Daniel Burgstahler, Fabian Knapp, Sebastian Zöller, Tobias Rückelt and Ralf Steinmetz: Where is That Car Parked? A Wireless Sensor Network-Based Approach to Detect Car Positions. In: Proceedings of the 9th IEEE LCN International Workshop on Practical Issues in Building Sensor Network Applications (IEEE SenseApp 2014), p. 1-9, September 2014.

Where is That Car Parked? A Wireless Sensor Network-Based Approach to Detect Car Positions

Daniel Burgstahler, Fabian Knapp, Sebastian Zöller, Tobias Rückelt and Ralf Steinmetz
Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Darmstadt, Germany
Email: {firstName.lastName}@KOM.tu-darmstadt.de

***Abstract*—The global trend of increased urbanization makes space rare in city environments in general and for parking in particular. In addition, cars become bigger and often use more than one parking space. As a result neighboring parking spaces can be affected by a parked car. So, a basically free parking space might be too narrow for an arriving car depending on the arriving car's size. Therefore, means to detect car positions on parking spaces in a fine granular way are required to detect such situations and avoid inefficient parking space searches. Wireless sensor networks provide the possibility to sense the exact occupation of a parking space and potential influences on neighboring parking spaces. However, current solutions focus only on the detection if a parking space is occupied or not. In our work, we present a sensor deployment and a machine learning-based approach able to provide the mentioned more fine-granular detection level. We have conducted an extensive real-world evaluation of our solution, in particular considering different characteristics of today's car bodies. In our tests, our approach achieved an accuracy of more than 98%.**

***Index Terms*—wireless sensor networks, magnetic field sensor, parking sensor, parking position**

I. Introduction

The continuously increasing urbanization all over the world makes space a scarce good in today's city environments [1]. Simultaneously, the amount of registered cars is still extensively growing and car sizes are still becoming bigger (cf., e.g., [2] [3]). Thus, parking space has become rare and the available space for parking must be known in detail and efficiently used.

Only with a detailed knowledge and efficient usage of parking space, a sufficient degree of utilization of the relatively expensive space for parking lot operators can be achieved and long search times for drivers looking for an appropriate parking space can be avoided. The necessary detailed knowledge and efficient usage requires a fine-grained detection of car positions on parking spaces, because a detection solely on the occupancy of a parking space is not enough, as parked cars might influence adjacent parking spaces due to their size or the way the car is parked. This leads to situations in which a basically free parking space might not be usable for an arriving car anymore.

With their sensing capabilities and extendability in addition to their communication and local computing and storage capabilities, wireless sensor nodes (*motes*) and their fusion to wireless sensor networks (*WSNs*) provide the opportunity to realize such a fine-grained sensing of car positions. Using appropriate sensors and locally fusing measurements received from other motes, a mote can decide on the position of a parked car and its potential influence on adjacent parking spaces. Afterwards, the mote can transmit this information for example to a central intelligent parking space control system.

Current approaches usually focus on simple decisions whether a parking space is occupied or not and do not conduct any measurements on car sizes or specific parking positions, which might influence adjacent parking spaces. Additionally, most approaches rely on a threshold-based detection of the presence of a car using magnetic field-based sensors. Such pure threshold-based car detection approaches might become problematic when car bodies are not mainly made of steel.

In consequence, we present a sensor setup for parking space monitoring, which allows a detailed determination of a parked car's position as well as influences on neighboring parking spaces and simultaneously incorporates the varying characteristics of different car bodies. For this purpose, a sensor deployment is required, which provides the possibility to gather diverse data on a sufficiently detailed level. In addition, an approach is needed for the analysis of the gathered data and the decision how a parking space is occupied and how an occupying car influences adjacent parking spaces. Our approach described in the work at hand is based on a WSN constituted of motes equipped with magnetic-field based sensors and an analysis method employing machine learning techniques. The development of our approach has been conducted iteratively on the basis of different field tests. We used as well real-world tests to finally evaluate our approach and the achievable position detection granularity against the background of varying car types. Thus, the major contributions of this paper are:

- A sensor setup enabling data gathering for fine-grained determination of vehicle positions within parking spaces.
- A data fusion and machine learning-based approach for determining car positions on a parking space and effects on adjacent parking spaces.
- An extensive evaluation with real-world tests.

The remainder of this paper is structured as follows: We provide an overview of related work in Section II. In Section III, we sketch our considered application scenario. The development of our sensor setup for vehicle position detection in parking lots is presented in Section IV. Section V describes our machine learning-based approach for determining a vehicle's position on a parking space and potential effects on adjacent spaces. The evaluation setup and results are presented in Section VI. We summarize our findings and provide an outlook on future work in Section VII.

II. Related Work

The detection of cars is not only relevant in parking lots, but as well for example at intersections with traffic lights. Thus, several base technologies already exist which facilitate car detection. Common technologies comprise image-based sensors, magnetic sensors, and inductive loops.

Image-based sensors offer the possibility to provide still images or videos in real time for example from highway parts susceptible to traffic jams or tunnels. However, such data provided by cameras is influenced by weather conditions and limited by the camera properties, e.g., with regard to dynamic range (cf., e.g., [4]).

Employing magnetic sensors for car detection exploit that cars are usually made of a huge part of ferrous material and therefore affect magnetic fields. Thus, magnetic field changes are measured by magnetic sensors to determine whether a car has passed by or not. In this context, anisotropic magnetoresistive sensors are for example employed. These sensors exploit the terrestrial magnetic field and take its strength as set point. Detecting any changes to it and comparing them to thresholds, a decision whether a car is present or not is taken (cf., e.g., [5], [6]).

