Summer 2016

Magnopark, Smart Parking Detection Based on Cellphone Magnetic Sensor

Maryam Arab
Old Dominion University, marab002@odu.edu

MAGNOPARK, SMART PARKING DETECTION BASED ON CELLPHONE MAGNETIC SENSOR

by

Maryam Arab
B.A. January 2005, Azad University of Tehran, Iran

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY

August 2016

Approved by:

Tamer Nadeem (Director)

Michele Weigle (Member)

Stephan Olariu (Member)

ABSTRACT

MAGNOPARK, SMART PARKING DETECTION BASED ON CELLPHONE MAGNETIC SENSOR

Maryam Arab
Old Dominion University, 2016
Director: Dr. Tamer Nadeem

In heavily congested urban areas, rapid growth of population is becoming more and more of an issue. Affected cities quickly demand solutions to areas such as: quality of life, waste management, public transportation, and accessibility to main resources. However, since the number of impacted areas of population growth is endless, we focus on public parking. As noted in [3], drivers spend a large portion of their travel time locating vacant parking spots. For this reason, we present Magnopark, a crowd sourced approach to identifying unoccupied spots accessible to the general public, which are typically free. Magnopark is a smart phone based sensing solution that detects empty parking spots using internal sensors of cellphones. While a pedestrian is walking on the sidewalk, we exploit magnetometer changes near metal objects in identifying where cars are located. The amplitude and rate of change shift dramatically when approaching or passing cars that are parked beside the street, giving us a great platform towards solving the defined problem. With empirical evaluation, we show that not only is our solution a notable step towards economical parking management, but it's also significantly more energy efficient and as accurate as traditional sensor based parking solutions.

ACKNOWLEDGMENTS

I would like to express my special thanks and appreciation to Dr.Nadeem for his great continuous support and guidance in completing this thesis. I would also thank Dr.Weigle and Dr.Olariu for consenting to be on the thesis advisory committee. I would like to thank my Lab colleagues for their help and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 MOBILE APPLICATION DEVELOPMENT

Mobile sensors and using them in developing mobile application have been a hot topic due to its high usage in every single person life. Nowadays, everybody uses a cellphone and its applications for daily purposes including navigation, weather situation check, exercise, gaming, recordings, to name but a few. Most of mobile built-in sensors are used to measure motion, direction, vibration, orientation, and various environmental conditions. These sensors are capable of gathering raw data in three-dimensional movement and positioning, with a high rate of accuracy. Also, they are capable of gathering raw data with different frequency rates in accordance with the user demand. The accuracy of these sensors concur with the development of a lot of motion-dependent applications. For example, a game might use a device gravity sensor to detect user gestures and motion. A secure message communication application might use the accelerometer to transfer messages or secret pass code via vibration.

1.2 ANDROID SENSORS

Android platform supports 3 categories of sensors:

- Motion Sensors: These sensors measure acceleration and rotation movements along 3 axes. This category includes accelerometer, gyroscope, and gravity sensors.
- Environmental Sensors: This sensors category includes barometer, photometer, and thermometer, which measure various environmental features, like ambient temperature, humidity, and illumination.
- Position Sensors: Orientation sensors and Magnetometers are placed in this category. This category contains sensors used to measure the physical position of a mobile device.

The subject matter of this thesis is to use mobile internal sensors to develop an application to help drivers to detect the curbs' parking spots while looking for parking spaces.

1.3 SEEKING PARKING SPOT IN BIG CITIES

In big cities, parking space is both an expensive and a hard to find resource. On a daily basis, a large portion of the vehicles on the road in urban environments constitute those seeking a parking spot [4]. While the impact is sporadic in nature, at times heavily influenced by the geographic location or the contextual side of its environment, it is a clear issue. According to [3], finding a place to park can take as much as 15 minutes on average in major metropolitan areas. The cause for extraneous search is primarily developed due to two reasons. First, drivers tend to search for spots by preference, where free curb side parking closest to a particular destination is of ideal value. If all spots are occupied, the issue escalates in which driver tendency leads them towards waiting or actively seeking alternative locations, such as garages or pay stations. This is the stage that amplifies the situation of the originating problem, as the inability to know where else to park promotes misuse of driver time, increases traffic congestion and creates health issues due to the emissions released by vehicles [3].

Since this is certainly not a new issue, and is only increasing levels of inconvenience, dense urban areas are beginning to invest heavily towards implementing potential solutions. Some of those include Fastprk [3] and SENSIT [2], which are both sensor based systems for identifying when parking spots are occupied. While they both require physical equipment to fully function, they integrate with public payment and notification systems to help streamline the parking process. Both of these solutions aim to identify vacant spots, guide drivers towards potential locations, increase driver satisfaction and the overall city management. Fastpark claims a 35% decrease in the time needed to park [3], while SENSIT claims both a 64 % reduction in park violations and a decrease in space occupancy [2].

While this is a good approach for locations which generate revenue, for instance paid parking on popular streets, cities lack similar technology for free curb side parking. Investing into previously mentioned solutions is still an option for well-funded cities. However, for those which have a limited budget, covering all potential streets of interest can quickly grow into financial exhaustion. Take for example a city like Chicago or New York, where the number of crowded streets is potentially endless, demanding a large portion of the city's budget for a complete conversion.

1.4 OBJECTIVE AND PROPOSAL

Due to the abovementioned issues, we introduce a solution that uses the availability of heavy crowds and their smart devices, to gain more result as to where potential parking

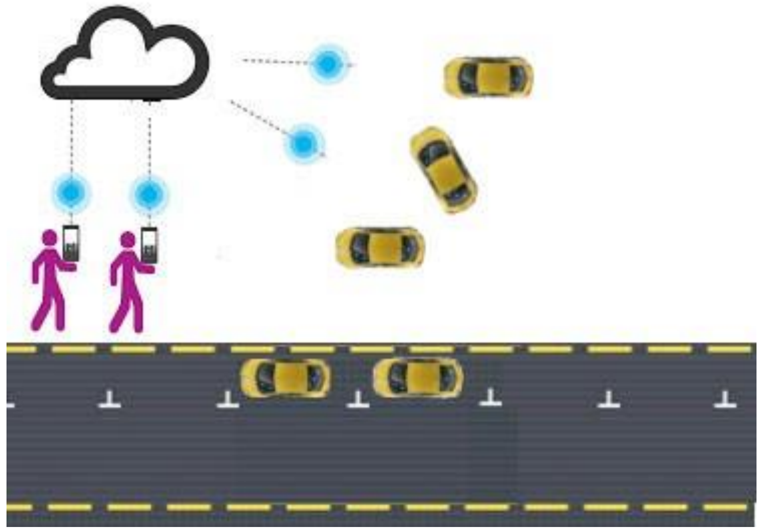


FIG. 1.1: Magnopark System

is possible. By leveraging the raw magnetometer, gyroscope, and accelerometer data, we are able to detect parking spots through the natural movement exerted by the walking pedestrians on the sidewalks beside the streets. Dating back as far as 2013, a very large portion of pedestrians composing the crowds on the sidewalk, possessed at least one smart device in their hand or pocket [14]. It is this statistic that fuels our application, in which we depend on crowds or even a steady rate of pedestrians, telling others around them where unoccupied parking sport are, without making a single bit of noise. In other words, we use the walking pedestrians' cellphone sensors to classify the sidewalk parking spots as occupied and vacant. The more pedestrians walking on the sidewalk, the more accurate our application works. As the years and technological advances both increase, we predict that the number of smart devices will only increase, allowing our solution to become much more precise and useful.

The biggest contribution of our study can be summarized as follows:

- Implementation of Magnopark; a high accuracy parking spot localization system using internal smart phone sensors
- Evaluation and test of Magnopark in different situations and places
- Test of Magnopark for different users with different walking habits and speed

- Development of an algorithm to detect the users’ stride, speed, and direction change
- Building a classification model based on the features extracted from the cellphone sensors
- Pushing the classified data to the cloud for the drivers’ use

CHAPTER 2

BACKGROUND AND RELATED WORKS

2.1 BACKGROUND

Magnetometers are very sensitive to soft and hard iron. This sensitivity is caused by distortion in the earth's magnetic field. Magnetometers sense the change in the earth's magnetic field that is caused by a metallic object. The reason for this distortion is that the magnetic field flows more easily in the ferromagnetic materials than air. This effect causes the earth's magnetic field lines to be bent quite a bit in the presence of any metallic object, including cars. This distortion is caused by the iron used to construct a vehicle [1]. Magnetic fields sensing has expanded vastly as many magnetic sensors are used to detect the strength, direction and distortion of not only the earth's magnetic field, but also the fields generated by electric currents, permanent magnets, and vehicle magnetic field disturbance. Magnetic sensors are able to detect these changes without any physical contact.

Many navigation control systems have an eye to this feature to correct the magnetic deviation caused by hard iron and soft iron in order to reach an accurate tracking for both under water and out of water vehicles. Strong algorithms including Kalman filter are used in these systems to correct the distortion that is caused by any kind of hard or soft iron objects in the earth's magnetic field. In addition to tracking purposes, portable sensor systems are designed and developed to be used beside the roads for vehicle counting and classification and also for speed measurements [15].

Figure 2.1 shows how the earth's magnetic field distorts in the presence of a metallic object. As you can see, the vertical lines represent the earth's magnetic field that are almost parallel, and the presence of a large metal car causes these parallel lines to be bent and distorted.

By observing those characteristics, leveraging the internal magnetometer of smart de- vices allows us to detect the presence or absence of a vehicle. The idea came from the significant changes in the magnitude value of the magnetometer that have been seen while passing beside vehicles. Although many researchers did not count on the magnetometer sensor due to its high variation and inaccuracy, we leverage this feature to localize empty parking spots and notify drivers who are looking for the parking space beside the curbs.

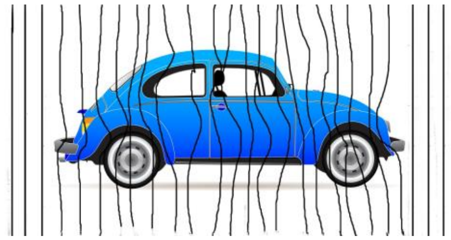


FIG. 2.1: Earth magnetic field distortion in the presence of a vehicle

2.2 RELATED WORKS

2.2.1 RFID BASED SMART PARKING

One of the most popular ways researchers are battling smart parking is through RFID technology, where small instruments are installed in each vehicle to communicate with a base station. Using such an approach, individuals can be identified by their device, and management applications can get a head count as to how many spots are vacant or filled [12]. While such a system decreases wait times and traffic jams, it comes with three main disadvantages:

- Cost: The system must be implemented in all vehicles, including those of the drivers and those maintaining the proposed technology. These implementations and maintenance is a rather costly solution.
- Accuracy: Such a solution can be very error prone in dense areas, as multi-broadcast collisions can prevent several vehicles entering a parking lot simultaneously.
- Security: Security issues can arise as a limited amount of preventative measures are being taken towards ensuring devices do not spoof their unique identifier.

