY, USA, 2014. ACM.

[14] Sarfraz Nawaz, Christos Efstratiou, and Cecilia Mascolo. Parksense: A smartphone based sensing system for on-street parking. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, MobiCom '13, pages 75–86, New York, NY, USA, 2013. ACM.

[15] Vehicle Sense. http://www.vehiclesense.com (accessed on november 9th). 2017.

[16] SFMTA. http://sfpark.org (accessed on november 9th). 2017.

[17] Jae Kyu Suhr, Ho Gi Jung, Kwanghyuk Bae, and Jaihie Kim. Automatic free parking space detection by using motion stereo-based 3d reconstruction. *Machine Vision and Applications*, 21(2):163–176, Feb 2010.

[18] Jin Xu, Guang Chen, and Ming Xie. Vision-guided automatic parking for smart car. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511)*, pages 725–730, 2000.

[19] N. R. N. Zadeh and J. C. D. Cruz. Smart urban parking detection system. In *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 370–373, Nov 2016.