Inductive loops are a special category of magnetic sensors. Opposed to the anisotropic magnetoresistive sensors, inductive loops are not depending on the terrestrial magnetic field, but create a magnetic field themselves and measure any disturbances of this magnetic field. Using thresholds for the measured disturbances, inductive loops decide whether a car has been sensed or not. Inductive loops constitute by far the most common technology for car detection, in particular for traffic lights signal timing. However, installation and maintenance costs are high and the required wiring is complicated and error prone, sometimes even not possible. Furthermore, inductive loops are tailored to decide whether a car is present or not and thus do not support the exact detection of a car's position (cf., e.g., [7], [8]).

In consequence, possibilities to employ WSN technology as an alternative to inductive loops for car detection in parking lots have been researched by several authors already. Most often anisotropic magnetoresistive sensors are used in the different approaches.

In [9], acoustic sensors have been additionally used. In order to detect cars, the authors proposed a threshold-based approach for the anisotropic magnetoresistence sensor. For the acoustic sensor, the authors developed two approaches based on the detection of temporal acoustic energy concentration. However, the authors found that the acoustic-based approach was rather computing intensive and still not as accurate as their approach employing the anisotropic magnetoresistence sensor.

Problems with acoustic sensors have as well been reported by [10]. Additionally, the authors tested visual light, infrared, temperature, ultrasonic, and magnetometer sensors. Besides the magnetometer sensors, the authors found only ultrasonic sensors usable for car detection. However, the authors experienced as well problems using ultrasonic sensors, as they were not sufficiently able to distinguish between different objects, like people or cars. Thus, the authors proposed a checkpoint-based hybrid approach using both ultrasonic and magnetometer sensors to detect cars entering and leaving a parking lot.

Temperature, light, and acoustic sensors have been employed in [11]. The authors deem light and acoustic sensors as suitable for car detection in a parking lot. However, the authors focus on the development of an encompassing car park management system providing auto-tolling, utilization reports, etc. Thus, they evaluated the mentioned sensor technologies only in a toy car setting without noise and other environmental influences.

Based on anisotropic magnetoresistive sensors and the approach provided in [9], the authors in [12], devise a threshold-based algorithm. With Matlab simulations, the authors could show that suitable car detection is possible even facing noisy measurements. Another threshold-based approach using anisotropic magnetoresistive sensors is proposed in [13]. The authors conducted real-world tests and could detect the occupancy of a parking lot with an accuracy of over 99%.

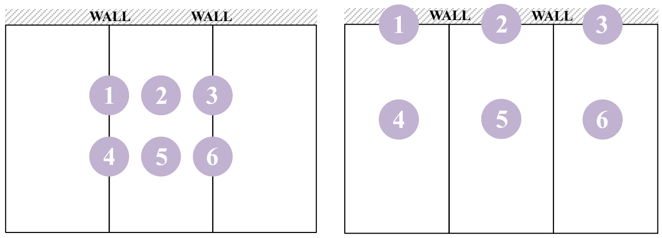
Magnetic field-based sensors constitute the technological basis in most approaches for car detection and is employed in our approach, as well. However, the described approaches specifically tailored for parking lot monitoring focus on simple yes or no decisions, only indicating whether a parking space is occupied or not. Yet, skewed parking and different car sizes affect not only the occupied parking space, but adjacent parking spaces, as well. Thus, our approach is tailored to not only detect whether a parking space is occupied or not, but in particular to determine a car's parking position on the parking space to identify potential influences of a parked car on the vacancy of adjoining parking spaces.

III. Application Scenario

Already today some parking garages are equipped with sensors to detect if a single parking space is occupied or free. However, the overall goal of our work was to elaborate if it is possible to detect vehicles on parking sites in a higher granularity with respect to the vehicle position. With such a knowledge one can derive more information than just the occupancy of a parking space. Also the interference of neighboring parking spaces with respect to an overlap or additional free space can be detected.

The motivation of our objective was initiated by a characteristic situation every motorist has already witnessed several times. On the search for a parking space one notices a potentially free parking space, but on a closer distance one has to recognize that the parking space does not provide sufficient space for the own car. Such situations can be caused by askew parked cars that occupy two parking spaces, big cars on neighboring parking spaces or simply by cars on neighboring parking spaces that park very close to the boundary of the parking space.

Modern parking sites are able to count and display the free parking spaces by the use of an entrance control. However, in situations like the previously described scenario, it is possible that some free parking spaces are displayed, but in reality no single parking space is free or provides sufficient space. Smart



(a) Position 1 (b) Position 2

Figure 1: Overview of the three considered parking spaces with two different sensor position configurations.

parking spaces equipped with sensors have the potential to provide the necessary information to prevent such situations. Based on this information a car driver can be guided directly to a suitable parking space.

To reduce the complexity of external influences, we consider all parking spaces oriented vertically to the driving lane. Our special focus is on a single parking space and both direct neighboring parking spaces. The arrangement of these three parking spaces is depicted in Figure 1. The three rectangles depict single parking spaces with a wall on the top side and the driving lane on the bottom side. The numbered circles thereby depict two different sensor arrangements. The specific sensor deployment is described in more detail in Section VI.

We deployed sensor nodes in our department underground parking to create a testbed. The used sensors are explained in detail in the following Section IV. During our tests we moved different cars into and out of the intermediate parking space and recorded continuously the sensor values of our deployed sensor nodes. All test runs were recorded by video to have an evaluation base for later data analyses. For this purpose a timestamp in the video recording was necessary to have the possibility to match the gathered sensor data with the recorded video. Finally, we implemented a live monitoring functionality that was necessary to verify if everything in the testbed is working correctly.