2.2.2 LED LIGHT BASED SMART PARKING

Unlike RFID solutions, there are those leveraging light as a medium for parking identification. By measuring the distance that vehicles cover as they travel throughout a particular area, similar solutions can be implemented. One of those can be seen in [8], where the authors develop a LIDAR system consisting of light sensors tracking movement of all entered vehicles. At the end of a travel cycle, a map is generated for a particular path, and an estimation can be made as to what spots are no longer vacant. A similar approach is taken by [10], which while very accurate, is a rather expensive solution requiring a plethora of equipment to configure.

2.2.3 VANET-BASED APPROACH FOR PARKING SPACE AVAILABILITY

In this approach a network model is proposed for the Parking Spot Locator (PSL) and Parking Lot Notifier (PLN). The network is modeled as vehicle-to-infrastructure (V2I) communication with onboard units (OBU) on vehicles and static Road Side Units (RSU). In this approach the entire parking areas of each city is divided to many overlapping zones. This model assumes that all vehicles have sensors on all sides to sense the presence of an object in a small range. Each RSU maintains the occupancy state of the parking spots in parking lots. In order to do so, when a vehicle is arriving within a specified zone which is farther from the destination with a specific distance, the OBD queries the RSU, giving the driver current GPS location. The RSU responds back with the state of suitable parking lots that are closer to the driver [13].

Another approach in this category is Murat [6]. Murat proposes an algorithm that is dividing the entire area into a Grid Tree Structure with each vehicle maintaining a resource report and aggregate report specifying the capacity and occupancy of the parking lots within the grid.

2.2.4 DATABASE OF AVAILABLE LOCATIONS

This model includes a database which store data that is related to the vehicle parking locations. This data includes available parking locations, and a communication link for communicating with vehicles and other sources. The communication link receives parking location information including information of the available parking locations and then provides vehicles with parking location information. In this approach, the system processes the stored data in the database and provides parking location information to vehicles. [11]

2.2.5 IMAGE PROCESSING BASED SMART PARKING

Other solutions, like those outlined in [5], entail video and image processing, scattered transmitter nodes for information relay, ultrasonic waves and microwaves have been used towards vehicle localization. These kinds of solutions are not only very costly and expensive, but they are also very energy consummating and not accurate.

Alike our project, current research attempts to identify new means of tracking vehicles and preventing side effects of congested parking in crowded cities. However, a line between accuracy and cost is quickly expanding, in which the financial aspect dictates the level of performance. Our research, on the other hand, is distinctively different as Magnopark is both accurate and considerably less costly than current solutions. With Magnopark, we only count on the cost effective internal sensors. The only energy consumptive sensor in our application is GPS, which only gets used when the classified data is going to be pushed to the cloud.

CHAPTER 3

MAGNOPARK ARCHITECTURE

3.1 INTRODUCTION

The main objective in designing the Magnopark algorithm is to leverage the low cost, efficient, accurate, and easy to use mobile internal sensors to help drivers finding a cheaper parking spot in the street comparing to the parking lots, in big crowded cities. In the Urban areas, finding a parking place is very time consuming, costly, and boring work. The main objective in designing the Magnopark is to design and develop a cost effective application for detecting the parking spots in the streets.

There are two parameters that we take care of: the cost of the application, and the accuracy. In most parking seeking applications, most of the calculations are performed on the user device, and also they use GPS for finding the parking spots which is a very expensive and non-accurate approach to be used in such a sensitive subject. In this thesis, the idea is to use a cost effective approach which helps people in urban crowded cities to use the negative point of always crowded streets as a positive issue. As almost all the people are carrying a cellphone in their bag or hand, we design an application that uses the pedestrians' cellphone sensors to detect the magnetometer changes while walking beside the vehicles parked beside the street.

We propose the design, implementation, and evaluation of Magnopark application, which represents a system that combines the interface of the presence of pedestrians who are walking on the sidewalk, and their sensor enabled cellphone, with pushing this information to the cloud server to process and be kept in order to be used by the drivers who are seeking curb side parking spots. We design a classification model, whereby parking spot availability is derived from the classifier which executes in part on the pedestrian cellphones and in part on the backend servers to achieve mapped results to inform the drivers who then send request for the nearest parking spot to their location. The framework allows the application to gather data from all pedestrians walking on the street who have cellphones, and uses a very straight forward computing and coding.

3.2 ARCHITECTURE AND DESIGN

The system uses the mobility of the pedestrians, opportunistically gathering data from their motion sensor and GPS and processing the data to assess the curbside parking spots conditions. Using a simple and straight forward algorithm, we show that we are able to differentiate between empty parking spots and non-empty ones. We also have been able to classify the detected part as an empty parking spot or a small space between two consecutive cars. Via careful selection of training data and sensor features and behaviors, we have been able to build a classifier to not only detect the cars, but it is also able to differentiate between a pole and a car. In other words, the surface length of the metal has been detected very accurately.

Figure 3.1 shows the big picture of the Magnopark architecture. As you can see, Magnopark contains three main components:

- Pedestrian component
- Back-end server component
- Driver component

3.2.1 PEDESTRIAN COMPONENT

In the pedestrian component, after the Magnopark system initiates, cellphone starts collecting sensors row data from accelerometer, gyroscope, magnetometer, and GPS. We need GPS data to get the location of the user in order to map it with the classified data on the cloud server. Accelerometer sensor data in company with gyroscope data are used to detect the status of the user whether walking or standing still. The main data that is used for classification is the magnetometer variation. We use the compass magnetometer sensor to calculate its variation of the magnitude of its raw 3 axis value to detect the cars parked beside the street. We calculate the magnitude of the raw 3 axis vector magnetometer sensor in order to reduce the effect of the cellphone orientation in our calculation. Also wee use gyroscope and accelerometer sensors to calculate the step length of the user. User Step length is used to calculate the walking speed. Since different users have different walking rates, it is important to calculate their walking speed to be able to figure out if the user is walking, or standing still. Also based on the user speed, the length of the cars is measurable. Moreover, we use gyroscope to detect the direction change of the user. This helps us to avoid the system from detection one parking spot twice, in the situation that

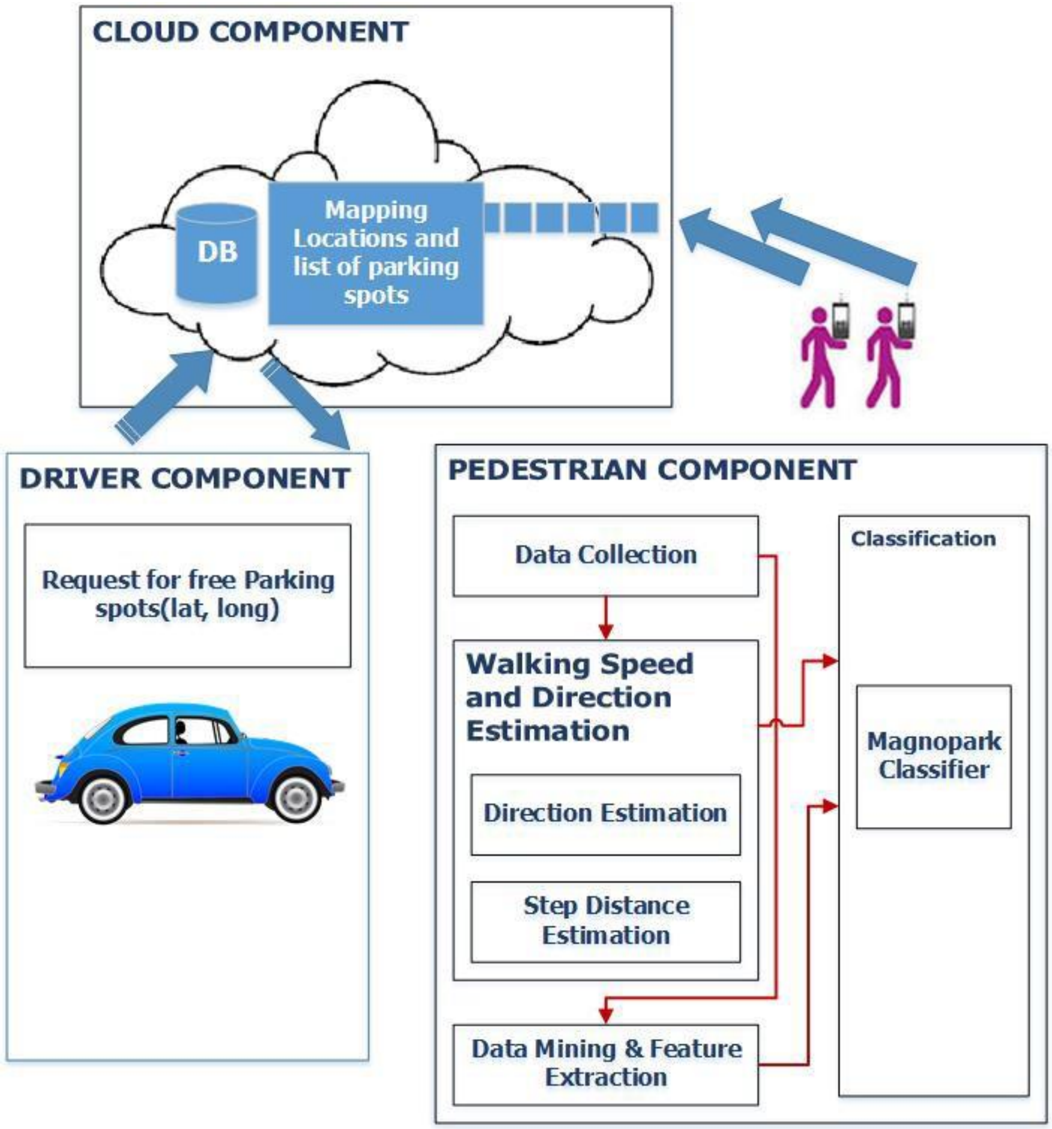


FIG. 3.1: Architecture overview of the different components of Magnopark system.

user turns completely and start walking in the opposite direction. With all these motion sensors, we design an algorithm to detect the cars parked beside the curbs. After specifying the location of the car in the street, we would be able to specify the available parking spots.