IV. Sensor Development and Setup

In a first step, we define requirements for our real-world deployment, including hardware components and used sensors. As already mentioned in Section III, we used an underground parking for our deployment. Since it was a public parking, no long term impairments or changes were allowed. This results in the requirement of flexibility in terms of installation and deinstallation.

A basic requirement regarding the sensor platform was to restrict the used sensor to one single sensor type. We decided to use a magnetic field sensor since this sensor type seemed to be the most promising for our application scenario. The used sensor had to be energy-efficient and small to be suitable for a WSN deployment. For the sensor node platform we decided to use TelosB motes. We wanted to have the possibility to gather sensor data from several positions simultaneously. The gathered data was immediately stored in a non-volatile database. Every data record was annotated with a timestamp and an identifier of the respective sensor. The data transfer from the sensor to the recording database had to be reliable for the initial data gathering. Thus, we used a wired connection for data collection within our testbed.

Due to the small overall size and availability we decided to use the HMC5883L 3-axis magnetic field sensor from Honeywell [14]. We used a version directly installed on an extension board as depicted on the left side on Figure 2. This sensor can directly be connected to an I^{2}C bus. However, it has a fixed predefined I^{2}C slave address that excludes the use of multiple sensors on one bus. This results in the use of one sensor per TelosB mote, but this would also be the preferred setup for a real deployment.

To store the sensed data, we decided to use an Android smartphone since this is a very small mobile device, provides USB host mode support, an SQLite database can be used, and sufficient storage is available. For the data transfer from the mote to the storage unit the TelosB supports a wireless connection or the use of USB. Since a reliable connection is of high importance for our setup, we preferred a wired USB connection. The USB connector of the TelosB is provided by a chip from FTDI that also provides an Android driver that allows to directly connect TelosB motes to an Android device [15]. This FTDI driver is basically a wrapper around the Android USB Host SDK that supports a serial port to USB conversion as well as defined control and configuration messages, e.g., baud rate setup. Furthermore, by the use of a USB-hub multiple motes can be connected simultaneously to the Android device. In our test deployment we have connected up to six TelosB motes to one Android device.

The HMC5883L sensor has a fixed 7-bit I^{2}C address and acts as slave. The I^{2}C packet shift register of the HMC5883L is multiplexed to 13 different registers that contain configuration and measurement data [14]. The digital waveform of the I^{2}C bus signal for setting up the configuration register A of the HMC5883L is depicted in Figure 3. This multiplexer is controlled by an internal address pointer (AP). The internal sampling rate of the HMC5883L is configurable. We set it to the maximum of 75 Hz and configured the sensor to output the average value of the last eight measurements to get more

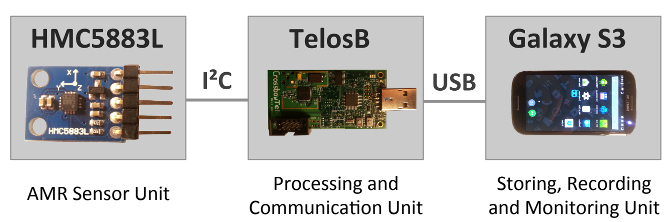


Figure 2: Hardware Overview: The HMC5883L magnetic field sensor is connected to the TelosB mote via the I^2C bus. The TelosB mote is connected to the Samsung Galaxy S3 Android phone via a USB cable.

robust values, cleaned from noise. The HMC5883L can be configured to a continuous measurement mode or to measure only on request. The latter one is more energy efficient, but we used the continuous sensing mode, because our focus was to gather as much data as possible for later analysis and we had no strict energy limitations within our testbed. For a later deployment the measure on request mode would be sufficient. Furthermore, it is important to adjust the output level of the HMC5883L according to the sensed environment. For this purpose the HMC5883L provides a configurable internal measurement amplifier that provides an output range of $[-2048; 2047]$. As a result this output value describes a relative value of the terrestrial magnetic field. To prevent understeering or oversteering, it was essential to find a balanced gain value that also provides a sufficient quantization resolution. For this purpose we empirically determined during several tests a gain value of $390 LSb/G$ (gain configuration bits: $GN2 = 1$, $GN1 = 0$, and $GN0 = 1$) as good choice for our environment.

We connected the HMC5883L sensor to the TelosB mote with wires via the I^{2}C bus. Following the HMC5883L datasheet, a standard request over the I^{2}C bus initiated by the TelosB mote contains a minimum of two bytes. The first byte is the address of the HMC5883L sensor and a flag if it is a send or receive request. In case of a send request the second byte contains the address location the AP should point to. In case of a receive request the second byte contains the amount of address locations that should be read out, starting from the current AP position. This should work since the AP is incremented automatically after every readout to reduce the traffic on the data bus. However, this configuration is not compatible with the official I^{2}C bus specification since there the master is listening after sending a read request and therefore is not capable of sending an additional byte for the AP. Due to this distinction it seemed to be not possible to use the hardware implemented I^{2}C interface of the TelosB mote. We further investigated the behavior of the HMC5883L and figured out that it also works to read out the next bytes without previously sending the amount of expected bytes. As mentioned above, then the AP is incremented automatically, because the TelosB does not stop the communication and the HMC5883L sends the content of the next register. Consequently, we identified that the behavior is compliant to the I^{2}C specification and thus the datasheet of the HMC5883L [14] is misleading.