3.2.2 CLOUD SERVER COMPONENT

In the cloud component all the pedestrians' classified data in company with their corresponding GPS location will be collected for processing and mapping. Then the mapped result will be stored and get updated periodically on the server.

3.2.3 DRIVER COMPONENT

The Driver component has just one module in which the driver uses the cellphone GPS location to send a request to the server for the nearest parking spots and get respond back with a list of locations that is marked on the driver cellphone local map. Then the user would be able to select one of the location and get direction toward that.

3.3 SUMMARY

In Magnopark, pedestrian smart phone calculates the spots locally and no collaboration with neighboring devices is required. Also pedestrian component does not deal with maps and mapping process. It only requires to transmit the result to the cloud and therefore the user application does not require to share any information and data with the driver's clients directly. This feature ensures to preserve the user security and privacy. In the next chapter, we provide an overview of each component in details.

CHAPTER 4

MAGNOPARK COMPONENTS

Magnopark consists of three main components which will be described in details in the following sections.

4.1 PEDESTRIAN COMPONENT

Figure 3 shows the six main modules in pedestrian component. Figure 4.1 shows the design overview of the Magnopark system on the pedestrian component. The first module is data collection in which the cellphone motion sensors' data and GPS data are collected. The whole collected data in company with the time stamp are used to extract the most prominent features in our classification. In this step, the algorithm uses the accelerometer and gyroscope to not only detect the walker steps and estimate the speed of walking, but it also uses the gyroscope sensor to detect the changes in the direction of walking. Therefore, if the user rotates to a different direction, walks to the same location that already passed, the application will detect it. Also we need to calculate the speed of walking to differentiate between the case in which the user is walking beside a big car, and the case in which the user is walking beside a small car but stops for a couple of seconds beside the car. Each module is described in details in the following:

4.1.1 DATA COLLECTION

We developed an android application to collect mobile internal sensors data. The system collects readings from accelerometer, gyroscope, magnetometer, and GPS. We resort to vector magnitude of 3-axis magnetometer sensor samples, which reads the variation of magnetic field in each of 3 coordinates, while the user is walking in the sidewalk in order to make Our algorithm independent of the orientation of the cellphone. Therefore, we can count on the data that is collected while the cellphone is being kept either in user's hand or pocket or bag. This is a good feature because it reduces the trouble of maintaining specific position or direction in the phone for the user. In other words, there is no need to hold the phone in hand, and in a specific direction to collect data.

The application collects magnetometer data with frequency rate equals to 100 Hz. This rate is high enough to detect the slightest changes in the magnitude value. We also need

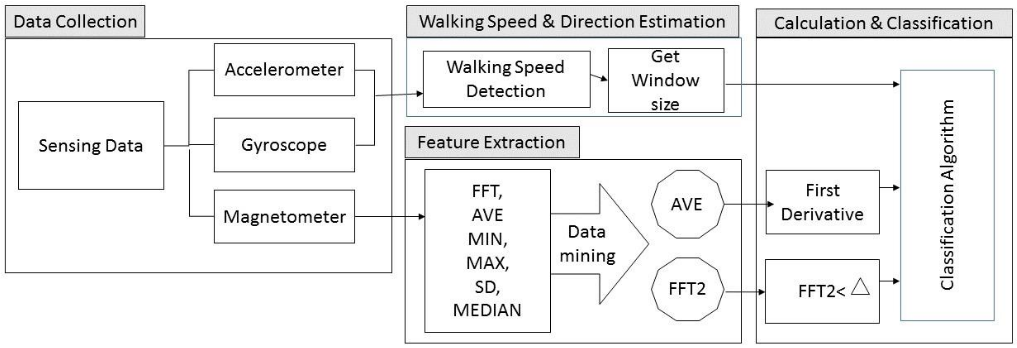


FIG. 4.1: Magnopark system design

gyroscope data to differentiate between 2 status of changing direction of the phone, and approaching a vehicle parked beside the street and accelerometer data is used to calculate the speed of walking. Also, using accelerometer data in company with magnetometer changes, we will be able to calculate the length of the cars parked beside the street. We need to calculate the speed of user walking to be able to decide about the window size that we need to detect a car and distinguishing its length. Also we would be able to tell the number of free spots by calculating the accelerometer rate. Moreover, GPS data is collected to use later for uploading the corresponding free parking spot latitude and longitude on the cloud to be mapped.

4.1.2 WALKING SPEED AND DIRECTION ESTIMATION

This module consists of 2 main sub-modules:

Distance Estimation Module

This module estimates the distance traversed by pedestrian at each step. We used accelerometer and gyroscope sensors of the user cellphone to detect and track user steps. It initiates with user moving and in order to detect the user moving, we utilize the changes in gyroscope sensor. Whenever the gyroscope sensor readings reaches above a certain

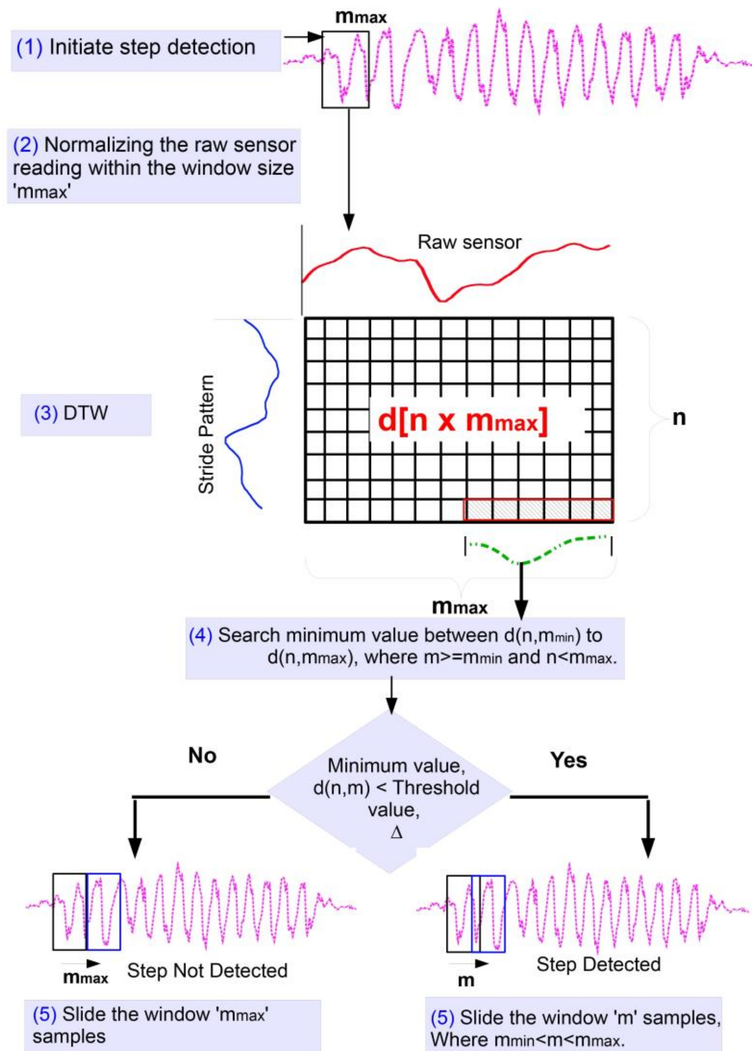


FIG. 4.2: Adaptable step detection module in SpyLoc[16] localization system

threshold (in our case 0.3), we infer the user movement. Detecting the user walking triggers the application to start capturing accelerometer sensor data. Since the user steps length is, to some extent, proportional to his speed [7], we consider 2 parameters m_{max} and m_{min} to represent the maximum and minimum length of a person's step in terms of number of sample:

$$m_{max} = S_{max}/V_{max} * f_a \tag{1}$$

$$m_{min} = S_{min}/V_{min} * f_a \tag{2}$$

S is the length of a person's speed, V is the walking speed, and fa is the frequency of the accelerometer samples in the smart phone. After collecting m_{max} rows of accelerometer samples, we calculate the magnitude of 3-axis accelerometer sample to be sure that it is independent of the orientation of the cellphone. Next, we apply the Finite Impulse Response low pass filter to remove the noise following by normalization. Then, we feed the samples to the Dynamic Time Wrapping (DTW) algorithm. DTW consider a window size of mmax and check if a predefined step pattern with size of n is within this specified window or not. If a step is detected, then we shift the window to the next window sample. Regardless of different step length in different users, the DTW could detect step [16]. The whole overview of the algorithm is shown in figure 4.2.

Direction Estimation

We need to track the direction of each user walking in order to avoid the erroneous duplication parking spot detection, as well as to differentiate between the states in which the user is standing still beside a parking spot, and the status in which the user is walking beside multiple consecutive parking spots. In order to estimate the user direction, we have to align three different coordinate system which is shown in figure 4.3: cellphone coordinate system, users walking coordinate system, and global coordinate system [16]. As the global coordination is fixed, we map the user coordinate and the cellphone coordinate to the global coordinate. After aligning these 3 systems on the globe, then the highest variation of the linear acceleration readings will show the users walking direction. We apply Principal Component Analysis (PCA) [9] analysis to find out the direction of the user face which corresponds to one of its coordinate systems. [16]

4.1.3 DATA MINING AND FEATURE EXTRACTION

After calculating the walking speed, we have to specify a window size based on the

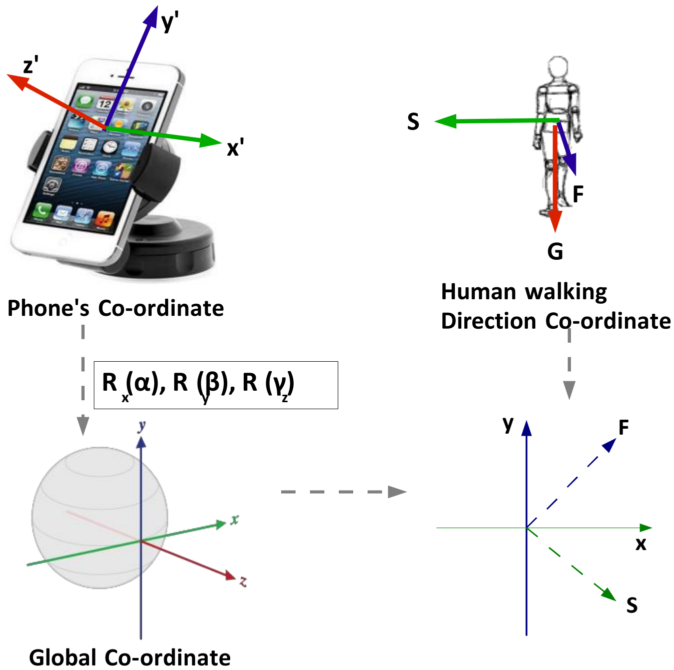


FIG. 4.3: Phone, User's walking and global coordinate systems and relationship between them.

walking rate, to classify the data. The main purpose of the window specification is to be able to get the length of the cars, and also to calculate the number of parking spots available. For instance, with the normal walking rate, the time taken to pass a mid-size car is 3 seconds. As the length of the car is approximately 3-3.5 meters, it means that the normal walking speed is 1 meter per seconds. With this in mind, if the user walks slower, for instance 0.5 meter per second, the time to traverse the length of a mid-size car would be 6 seconds. On the other hand, with the normal walking speed, it takes the same 6 seconds to pass a bus or truck. So the user walking speed is one main feature that should be considered in our classification. The same scenario is applicable for detecting the parking spots; with the low speed walking rate, one parking spot could be detected as two consecutive parking spots. Hence, we need to specify the window size to feed it to the Calculation and Classification part for accurate detection.

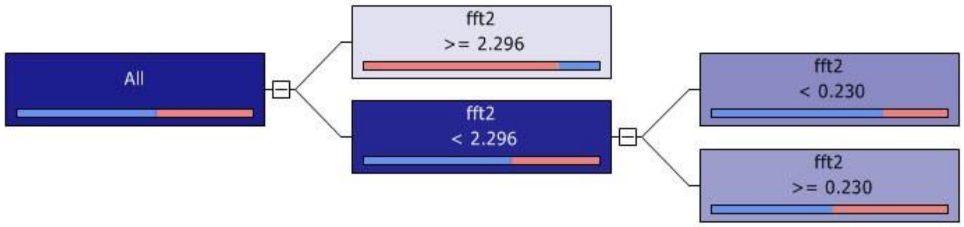


FIG. 4.4: Excel SQL Datamining tool Feature Extraction (Decision Tree Algorithm). The blue part in the first box represents the cars. It shows that around 60% of the cars have the value fft2 less than 2.296

Features

Rather than the sensors data and walking speed calculation, we need to extract the most prominent features in car detection. We calculate the magnitude of 3-axis raw magnetometer data which is collected with the frequency equals to 100 Hz. So, for every 100 rows of data that corresponds to 1 second, we calculate the following five values as the common features:

- **Median:** Median of every 100 samples of data
- **Min:** Min value of every 100 samples of data
- **Max:** Max value of every 100 samples of data
- **Standard Deviation**

$$SD = 1/100 * \sum_{i=1}^{n}(Xi - mean)^2 \quad (3)$$

- **Average**

$$Average = 1/100 * \sum_{i=1}^{n}(Xi)^2$$

In addition to these five values, in order to explore more features, we apply Fast Fourier Transform (FFT) on the average samples of the magnitude of the magnetometer to get the frequency spectrum of the samples in the frequency domain. Since the calculated values for higher frequencies is negligible, we use the first 10 transformed values of magnetometer magnitude in our features list.

- Avg(Mag)
- Min(Mag)
- Max(Mag)
- Mean(Mag
- SD(Mag)
- FF1 .

.

.

- FFT10

We detect 15 values as the general features that is mentioned above. To extract the most prominent features in our detection, we use Decision Tree and Naive Bayes algorithm to detect the features that have the highest effect on our detection. All the 15 calculated features are fed to the Extract Prominent Features under the Feature Extraction component in figure 4.1. In this step, we apply the mentioned algorithms to extract the most prominent features in classifying the curb side spaces with vacant and non-vacant parking spot classes. Weka data mining tool in company with Microsoft Excel SQL Data Mining tool are used for this purpose. These features include:

- Ave(Mag)
- First Derivative(Mag)
- Win Subtraction(FD)
- FFT2

Figure 4.4 shows one of the main features that is extracted. As you can see, more that 60% of the magnetometer samples, which have FFT2 greater than a specific value (2.296), collected while the user walked beside the vehicle. Although it would be a good feature to detect around 60 percent of, the cars correctly, we need to extract more reliable features that bring us more accurate results.

Another prominent feature that is extracted from our decision tree algorithm is the magnetometer average of the magnitude. Applying the mining algorithm, we conclude that sharp changes in the average of magnitude has the most prominent effect in the detection. Therefore, instead of using the magnetometer raw value, we calculate the first derivative of the magnetometer as the main prominent feature:

$$FD = \frac{AVE(magnitude)_n - AVE(magnitude)_{n-1}}{t_n - t_{n-1}} \tag{5}$$

As frequency is equal to 100, then dt = 0.01,

Also, we observe that variation of the first derivative at the beginning and end of each car is significantly larger than the first derivatives of magnetometer value while the user walks beside the parking spots. Because of this behavior of magnetometer, we extract another feature," Win-subtraction", as the second effective feature. This feature is the subtraction of the last and first value of first derivative in the specified window size. In other words, with the normal walking speed, if the window size is equal to 3, then the Win-subtraction feature will be calculated as follow:

$$Win - subtraction_n = FD_{n+3} - FD_n \tag{6}$$

The most effective features that are extracted in the second level in figure 4.1 are fed to the Calculation and Classification part. The combination of the two features - First Derivative of magnitude and Win Subtraction(FD) - leads us to reach an algorithm which with 98 percent accuracy is able to detect the car spaces and parking spots. The classified data alongside the corresponding GPS location is pushed to the cloud server for processing, mapping, and updating. Then the mapped data will be stored on a database on the server to be fetched on the drivers' demands.

4.1.4 CLASSIFICATION

This section includes the training and testing part of classification that we developed to detect whether the data sample corresponding to the car or parking spot. We split the collected data to two sets; training data and testing data. We used 80% of the whole combined data as the training set to build the classification model, while the rest 20% is

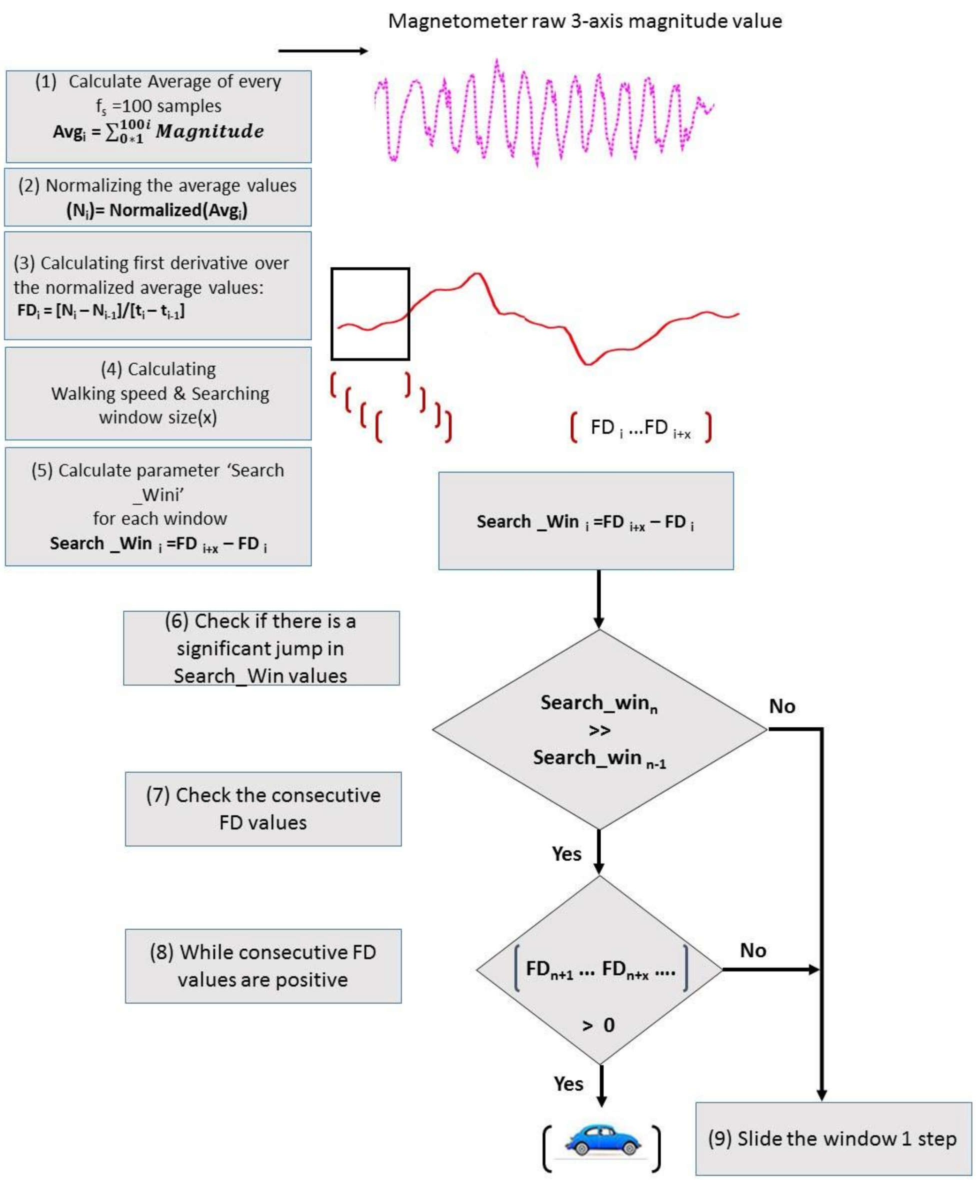


FIG. 4.5: Magnopark classification algorithm.

used as a testing data to evaluate our developed classifier. In training classification model, based on the calculated pedestrian walking speed, we select a window size for our classifier. The default window size which is corresponding to the normal walking speed is equal to three. The feature set F={Ave(magnitude), First derivative of magnitude, Win-subtraction, FFT2} is extracted for each sample in data set. Finally, we use the classification model in the following equations to classify the extracted features F:

$$Classify(c, nc) = \begin{cases} c & : Condition1 \\ nc & : Otherwise \end{cases}$$

in which condition 1 is:

$$Win - sub_{n+1} >> Win - Sub_n \tag{7}$$

AND

$$[FD_n, FD_{n+3}] > 0.1 \tag{8}$$

OR

$$FFT2 > \boldsymbol{fft} - threshold \tag{9}$$

The classification model that we use in Magnopark is shown in figure 4.5. This model basically performs binary classification to classify the samples as C (car) or nc (No-car).