In order to write to different registers on the HMC5883L, the TelosB has to start the communication over the I^{2}C bus by addressing the HMC5883L with a write request followed by the address of the selected register and the payload. However, before a measurement can be started, the sensor has to be calibrated to zero due to production tolerance and temperature changes. Empirically, we determined the necessary offset as the mean value of the first 40 samples. This offset is subtracted from all following measurements.

In order to be able to place and adjust the HMC5883L sensor along with the TelosB mote, we have mounted it on a wooden block of 20 cm length as depicted in Figure 4. We have mounted the TelosB on the opposite side of the wooden block to minimize errors on the measurement of the magnetic field. The attachment on the wooden block is done by double faced adhesive tape and elastic bands.

So far, we established the connection between the TelosB mote and the HMC5883L magnetic field sensor. The next step was to implement a connection between the TelosB mote and the Android device. The used Android device was a Samsung Galaxy S3. For this connection, we have developed a software component that acts as bridge and correspondingly implements the message protocol to handle the communication. Our protocol basically consists of a header with a fixed length of four bytes and a payload of variable length. The header defines the type and the length of the payload. A sensor value received by the TelosB mote is converted and forwarded to the USB host. A control message querying the ID of the mote that is send from the USB host to the TelosB is handled and answered directly by the TelosB. Furthermore, the protocol has a generic structure which allows to easily include other sensors in the future without changing existing application components.

For the implementation on the TelosB motes, we have used TinyOS and built it onto the available I^{2}C and UART components. It is responsible for the transformation and the processing of messages between the HMC5883L sensor and the Galaxy S3 Android device. A buffering mechanism was necessary to get the application non-blocking and to ensure that received data is not discarded since the used communication units have a strict timing. We ensured that time consuming or dependent invocations are executed asynchronous.

On the Android side, we have split our application into three decoupled parts: USB communication, storage into the database, and the GUI. As software platform we have used the newest custom ROM Android Cyanogen Mod [16], based on Android 4.4 to fully utilize the functionalities of the smartphone hardware. To store the sensor data on the Android device, we use an SQLite3 database that is natively provided by the Android SDK. Thus, our application can use its own database or even generate new databases if necessary. The values of all connected HMC5883L sensors are stored in the database and are linked with the respective unique mote id of the intermediate TelosB. The GUI of our Android application allows to activate or deactivate the previously mentioned zero offset. If a new data recording is started, a new database file is created that can be freely named. The application provides a start button to initialize the USB connection to all connected TelosB motes. After pressing this button, the application is ready to receive,

Figure 4: TelosB mote setup with the HMC5883L sensor mounted on a wooden block.

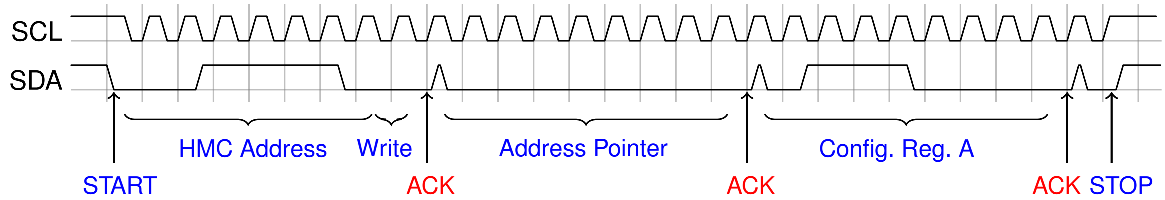


Figure 3: Digital waveform of the I^2C bus signal for setting up the configuration register A of the HMC5883L. Blue denotes the signal of the TelosB and red the acknowledgment of the HMC5883L sensor.

allocate, show, and record sensor data. Received sensor data is displayed twice, in a string formatted manner as well as in graphical manner as a time curve. This allows live observation of the current test run. An overview of the used hardware components is given in Figure 2.

V. Machine Learning based Data Analysis

So far, existing car detection systems only provide information if a parking space is occupied or not, although many systems are based on magnetic field measurements that potentially allow to provide information with a higher granularity. As already mentioned, our goal is to achieve such a higher granularity with respect to the car's position on the respective parking space, e.g., if a car is parked more to the left or right boundary of the parking space. Such a more detailed level of information allows to detect unusable neighboring parking spaces and a more precise information of the effectively available parking spaces of a parking lot.

To achieve this goal we had to decide between a regression based approach or an approach based on classifications. A regressive approach would handle the output attribute as numeric value, e.g., the distance between a parking space margin and the car. Such an approach has several drawbacks: As the width of a car differs between different models, the distance must be measured on several positions to predict an accurate position of a parked car. Furthermore, multiple distances cannot be mapped to one single output attribute that makes this concept even more complex. Therefore, we have decided to develop a classification-based approach. The output value of a classification approach is not a numeric value, but a result mapped to a set of possible outcomes. Our approach is therefore to split a parking space into different sections. Each possible combination of occupied sections constitutes one possible output value.

This also allows an easier aggregation of test data since one single output value per test run is ensured. The complete set of input parameters of our classification system consists of $3 \times n$ numeric input values from the 3-axis magnetic field sensors, where n is equal to the amount of used motes. For the classification, we combined the measurements from multiple sensors on different positions and used all three values from the 3-axis magnetic field sensors. This leads to a large amount of correlated input data, with which a simple threshold based approach cannot cope. The gathered data was then investigated with the use of machine learning (ML) algorithms. In order to reduce complexity, we assumed only passenger cars to be detected. In the next step, we segmented the parking space in order to be able to pursue our classification approach. The parking space is segmented into several segments parallel lengthways to the parking space, whereas on classification level each segment can be detected to be occupied or free. Additionally, we restricted the possible width of the cars since according to [3] the width of nearly 300 popular cars only ranges from approximately 1.47m to 2.0m.