We conduct our experiments in different places with different situations, including on the heart of the campus where there is no car around the user, on the less crowded streets, and also on the shopping store street while the cars are passing in the middle of the street. All these scenarios will be explained in more details in the Experiment chapter. We also asked multiple users with different walking speed and habit to test the Magnopark. Our straight forward classifier is enough to achieve more than 96 percent accuracy in detecting the parking spots.

4.2 CLOUD SERVER COMPONENT

The classified pedestrians' cellphone sensors, samples are pushed to the cloud server in the form of lists of detected parking spots and their corresponding latitude and longitude. The received data from different pedestrians will be queued to the server and periodically gets updated with any changed that received from the pedestrians' cellphone. The most recently updated data will be saved in the form of latitude, longitude and a list of its corresponding free parking spots. The responsibility of connecting different user cellphone data in a manner to achieve the most accurate and recent changes is performed on the cloud.

4.3 DRIVER COMPONENT

Whenever the driver who is seeking parking spots, starts the system on is local cellphone, Magnopark uses the GPS to get his current location. As soon as detecting the location a request will be sent to the cloud server for his most recent location. In case there is any available parking spot near his location, Magnopark will receive them and mark them on the driver local map. Next, the driver is supposed to select among the received location and get directed to the parking space.

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1 INTRODUCTION

In all our experiments we asked users to hold the cellphone in whatever direction they want. The approach used to evaluate the accuracy of this framework and conducting the experiments is using an application for collecting data which not only records all the internal sensors data, but it also had a button to log the times in which the user is beside the head and tail of the cars. For this purpose, we asked the users to push the log time button once they reach the head of each car, and another time when they reach the end of the car. These log times are used as ground truth in marking the car ranges in the whole experiments. We do not use the camera for our ground truth because it is hard to extract the very accurate times that users pass the car.

Another important issue that we considered in our experiments is to differentiate if the users walk beside a car or a metallic object that could be detected as a small car like a metal pole. In order to do so, we use the camera as a helping device to detect the approximate time periods that the users pass the non-vehicular device. In the real application, camera is not used. This is just for the purpose of reaching an algorithm to work for our purpose.

The last but not least important factor that we considered in our evaluation is to evaluate different situation in the parking areas. For Instance, differentiating between the case in which 2 consecutive cars are parked, comparing to a big vehicle like a truck or a bus. The magnetometer behavior in these two cases is not similar.

To evaluate the device independence accuracy of the Magnopark idea, we collect sensors data on a Samsung Galaxy 5, running Android Lollipop.OS, and LG Nexus 4 E960. In order to make sure that the experiments are done accurately and they are consistent, we asked the users to perform each experiments two or three times. We also asked different users to repeat the same experiment in the same place and situation with their different walking habit and speed. In order to evaluate the independence of the Magnopark idea from the location of the experiment, we conduct our experiments in different locations, including the parking lots, private streets, shopping center streets, crowded streets, and also in the heart of the campus where there is no vehicle around.

Magnetometer calibration is another important issue. It is crucial that the magnetometer is aligned and calibrated for not only sensor errors, but also for magnetic distortions. Therefore, we asked users to calibrate the campus sensor in order to remove the noises. For this purpose, they have to rotate the cellphone around its 3 axes.

5.2 EXPERIMENTS

After applying all the prerequisite conditions described above, we conduct 2 sets of experiments:

- Controlled Experiments
  The purpose in this set of experiments is to test if Magnopark is able to differentiate between different scenarios, including differentiating between two consecutive cars and a big truck, differentiating between a small car and any other metallic object like an electricity pole, and detecting the number of consecutive parking spots.
- Real World Experiments
  In this set of experiments, we test Magnopark accuracy in numerical study. We asked users to collect the sensors' data in different places and combine the collected results under a large file. Also, we asked them to log the number of the cars and parking spots via the application. Then we test Magnopark for the collected data and calculate the accuracy of the results.

5.3 CONTROLLED EXPERIMENTS

In the following sections, we evaluate the performance of Magnopark framework under 2 main scenarios:

5.3.1 SCENARIO 1: MAGNETOMETER BEHAVIOR IN NO-CAR SITUA- TION

In order to make sure that the magnetometer variation is due to the existence of the cars, as it is shown in figure 5.1, we set up our first experiments in the middle of the campus, where there is no car around the user. The data set are collected while the users walk straight. Figure 5.2 shows a variation of the magnitude of raw 3 axes magnetometer data. The magnitude in this scenario stays almost stable. Another experiment that we conducted is to walk in the shopping street that there is no parking part in front of the stores, beside the curbs. The result can be seen in fig 5.3. As you can see, although the

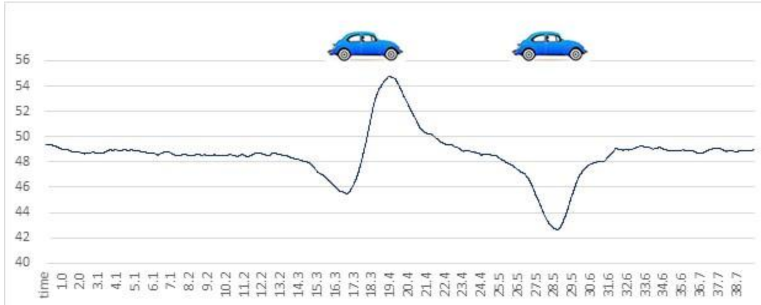


FIG. 5.1: Magnetometer variation while walking beside 2 cars with one parking space in between

base of the magnitude value is not the same, the magnitude stays stable in both scenarios. The higher value of the magnetometer in the shopping area is due to the live stores, and the metal bars that are used in their construction. Based on this experiment we conclude that even though in the shopping street, the cars pass the street, they do not have a significant effect in our detection. The reason for this behavior is that the magnetometer changes is very dependent to the distance of the car with respect to the cellphone; the more is the distance between the passing cars and cellphone, the less sensation of the magnetometer changes.

5.3.2 SCENARIO 2: MAGNOPARK BEHAVIOR IN DIFFERENTIATING BETWEEN CONSECUTIVE CARS AND A BIG CAR

The purpose of this experiment is to check if Magnopark is able to differentiate between multiple consecutive cars and a big vehicle like a bus or a truck. For this purpose, we conduct our test under the following scenarios:

- **Car - Park spot - Car**

  Figure 5.4 shows the cars and their distance in this experiment case. Figure 5.5(a) represents the corresponding behavior of magnetometer, by showing the variation of its first derivative and the calculated win_subtraction value corresponding to the user walking speed, while the user walks beside 2 cars that are parking consecutively with a distance of a parking spot between them. Figure 5.5(b) shows the corresponding ground truth, in which 1’s shows the range that user walks beside cars.

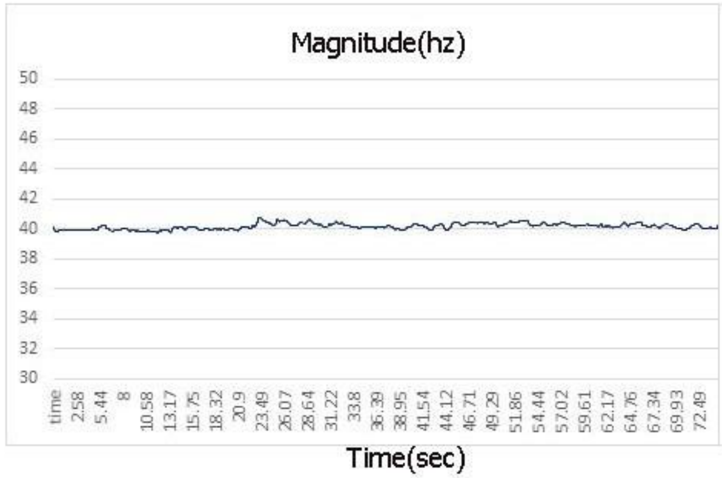


FIG. 5.2: Magnetometer variation while walking in the campus.(No car around)

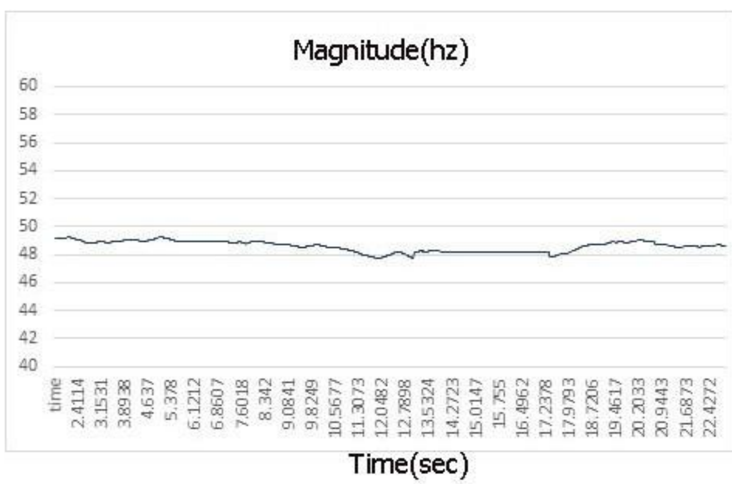


FIG. 5.3: Magnetometer variation while walking in a shopping area with no parking area in front

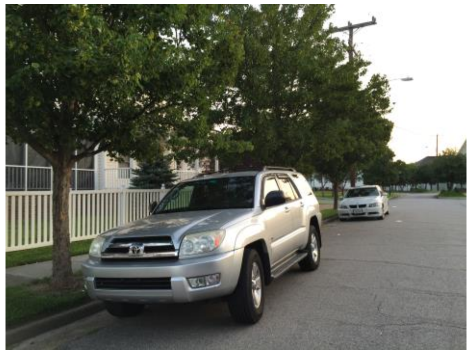
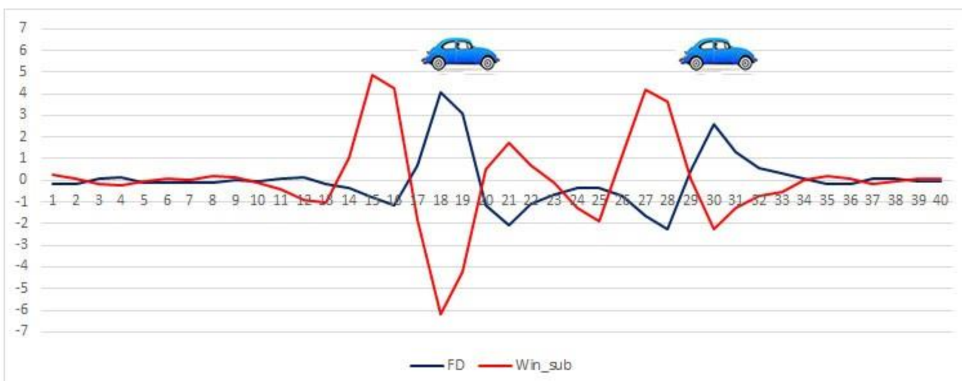
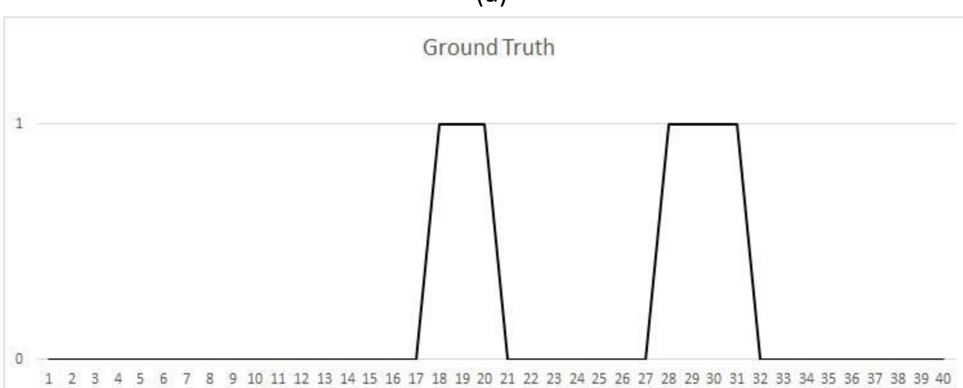


FIG. 5.4: Experiment screenshot which the user walks beside 2 cars.



(a)



(b)

FIG. 5.5: (a) First derivative and Win subtraction variation of magnetometer while walking beside 2 cars with one parking space in between. (b) The corresponding ground truth.

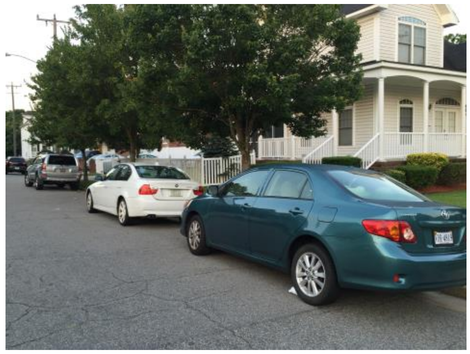
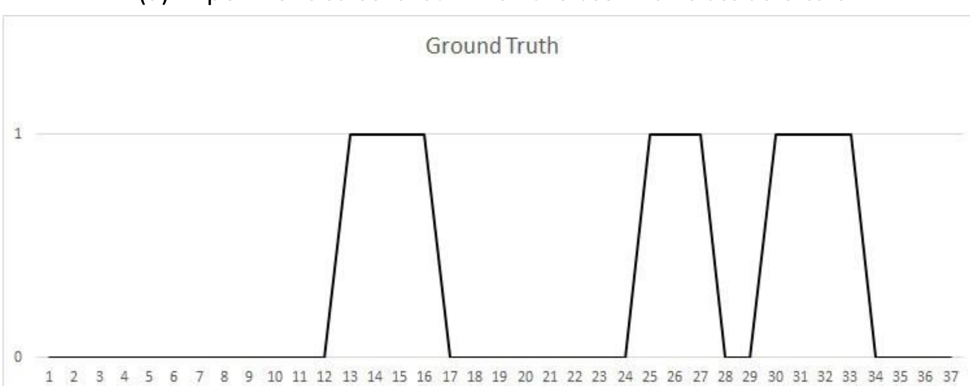


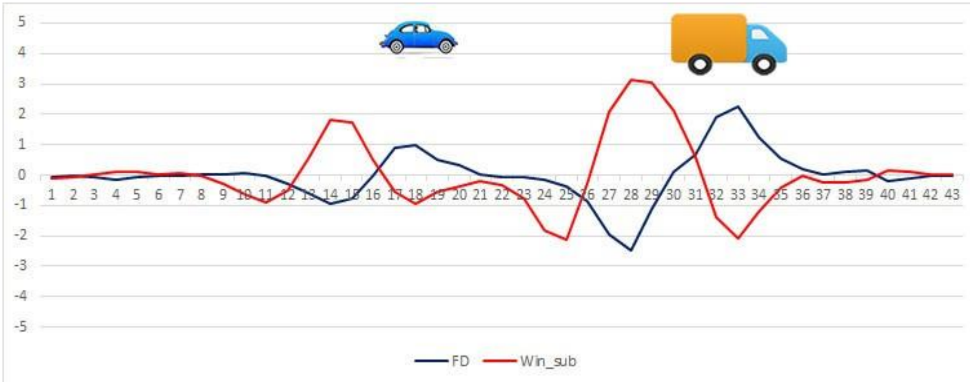
FIG. 5.6

(a) Experiment screenshot which the user walks beside 3 cars.

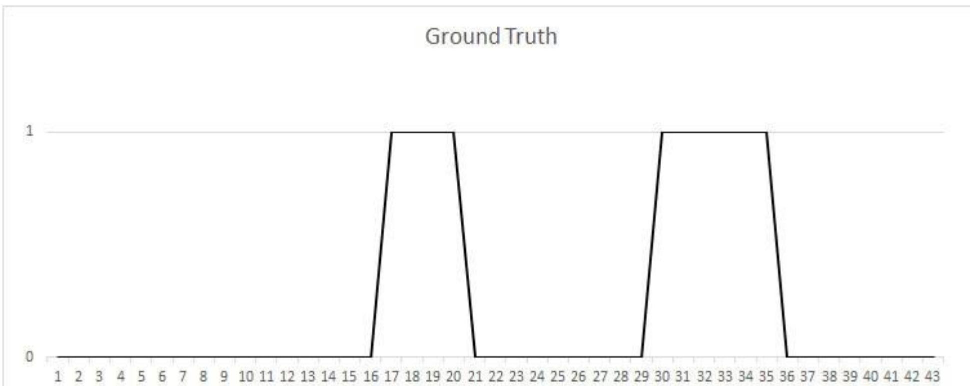


(b)

FIG. 5.7: (a) First derivative and Win subtraction variation of magnetometer while walking beside 3 consecutive cars with one parking spot between them (b) The corresponding ground truth.



(a)



(b)

FIG. 5.8: (a) First derivative and Win subtraction variation of magnetometer while walking beside a car and a truck with a parking spot between them (b) The corresponding ground truth.

- **Car - Car - Park spot - Car**
  In this experiment, we want to compare the behavior change of the magnetometer variation when 2 cars are consecutively park, following by a parking spot and another car with the previous experiment. The situation of the cars with respect to each other is shown in figure 5.6. As it is shown in figure 5.7(a), the 2 consecutive cars are detected separately. The detection is shown in the blue line bumps that are followed by the sharp increase in the red line chart. Figure 5.7(b) is the corresponding ground truth for this experiment.

- **Truck - Park spot - Car** Figure 5.8(a) shows the behavior of first derivative of magnitude and the user walking speed corresponding window subtraction value, while passing beside a car, following by a parking spot and a truck. The purpose of the third scenario is differentiating between big-sized cars, like a bus, truck, or van vehicle, and multiple consecutive cars.
  As it is shown in figure 5.8(a), not only is Magnopark able to differentiate between a

big vehicle and a count of consecutive cars, but the corresponding length of the car is also measurable in our algorithm. Also, the max value of the bumps in the blue chart is higher for the truck comparing to the previous scenarios. This feature might be further used for future works in recognizing the cars model and size for statistics or similar projects.

To better understand the algorithm, we used, tables 1, 2, and 3 show the corresponding detection that is performed with our classifier model in the 3 scenarios.

In table 1, as you can see the set of data for column Win sub, the whole data is a very small positive value or a negative value, except the rows in which the user approaches the car (blue background color cells.) These values, which I refer them as the" Detection Boundary", are large positive values comparing to the rest of the data column. The last large positive value shows the beginning of the car. After algorithm hits the last large value of the detection Boundary, it starts checking the values of FD (first derivative and Fast Fourier Transform. For the length of a window size, the algorithm checks the values of FD, and if while they are positive, they represent the existence of the car. Otherwise, as it is shown in table 5.1, in red color, they do not represent a car.

TABLE 1: Walking beside 2 cars with one parking space in between.

Ave(Mag)	FD	Win_Sub	FFT2	GT	Results
49.0587	-0.1539	0.2806	2.2211	0	n
48.9048	-0.1829	0.0885	2.1827	0	n
48.7219	0.0829	-0.1610	2.1743	0	n
48.8049	0.1268	-0.2088	2.1711	0	n
48.9317	-0.0944	-0.0272	2.2109	0	n
48.8373	-0.0780	0.0662	2.1680	0	n
48.7593	-0.0820	0.0354	2.1830	0	n
48.6772	-0.1215	0.1877	2.1643	0	n
48.5557	-0.0119	0.1270	2.1475	0	n
48.5438	-0.0467	-0.1220	2.1597	0	n
48.4972	0.0662	-0.4357	2.1381	0	n
48.5633	0.1151	-0.9083	2.1544	0	n
48.6785	-0.1687	-1.0053	2.1660	0	n
48.5098	-0.3695	1.0603	2.1426	0	n
48.1403	-0.7932	4.8835	2.1098	0	n
47.3471	-1.1740	4.2794	2.0210	0	n
46.1731	0.6907	-1.8141	1.9012	0	c
46.8638	4.0904	-6.1912	2.4628	1	c
50.9542	3.1054	-4.1947	3.0790	1	c
54.0596	-1.1234	0.4925	3.3100	1	n
52.9362	-2.1008	1.7410	2.6619	0	n
50.8354	-1.0892	0.7207	2.3962	0	n
49.7462	-0.6309	-0.0771	2.2704	0	n
49.1153	-0.3598	-1.2544	2.2061	0	n
48.7556	-0.3686	-1.9081	2.1837	0	n