A segmentation into m segments leads to a set of 2^m possible output values. Thus, we used the information of the range of the typical width of a car and simplified the segmentation into five segments per parking space. We also decided to select a smaller size for the outer segments since we are especially interested in the influence of a parked car on the neighboring parking spaces. An overview of this segmentation is depicted in Figure 5.

To test our concept with different ML algorithms, we have used the Waikato Environment for Knowledge Analysis (WEKA) [17] open-source machine learning tool-suite. In a first step, we only concentrated our investigations on one single parking space. We have used a supervised learning approach where the gathered data is manually annotated with the respective output class. To collect the necessary training data we extended our previously mentioned Android application with the functionality to easily annotate the currently measured data as shown in Figure 7. Our approach also allows to

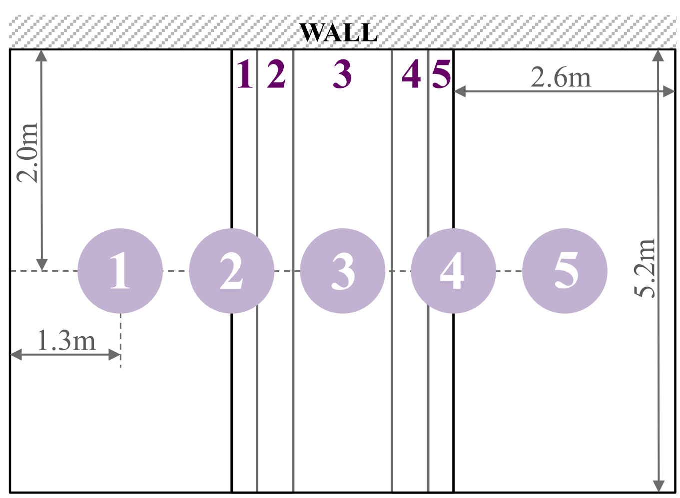


Figure 5: Sensor position configuration and segmentation of a parking space for the second deployment.

Table I: Considered cars.

Car	Peugeot 206	BMW 420d	BMW i3	Audi A2	Saab 9-3 Cabrio	Mercedes C220 CDI
Build Year	1998	2013	2013	2002	2007	2003
Body Material	Steel	Steel	Carbon	Aluminum	Steel	Steel
Width [mm]	1669	1825	1775	1673	1753	1770

take magnetic field measurements from sensors placed on neighboring parking spaces. To prevent a cyclic dependency among instances, our concept does not need any occupancy information of neighboring parking spaces.

Overall, our ML-based concept provides thus a more fine granular detection level with respect to the vehicle position on the parking space than existing parking space sensor solutions. We split a parking space into several segments and determine the combination of occupied segments. For the classification, we combine multiple measurements from different positions. To extract the information, we use all three axis measurements of all sensors combined with machine learning algorithms, in contrast to a simple threshold-based algorithm used in other solutions.

VI. Evaluation

A. Prototypical Deployment

For the deployment of our sensors we have used our department underground garage, as mentioned before. All tests were performed on weekends to exclude external influences, e.g., from other moving cars or pedestrians. However, two basic questions had to be answered in our first tests to build a suitable machine learning model: How many sensors per parking space are required and what are the best sensor positions? Therefore we recorded different sensor position configurations in our first test runs to experimentally find a good combination. Additionally to the gathered data we have recorded a video of our test set for later analysis. By a later comparison of the sensor values and the corresponding test set on the video we were able to decide on the necessary number of sensors and corresponding positions. But previously we had to decide about the orientation of our sensors. Since the HMC5883L measures the magnetic field in orthogonal 3-axis, only two different concepts had to be distinguished: Adjust each sensor to a specific cardinal direction or to a specific direction relative to the parking space. We have tested both concepts and figured out that only the latter one, i.e., positioning relative to the parking space, is suitable to create a general model that is independent of a specific cardinal direction of a parking space.

In our first deployment we have used two different sensor position layouts to gather data as depicted in Figure 1. In configuration 1a all six sensors were deployed on the ground. This configuration is not intended to be a favorite in a real deployment as such a configuration would need four sensors per parking space which is most probably not suitable in terms of costs. However, it gave a detailed first distribution of the terrestrial magnetic field distortion caused by a vehicle. Therefore it helped to identify general relations between measurements and vehicle positions. In configuration 1b sensors $1-3$ were mounted on the wall in a height of 1m on the top side of the parking space. Sensors $4-6$ were deployed on the ground. This configuration was intended to give an insight if measurements taken from an upper position are suitable or even better than measurements taken from the ground. It has to be considered, that sensors $1-3$ were mounted at the wall with an orientation of the x-axis to the top. After analyzing our first measurements we had decided to test a third deployment with a further sensor position layout. All sensors were again deployed on the ground. The sensor position layout is depicted in Figure 5. Our intention was to get information about the difference between overlaid and adjoining sensors. Such a sensor position configuration is also preferred by already existing solutions that are only able to detect if a parking space is occupied or not like the Libelium Waspmote [18].