TABLE 1

Ave(Mag)	FD	Win_Sub	FFT2	GT	Results
48.3870	-0.7080	1.1705	2.1389	0	n
47.6790	-1.6141	4.2131	2.0545	0	n
46.0649	-2.2767	3.6103	2.4185	1	n
43.7882	0.4625	0.0828	2.2486	1	c
44.2507	2.5990	-2.2503	2.3135	1	c
46.8497	1.3336	-1.2833	2.5070	1	c
48.1833	0.5454	-0.7068	2.1097	0	c
48.7287	0.3487	-0.5164	2.1936	0	n
49.0774	0.0503	0.0452	2.2109	0	n
49.1277	-0.1615	0.2272	2.2195	0	n
48.9662	-0.1678	0.0971	2.1738	0	n
48.7984	0.0955	-48.9844	2.1801	0	n
48.8939	0.0657	-0.0657	2.1860	0	n
48.9596	-0.0707	0.0707	2.1933	0	n
48.8889	-48.8889	48.8889	2.1828	0	n

TABLE 2: walking beside 3 cars with one parking space in between

Ave(Mag)	**FD**	**Win_Sub**	**FFT2**	**GT**	**Results**
48.9314	0.0457	-0.0753	2.6939	0	n
48.9770	-0.0131	-0.0592	2.7117	0	n
48.9639	-0.0279	-0.1397	2.7489	0	n
48.9360	-0.0297	-0.1168	2.7077	0	n
48.9063	-0.0724	-0.2249	2.6929	0	n
48.8339	-0.1676	0.0068	2.7036	0	n
48.6663	-0.1465	-0.6737	2.6721	0	n
48.5199	-0.2972	-1.9371	2.6717	0	n
48.2226	-0.1608	-0.7389	2.6228	0	n
48.0619	-0.8201	2.3177	2.5433	0	n
47.2417	-2.2344	3.8875	2.5677	0	n
45.0074	-0.8997	1.8543	2.1384	0	n
44.1077	1.4976	-0.9646	2.0595	1	c
45.6053	1.6531	-1.4760	2.2589	1	c
47.2584	0.9546	-0.7312	2.5571	1	c
48.2130	0.5330	0.1521	2.5948	1	c
48.7459	0.1771	-1.8781	2.6738	0	n
48.9230	0.2234	-2.3034	2.6964	0	n
49.1464	0.6851	-1.5403	2.7016	0	n
49.8315	-1.7010	-2.1974	2.7930	0	n
51.5319	-2.0800	-0.3674	3.0351	0	n
53.5500	-0.8552	1.4017	3.4316	0	n
52.6948	-3.8984	5.2134	3.3699	0	n
48.7965	-2.4474	2.6874	2.5412	0	n
46.3491	0.5465	-1.6003	2.2846	1	c
46.8956	1.3150	-2.3069	2.4810	1	c
48.2106	0.2400	-0.0404	2.5989	1	c
48.4506	-1.0538	1.9805	2.6923	0	n
47.3968	-0.9918	1.6950	2.4727	0	n
46.4049	0.1996	0.1613	2.3501	1	c
46.6045	0.9267	-0.7889	2.3822	1	c
47.5313	0.7032	-0.7563	2.5095	1	c

TABLE 2

Ave(Mag)	FD	Win_Sub	FFT2	GT	Results
48.2344	0.3609	-0.4050	2.6682	1	c
48.5953	0.1378	-48.7738	3.1688	0	n
48.7332	-0.0532	0.0532	3.1631	0	n
48.6800	-0.0440	0.0440	3.1419	0	n
48.6360	-48.6360	48.6360	3.1534	0	n

TABLE 3: walking beside a car and a truck with one parking space in between

Ave(Mag)	**FD**	**Win__Sub**	**FFT2**	**GT**	**Results**
48.491	-0.0536	-0.1129	3.1731	0	n
48.4375	-0.0248	-0.0596	3.1096	0	n
48.4127	-0.0463	0.0232	3.1408	0	n
48.3664	-0.1665	0.1299	3.1061	0	n
48.1999	-0.0844	0.0946	3.1137	0	n
48.1156	-0.0232	0.0315	3.1426	0	n
48.0924	-0.0365	0.0866	3.0935	0	n
48.0558	0.0102	-0.0315	3.0924	0	n
48.0661	0.0083	-0.2815	3.0916	0	n
48.0744	0.0501	-0.6235	3.0885	0	n
48.1244	-0.0213	-0.9197	3.1	0	n
48.1031	-0.2732	-0.4856	3.1078	0	n
47.8299	-0.5735	0.5561	3.0362	0	n
47.2565	-0.941	1.8284	2.9889	0	n
46.3155	-0.7588	1.7303	2.8777	0	n
45.5567	-0.0173	0.5229	2.7537	0	n
45.5394	0.8874	-0.5575	2.761	1	c
46.4267	0.9715	-0.9436	2.838	1	c
47.3982	0.5056	-0.5651	3.009	1	c
47.9038	0.3298	-0.393	3.0743	1	c
48.2336	0.0279	-0.1761	3.1077	0	n
48.2615	-0.0595	-0.3231	3.1084	0	n
48.202	-0.0632	-0.7765	3.1044	0	n
48.1388	-0.1482	-1.8156	3.113	0	n
47.9906	-0.3826	-2.1065	3.0756	0	n
47.6079	-0.8397	-0.2837	3.0448	0	n
46.7682	-1.9638	2.0634	2.9354	0	n
44.8044	-2.4891	3.1549	2.6965	0	n
42.3153	-1.1234	3.0279	2.3455	0	n
41.1918	0.0996	2.1365	2.2576	1	n
41.2915	0.6657	0.5726	2.2812	1	c
41.9572	1.9044	-1.3766	2.305	1	c
43.8617	2.2361	-2.0608	2.6064	1	c
46.0978	1.2383	-1.2021	2.8582	1	c
47.3361	0.5278	-0.4011	2.9953	1	c
47.8639	0.1753	-0.0401	3.0693	0	c
48.0392	0.0362	-0.229	3.0747	0	n
48.0754	0.1267	-0.2365	3.0751	0	n
48.2021	0.1352	-0.1687	3.104	0	n
48.3374	-0.1928	-47.8085	3.1169	0	n
48.1445	-0.1097	0.1097	3.0958	0	n
48.0348	-0.0334	0.0334	3.0769	0	n

5.3.3 SCENARIO 3: DETECTING POLES

In this part, we try to show how Magnopark is able to differentiate between a car and a piece of metal bar like an electric pole. Figure 5.9 shows how Magnopark detect the pole on row number 57. The corresponding detection table is shown in 4. In the table, it is shown that the pole is detected as one sample row. In these cases, we checked the recorded video as the confirmation ground truth in our experiments. Hence, whenever the Magnopark detects one sample row as the car range, we disregard it, because it is related to a small metallic object rather than vehicles.

FIG. 5.9: Magnetometer variation while walking beside cars, and electric poles

TABLE 4: Walking beside 2 cars with one parking space in between.

Ave(Mag)	FD	Win_Sub	FFT2	GT	Results
46.7682313	-1.963793424	2.06343135	2.935350751	0	n
44.80443787	-2.489141527	3.154889176	2.696532245	0	n
42.31529635	-1.123448453	3.027878654	2.345485754	0	n
41.19184789	0.099637926	2.136493019	2.257551644	1	n
41.29148582	0.665747649	0.57255968	2.281155955	1	c
41.95723347	1.904430201	-1.376638049	2.30501224	1	c
43.86166367	2.236130945	-2.060822382	2.606422896	1	c
46.09779462	1.238307329	-1.202115486	2.858238318	1	c
47.33610194	0.527792151	-0.401065733	2.995262008	1	c
47.8638941	0.175308563	-0.040067386	3.069325192	0	n
48.03920266	0.036191843	-0.229029903	3.074683671	0	n
48.0753945	0.126726418	-0.236459464	3.075139221	0	n
48.20212092	0.135241177	-0.168673235	3.104006601	0	n
48.3373621	-0.19283806	0.109733046	3.116946922	0	n
48.14452404	-0.109733046	0.109733046	3.095782662	0	n
48.03479099	-0.033432058	0.033432058	3.076941555	0	n
48.00135893	-0.033432058	0.033432058	3.052888678	0	n
47.96792687	0.126726418	-0.236459464	3.116946922	0	n

TABLE 4					
Ave(Mag)	FD	Win Sub	FFT2	GT	Results
48.09465329	0.135241177	-0.168673235	3.095782662	0	n
48.22989447	-0.19283806	0.109733046	3.076941555	0	n
48.03705641	-0.109733046	0.109733046	3.052888678	0	n
47.92732336	0.010220687	-0.031545459	3.092394938	0	n
47.93754405	0.008293512	-0.28147908	3.091594507	0	n
47.94583756	0.050080764	-0.623537047	3.088524371	0	n
47.99591832	-0.021324772	-0.919667847	3.099951765	0	n
47.97459355	-0.273185568	-0.485631795	3.107765901	0	n
47.70140798	-0.573456283	0.556148777	3.036155224	0	n
47.1279517	-0.940992619	1.828353076	2.988894181	0	n
46.18695908	-0.758817363	1.730290744	2.877655145	0	n
45.42814172	-0.017307506	0.522894095	2.753736053	0	n
45.41083421	1.887360458	-0.557537062	2.76098144	0	c
47.29819467	0.027880236	-0.17612028	3.107709416	0	n
47.32607491	-0.059508417	-0.323125152	3.108416158	0	n
47.26656649	-0.063162736	-0.776543743	3.104394953	0	n
47.20340375	-0.148240045	-1.815553379	3.112955569	0	n
47.05516371	-0.38263357	-2.106507957	3.075635659	0	n
46.67253014	-0.839706478	-0.283741975	3.044788924	0	n

5.4 REAL WORLD EXPERIMENTS

In order to test the correctness of the algorithm we achieved, we asked users to collect multiple data sets while they walk in different streets and merge the data in one huge data set. To confirm the independence of Magnopark from users and their walking habits, we have 2 volunteers participating in our test, 1 male and 1 female. Each of them walked twice beside each car sets. They were allowed to keep the cellphone in their hand in whatever orientation they want. This way we confirm that the orientation of the cellphone does not have any impact on our detection. In both the experiments, we divide the data set as a training part and a testing part. We used 80% of each data set as our training model. Two experiments only differ in the users' walking speed difference. The results of the tests is shown in table 5.