Indeed a lot of more other sensor configurations could be suitable, but it is possible to derive many of them as subset of the selected ones. The configuration shown in Figure 5 also covers several other configurations, e.g., only one sensor in the middle of each parking space or only one sensor between two parking spaces. In our first test series we were able to do our tests with two different cars: a Peugeot 206 and a BMW 420d. An overview of all used cars to get test data is given in Table I. Each wood block containing a TelosB and a magnetic field sensor was connected via USB cables to a USB hub that was again connected to the Galaxy S3 Android device. We started to construct various different parking situations with each of the

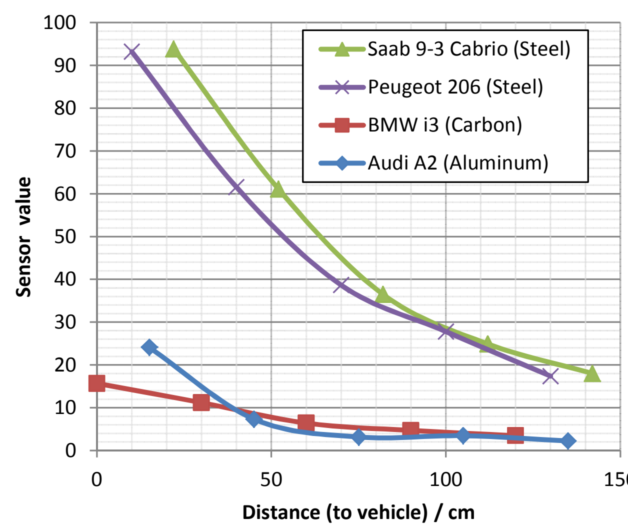


Figure 6: Behavior of the magnetic field strength with an increasing distance for the different considered cars.

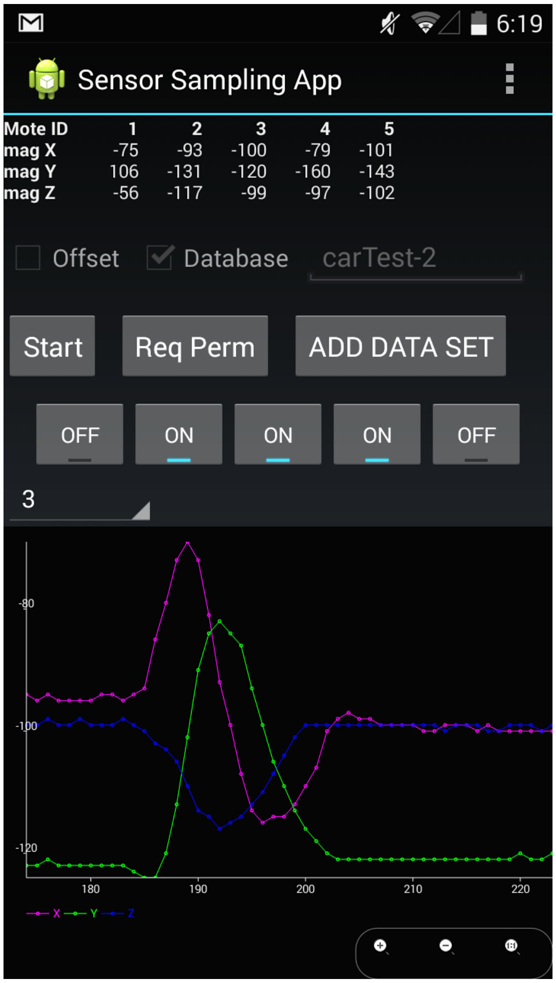


Figure 7: Screenshot of the developed Android application that was used to record the sensor readings.

available cars as well as with both combined. We have repeated these steps analogously for all three mentioned sensor position configuration. The result of our first test series is a database with more than 100,000 magnetic sensor data tuples in addition to about 75 min. of corresponding video recordings. In order to be able to analyze this huge amount of data, we copied it to a PC and accessed the data with Matlab by the use of Kota Yamaguchi's Matlab SQLite3 Driver [19]. In the following we developed a Matlab script that has mainly three tasks: to access the SQLite Database and readout the data, data preprocessing as well as visualization in a time dependent animation. In the first step all database entries of one selected mote are extracted and saved into a matrix in order to achieve processing capabilities of the sensor data in Matlab. The timestamp of the first and last magnetic sensor data entry are defining the interval. The values in between are arranged with a 100ms step width. Each sensor data tuple is a function value for a specific timestamp. In addition to that, the magnitude of each sensor data tuple is also calculated. A resulting continuous graph is created by interpolating the sensor data tuples. In the third step, we visualized and animated the time curve of the measured sensor tuples. The animation consists of a Matlab marker which passes through the resulting functions. Each value results in a step for the marker and therefore a video frame recording animation with a pass through of 10 frames per second results in a time synchronous movie of the occurred sensor values. We repeated these steps 36 times since we had 12 test runs per different sensor position configuration. Finally, we created a movie for each sensor position configuration. To do this we have used the open-source video editor OpenShot [20]. A screenshot of the resulting video for the first sensor position configuration, depicted in Figure 1a, is given in Figure 9. The recorded video can be seen on the upper left. The animation is synchronized with all six sensor data movies that are placed on the lower half. An overview of the sensor position configuration is given in the upper right. This results in movies that give an intuitive overview of the changes in the magnetic field caused by a moving car on a parking space by the animated visualization of several thousands of sensor data tuples. By analyzing the videos we detected significant differences of the magnetic field direction depending on whether a sensor is overlaid or just adjoined by a car. Additionally, we recognized that measurements in the z-direction, from ground to ceiling, can be neglected since these tend to a huge amplitude if a car is nearby. The second setting, depicted in Figure 1b, has shown to be not useful since the amplitudes of sensors mounted to the wall were too small. Based on this knowledge we have chosen the third sensor position configuration, depicted in Figure 5, as the most preferable with respect to information content and

Table II: Occupancy detection accuracy using different machine learning algorithms.

Algorithm Type	Algorithm	Accuracy
Tree	RandomForest	97.18%
Tree	RandomTree	95.23%
Rules	Ridor	80.57%
Rules	JRip	89.29%
Meta + Tree	RotationForest + RandomForest	98.07%
Meta + Tree	RotationForest + RandomTree	98.20%

Figure 8: Measurement setup for distance measurements.

costs. An overview of the relation of the magnetic field strength to the increasing distance to a car is given in Figure 6 for all considered cars. The x-axis shows the distance between the car and the sensor and the y-axis shows the magnitude sensor value that is formed by the magnitude of the single sensor values of all three axes. The figure clearly shows the effect of different vehicle body materials. Figure 8 depicts the measurement setup that was used to get the results shown in the diagram of Figure 6.

B. Machine Learning

We decided to segment a parking space into five sections. Thus, the set of possible outcomes that are our classification classes comprise 32 elements. To simplify our data recordings we have marked these sections with chalk on the ground of the parking space. We have used the information on possible vehicle width values to create an intelligent segmentation scheme. The selected segmentation is shown in Figure 5. The width of segments 1 and 5 is $25cm$, of segments 2 and 4 is $40cm$ and of segment 3 is $130cm$. This leads to a decrease of the possible outcomes as a middle segment (2, 3 and 4) can never be occupied on its own without a neighboring segment as well. Furthermore, the information resolution nearby the outer sections is much better than in the middle of the parking space. Due to these simplifications we were able to reduce the possible results to a set of 21 possible classifications.

We generated more than 50 different car parking situations with up to three cars simultaneously. We recorded each situation three times in order to get more robust results against measurement tolerances. Our developed smartphone application recorded the sensor data and directly formatted the recordings into a WEKA suitable ARFF file. Each sensor shown in Figure 5 provides three values for each measurement, consisting of the magnetic field strength in all cardinal-directions. Each value creates a single feature, which results in 15 different input attributes. Additionally, we modeled the possible outcomes as output attribute that was manually entered via the application GUI and added to the ARFF file. In a next step, we used the GUI-based WEKA Software *Experimenter* to evaluate the detection accuracy of the parking space occupancy by a set of well known machine learning algorithms. The accuracy of each algorithm by using WEKA standard settings and a 10 fold cross-validation is given in Table II. It can be seen that tree-based algorithms are more accurate than rule-based ones. The RandomForest algorithm reaches a quite high accuracy (97.18%) that can be even be increased to 98.20% by the combination with the RandomTree algorithm. These results were above our expectations. In particular, because our test data set also includes an aluminum and a carbon car that have significantly lower influences on the magnetic field.

VII. Conclusions and Outlook

The information about available parking spaces, especially in cities, gets an increasing attention. The amount of cars and also the size of cars has grown, while at the same time the amount of parking space in cities is stagnating. This causes the need for real-time information of available parking spaces. Several big cities have already deployed sensors that are able to detect if single parking spaces are occupied. However, until now none of these systems is able to detect if a parked car also affects a neighboring parking space. To achieve this, the detection of the position of a parked car on a parking space is necessary. The combination of magnetic field sensors, WSN technology, and machine learning exposed as promising technology to determine the position of parked cars on a parking space with a high accuracy. In our work, we have introduced our sensing approach and the used hardware setup in detail. We have conducted an extensive real-world evaluation of our solution, in particular considering different materials of today available car bodies. In our tests we achieved an accuracy of more than 98%.

In our future work, we aim to extend our analysis and try to reduce the amount of necessary sensors. First tests have shown that a reduction should be possible while preserving the detection accuracy. We additionally strive to analyze the effects of deploying the sensors embedded in the pavement. To evaluate in more depth, we will further evaluate our approach with more car categories and in other environments.

Acknowledgements

Parts of the research leading to these results have received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreements no. 285220 and no. 318201.

References

[1] United Nations, "Global share of people living in cities from 1950 to 2050," Apr 2011. [Online]. Available: http://www.statista.com/statistics/274520

[2] Ministry of Transport of China, "Amount of passengers cars in China from 2001 to 2012 (in millions)," Sept 2013. [Online]. Available: http://www.statista.com/statistics/278423

[3] ADAC e. V., "Zu enge Fahrstreifen in Autobahn-Baustellen," 2013. [Online]. Available: http://www.adac.de/_mmm/pdf/fi_enge_baustellen_versetztesFahren_mitAnlage_sp_0513_95200.pdf

[4] B. Hosticka, "Safety and Security: Smart Cameras for Intelligent Buildings," in *Sensors in Intelligent Buildings*, O. Gassmann and H. Meixner, Eds. Wiley, 2002, vol. 2, pp. 409–426.

[5] M. J. Caruso, T. Bratland, C. H. Smith, and R. Schneider, "A New Perspective on Magnetic Field Sensing," *Sensors*, vol. 15, pp. 34–47, 1998.

[6] M. J. Caruso and L. Withanawasam, "Vehicle Detection and Compass Applications using AMR Magnetic Sensors," in *Proceedings of Sensors Expo 1999*, vol. 477, 1999.

[7] C. Sun, "An Investigation in the Use of Inductive Loop Signatures for Vehicle Classification," *California PATH Research Report*, 2000.

[8] E. Schnieder, *Verkehrsleittechnik*. Springer, 2007.