In the first set, as it is shown in table 5, we have 55300 data samples. The corresponding searching window size for the first user is equal to 3. I other words, it takes the user 3 seconds to pass the length of a mid-size car. In this experiment we have 56 cars, among which, Magnopark was able to detect 56 cars which 2 of them was detected wrongly (false positive). In the second experiment, the user corresponding searching window size is equal to four. In this test, we have 94700 data samples, in which we have 98 cars parked beside the streets, and there were 172 parking spots among them. Among 98 cars, Magnopark

TABLE 5: Test Results for car detection

Test Sets	Set 1	Set 2
# Data samples	55300	94700
# Cars	56	98
# Park spots	74	172
True positive	54	96
False positive	2	1
False negative	2	2
True negative	72	171

was able to detect 97 cars which 1 of them is detected wrongly. The false positive, false negative, true positive of both tests are shown in table 5.

The rates for the both test is as follow: True positive rate = TP/(TP+FN); False positive rate = FP/(TN+FP);
Overall success rate = (TP+TN)/(TP+TN+FP+FN);

- Test Set 1:

  True Positive Rate = 54/54+2 = 0.9643

  False Positive Rate = 2/2+72 = 0.027

  Success Rate 1 = 54+72/130=0.9692

  Error rate = 1.0-success rate = 1-0.9692 = 0.03

- Test Set 2:

  True Positive Rate = 96/96+2 = 0.9795

  False Positive Rate = 1/1+171 = 0.0058

  Success Rate 2 = 96+171/270 =0.988

  Error Rate = 1- 0.988= 0.012

Magnopark achieves 97% accuracy in differentiating cars spots and empty spots, regardless of the phone’s orientation, walking habits, and sidewalk conditions.

5.5 CONCLUSION

In this chapter, the experiments were introduced and analyzed. First we conduct some experiments to evaluate if the magnetometer changes are due to the existence of the car and it's not related to other metallic object, including the metal bars that are used in the buildings construction. Then in order to make sure that the system will work with different users and in different places, we conduct the experiments in the same place with different users. Also, we asked users to perform the experiments with different mobile devices. In order to have the ground truth, we asked users to keep the cellphone in their hand in whatever direction the prefer. We test the accuracy of Magnopark in detecting consecutive cars separately. Also we test its accuracy in differentiating between big cars and small consecutive cars. The results show that Magnopark achieves a high rate of accuracy in all the experiments regardless of the users and their version of device.

CHAPTER 6

CONCLUSION

6.1 PRIMARY CONTRIBUTION OF THIS STUDY

In this thesis, I evaluate the feasibility of using the magnetometer sensor in smart phones to detect the parking spots beside the streets. The proposed framework uses the pedestrians' cellphone, while they are walking on the sidewalk. The application process is performed on both the pedestrian cellphones and cloud server. There is no load on the drivers' cellphones who are seeking parking. A key attribute of the Magnopark is that it is not requiring any specific orientation of the cellphone to be held by the pedestrians. In other words, it is orientation independence, and the users can keep the cellphones in their hand, pocket, or even in their bag. Performance evaluation of the experiments and the corresponding results show that this approach is feasible and promising. However, there are some challenges in using smart phones' magnetometer sensors in detecting the vehicles which are mentioned in the following.

Since magnetometer does not measure only earth magnetic field, erroneous estimates are unavoidable in our data collection. This problem always occurs when starting the magnetometer in the cellphone, therefore re-calibration is always needed when the cellphone magnetometer starts to work. If calibration is not performed before using the magnetometer in the application, the noises reduce the accuracy of the system. Calibration should be performed before using the magnetometer, in order to reduce the effect of noises and high variation of the sensor. As an another shortcoming in using the magnetometer sensor, I can point to the effect of the passing cars in the street on the magnetic sensor. Also, in case using the Magnopark in big cities like Manhattan which passing bikes are allowed in the sidewalk, the iron used in bikes affects the detection. Another issue is the distance of the walkers with respect to the cars. The more the distance from the curbs, the less variation in the signals. In brief, the system generates the best accuracy when the cellphone is calibrated, the sidewalk is not bike passing allowed, and the distance between the pedestrian and the sidewalk curbs is not more than 1 to 1.5 meter.

Regardless of all the shortcomings and challenges, we succeed to leverage the cost effective and in-access mobile sensors to differentiate between a vacant and non-vacant parking spot. We evaluate our system under different scenarios where we consider different users with different walking speed and attitude, in different places. Results shows that Magnopark achieves more than 98% accuracy in almost all of the scenarios.

6.2 FUTURE WORKS

For further improvement of Magnopark, we have to map the exact location of the user to the Google Map API. The idea for this improvement is to prevent Magnopark from detecting the non-parking empty spots as a free parking spot. For instance, if the user passes a building entrance parking door, or an intersection, Magnopark detect that space as a vacant parking place.

As an another idea, we can design and develop a complementary application for Magnopark, to be used in the cars, while moving. If Magnopark would be able to detect its neighbor's car, we could develop an application to alarm the driver and prevent him from approaching other cars. This could be a great help to avoid accidents. The last complementary work could be using the Magnopark features on the deaf and blind people cellphones to help them pass the streets. The application would be able to detect the passing cars, and as soon as the traffic light changes to red, the magnetometer variation reduces and become stable. Then, the application would be able to alarm the blind and deaf user to pass the street.

REFERENCES

[1] Compass primer. Wiki. Online; accessed 20 March 2016.

[2] Moscow installs smart parking system SENSIT. http://www.nedapidentification.com/news/news/moscow-installs-sensit.html, 2016. Online; accessed 28 May 2016.

[3] Addison-Wesley. ON-Street parking. http://www.worldsensing.com/solutions/mobility/solutions-en/on-street-parking.html, 1993.

[4] Barter, Paul. Is 30% of traffic actually searching for parking? http://www.reinventingparking.org/2013/10/is-30-of-traffic-actually-searching-for.html, 2013. Online; accessed May 2016.

[5] V. K. Boda, A. Nasipuri, and I. Howitt. Design considerations for a wireless sensor network for locating parking spaces. In *Proceedings 2007 IEEE SoutheastCon*, pages 329–344, 2007.

[6] M. Caliskan, D. Graupner, and M. Mauve. Decentralized discovery of free parking places. In Proceedings of the Third ACM International Workshop on Vehicular Ad Hoc. In *VANET '06 Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 30–39, Sept. 2006.

[7] Collins, S. H. and kuo, A. D. Two independent contributions to step variability during over-ground human walking. *Journal of Applied Physiology*, 2011.

[8] A. Ibisch, S. Stumper, H. Altinger, M. Neuhausen, M. Tschentscher, M. Schlipsing, J. Salinen, and A. Knoll. Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded lidar sensors. In *Intelligent Vehicles Symposium*, pages 829–834. IEEE, 2013.

[9] I. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[10] R. Kmmerle, D. Hhnel, D. Dolgov, S. Thrun, and W. Burgard. Autonomous driving in a multi-level parking structure. In *ICRA*, pages 3395–3400. IEEE, 2009.

[11] F. Kuhlman, D. Sarma, and A. Harback. Vehicle parking spot locator system and method using connected vehicles, Mar. 8 2012. US Patent App. 13/293,761.

[12] Pala, Zeydin and Inanc, Nihat. Smart parking applications using RFID technology. In *RFID Eurasia, 2007 1st Annual*, pages 1 – 3. IEEE, 5-6 Sept. 2007.

[13] R. Panayappan, J. M. Trivedi, A. Studer, and A. Perrig. Vanet-based approach for parking space availability. In *Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '07, pages 75–76, New York, NY, USA, 2007. ACM.

[14] Smith, Aaron. Smartphone Ownership 2013. http://www.pewinternet.org/2013/06/05/smartphone-ownership-2013/, 2013. Online; accessed 28 May 2016.

[15] S. Taghvaeeyan and R. Rajamani. Portable roadside sensors for vehicle counting, clas- sification, and speed measurement. *IEEE Trans. Intelligent Transportation Systems*, 15(1):73–83, 2014.

[16] M. Uddin and T. Nadeem. Spyloc: A light weight localization system for smartphones. In *SECON*, pages 72–80. IEEE, 2014.

Appendices

A. SOURCE CODE

```
    Fs = 100;
// Time vector of 1 second
data = xlsread('last-fft.xlsx');

    for i=0:length(data)/100-1
magnitude = data(:,1);
x = magnitude(i*100+1:(i+1)*100);
// Use next highest power of 2 greater than or equal to length(x) to calculate FFT. nfft=
2ˆ(nextpow2(length(x)));
// Take fft, padding with zeros so that length(fftx) is equal to nfft fftx
= fft(x,nfft);
// Calculate the numberof unique points
NumUniquePts = ceil((nfft+1)/2);
// FFT is symmetric, throw away second half fftx
= fftx(1:NumUniquePts);
// Take the magnitude of fft of x and scale the fft so that it is not a function of the length of x
mx = abs(fftx)/length(x);
// Take the square of the magnitude of fft of x.
mx = mx.ˆ2;
// Since we dropped half the FFT, we multiply mx by 2 to keep the same energy.
// The DC component and Nyquist component, if it exists, are unique and should not be
multiplied by 2.
if rem(nfft, 2) mx(2:end) = mx(2:end)*2;
else
mx(2:end -1) = mx(2:end -1)*2;
end
```

```
// This is an evenly spaced frequency vector with NumUniquePts points. f =
(0:NumUniquePts-1)*Fs/nfft;
mat(i+1,1:10)= mx(1:10);
end
// Generate the plot, title and labels.
plot(f(4:end),mx(4:end));
title('1400-1500');
xlabel('Frequency (Hz)');
ylabel('Power');
spectgrum(a,1024,fs)
```

VITA

Maryam Arab Computer Science
Department Old Dominion
University Norfolk, VA, 23529

Education:

Old Dominion University, Norfolk, VA, USA
Master in Computer Science (2014 - 2016)
Advisor: Prof. Tamer Nadeem (nadeem@cs.odu.edu)
Dissertation Topic: Detecting on street parking spots using the pedestrian sensor enabled cellphones.
Dissertation Committee: Prof. Stephan Olariu, Prof. Michele Weigle GPA 3.78/4.0
Azad University of Tehran, Tehran, Iran
B.S. in Electrical Engineering, 2005

Academic Projects:

Smart Devices Application Development
August 2015 to May 2016
Advisor: Prof. Tamer Nadeem

- Drivers help: developed in Java Scheme, using internal mobile sensors like gyroscope and accelerometer to detect the lane, in which a vehicle is driving, detecting all turns, U-turns, and lane switches.
- Secure File Transfer: developed android application for Transferring messages via vibration for security purposes, encoding and decoding text messages to Morse code and generating vibration patterns.
- Securing Mobile devices: developed android application to generate an alarm in the occurrence of robbery or officious movements.