[9] J. Ding, S. Cheung, C. Tan, and P. Varaiya, "Vehicle detection by sensor network nodes," *California PATH Research Report*, 2004.

[10] S. Lee, D. Yoon, and A. Ghosh, "Intelligent parking lot application using wireless sensor networks," in *Proceedings of International Symposium on Collaborative Technologies and Systems*, 2008, pp. 48–57.

[11] V. W. S. Tang, Y. Zheng, and J. Cao, "An Intelligent Car Park Management System based on Wireless Sensor Networks," in *Proceedings of 1st International Symposium on Pervasive Computing and Applications*, 2006, pp. 65–70.

[12] H. Hui, J. Backens, and M. Song, "A High-Performance Vehicle Detection Algorithm for Wireless Sensor Parking Systems," in *Proceedings of 5th International Conference on Mobile Ad-hoc and Sensor Networks*, 2009, pp. 327–333.

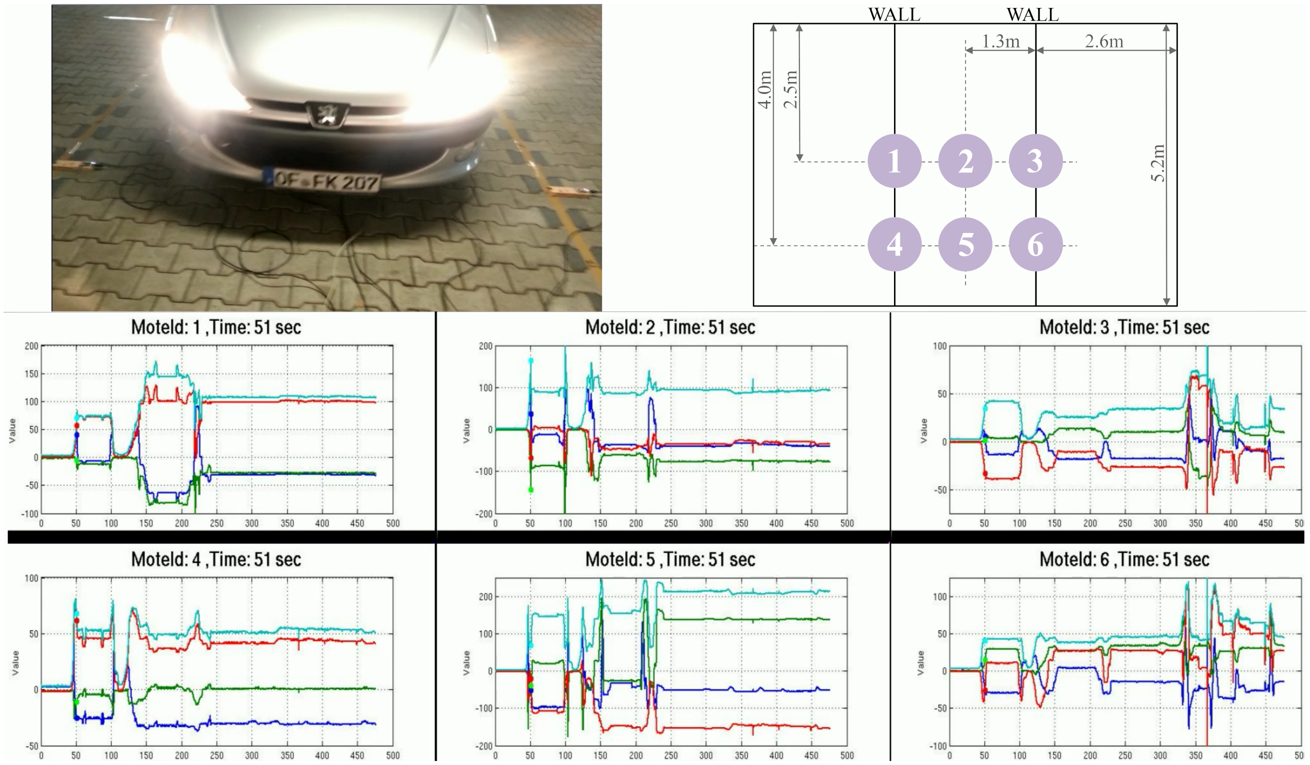


Figure 9: Developed movie combining time synchronized sensor value animations and video data.

[13] J. Gu, Z. Zhang, F. Yu, and Q. Liu, "Design and implementation of a street parking system using wireless sensor networks," in *Proceedings of 10th IEEE International Conference on Industrial Informatics*, 2012, pp. 1212–1217.

[14] Honeywell, "3-Axis Digital Compass IC HMC5883L," 2013. [Online]. Available: http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf

[15] Future Technology Devices International Ltd., "D2XX Direct Drivers," 2012. [Online]. Available: http://www.ftdichip.com/Drivers/D2XX.htm

[16] CyanogenMod, LLC, "CyanogenMod," 2014. [Online]. Available: http://www.cyanogenmod.org/

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[18] Libelium Comuicaciones Distribuidas S.L., "Smart Parking – Technical Guide," 2013. [Online]. Available: http://www.libelium.com/downloads/documentation/smart-parking-sensor-board.pdf

[19] K. Yamaguchi, "Matlab SQLite3 Driver," 2013. [Online]. Available: http://www.cs.sunysb.edu/~kyamagu/software/matlab-sqlite3-driver/

[20] OpenShot Studios, LLC, "OpenShot Video Editor," 2013. [Online]. Available: http://www.openshot.org

All online references in this paper were last accessed and validated in April 2014.