Received September 2, 2017, accepted October 16, 2017, date of publication November 1, 2017, date of current version November 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2767029

ParkSense: Automatic Parking Positioning by Leveraging In-Vehicle Magnetic Field Variation

PHUC NGUYEN[1], HIEU NGUYEN[2], DONG NGUYEN[2], THANG N. DINH[3], HUNG M. LA[4], AND TAM VU[1]

[1]Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309 USA
[2]Department of Computer Science and Engineering, University of Colorado Denver, Denver, CO 80204 USA
[3]Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284 USA
[4]Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557 USA

Corresponding author: Phuc Nguyen (vp.nguyen@colorado.edu)

ABSTRACT Drivers often park their vehicles without consciously (physically or mentally) noting down the parked location which makes it hard and inconvenient for the users to later locate their vehicles. Existing solutions either require users to explicitly note down the parked position on their mobile device by performing a certain action, such as pressing a button to turn on their global positioning system, or are erroneous and inaccurate. This paper attempts to build *ParkSense*, a system that allows a smart phone to *accurately* and *automatically* "sense," and later navigate to, the position at which the vehicle was parked. We propose to make use of the variations of magnetic fields and electromagnetic fields inside the vehicles to detect when a user stops and turns off his or her vehicle. Our evaluation with an actual implementation of the system on different mobile platforms and operating systems, tested on different mobile devices/phones and car models, shows the detection accuracy of more than 90%, confirming the feasibility of our approach.

INDEX TERMS Parking detection, magnetic field variation, vehicular and wireless technologies, phone embedded sensors, mobile computing.

I. INTRODUCTION

Remembering where one's car was parked has been a problem for many drivers. For example, after a long period of wandering around in a large crowded parking lot or in a busy urban area to find an empty parking spot, drivers tend to not pay attention to the parking location [1]. On another occasion, a traveler hurries and parks his car in a large airport parking lot to catch a flight. A few days later, he arrives at the parking lot not remembering where he had parked his car. Having a system that helps drivers remember where their cars were parked is highly beneficial.

With the presence of GPS localization system and smartphone, an intuitive solution would be to save the position of a car at parking moment using the phone-integrated GPS, and then later to navigate the driver back to the previously saved location through a navigation app on the phone itself. However, with such a solution, the key question is on how to correctly detect the parking moment to precisely save the parked location. Existing solutions resort to having the user explicitly indicate the parking moment to the phone, for example, by pressing a button [2], shaking the phone [3], or taking a picture of the parking position [4]. However, these methods are obtrusive and inconvenient since they would require users action every time they park the car. Another approach is to automatically memorize the parking location when the phone's bluetooth is disconnected [5], [6]. This, however, either requires users to connect the phone to the car or purchase additional hardware. Hence, the class of applications using this explicit approach is not viable, thus has been modestly adopted by users. In its recent update, Google Now released a solution that automatically saves the parking location [7]. When the application detects the transition from driving to walking, it records the device's location at transition and considers that a parking position. While simple, this approach has oversimplified many possible parking scenarios, thus, erroneous. For example, a user exiting a bus or a passenger getting out of a car pool can create a drive-to-walk transition, which triggers Google Now service to save parking location, but no parking position should be recorded. Additionally, in a scenario where a driver parks his car then immediately gets into another car (e.g. for joining a car pool), there will be no drive-to-walk transition detected, wherein Google Now fails to record the parking location [8]. The inaccuracy of the service could lead to frustration of users and reduce the application adoption rate. As a result, being able to autonomously detect the parking moment, without any user interaction with the device upon parking, is a must-have feature of any parking spot localization system. This confusion would have been avoided if such parking spot localization system can identify the actual parking moment of the vehicle.

To fill in this gap, we propose *ParkSense*, a system that leveraging embedded phone sensor to accurately and autonomously record parking locations of a vehicle. In particular, *ParkSense* leverages the electromagnetic signal variation of in-vehicle environment when a driver operates her/his car. It combines the magnetic field data with the output streams from accelerometer and gyroscope, enabling it to track the movement and orientation of the mobile device and the car, and to accurately detect the parking moment to record the parking location. We have identified and utilized a set of signature changes in the car's magnetic field that uniquely occur only when the car is transitioning from moving to parked state. We implemented and evaluated the system on different mobile devices and platforms, and tested on cars from different manufacturers. The results show that our system performs well in different scenarios, environment, devices and vehicles, while consuming less energy compared to other alternatives.

When invoked by the user on an as-needed basis, the *ParkSense* algorithm could navigate its users to the current or previous parking location using its in-app navigation component. For the scope of this paper, we assume the availability of a positioning service at the time of parking, such as GPS for outdoor and wireless fingerprinting [9]–[12], using magnetic based localization [13]–[16], and GPS-based indoor localization as presented in [17]–[22]. We note that localizing the position of the phone is not the focus of this paper. Instead, this paper focuses on when a device should acquire its location and label it a parking position. Our contributions are summarized as following:

- We exploited the electromagnetic variation in in-vehicle environment and identifying key signatures of user operating behavior inside a car, such as turning the engine on/off and hitting the brake.
- We implemented ParkSense to autonomously identify and navigate users to the parked location of their vehicles. The ParkSense algorithm works without any training data.
- We evaluated the system in real world on different mobile devices, operating systems and car models. The system obtained more than 90% accuracy confirming the feasibility of the approach.

The rest of the paper is organized as following. In the next section, Section II, we discuss the background of electromagnetic field and magnetic field inside a car, and its applications. The system design, brake hitting detection algorithm and device's movement detection algorithm are detailed in Section III and Section IV. Section V shows our evaluation methodologies, implementation, and evaluation results. We discuss the related work in Section VII and conclude the paper in Section VIII.

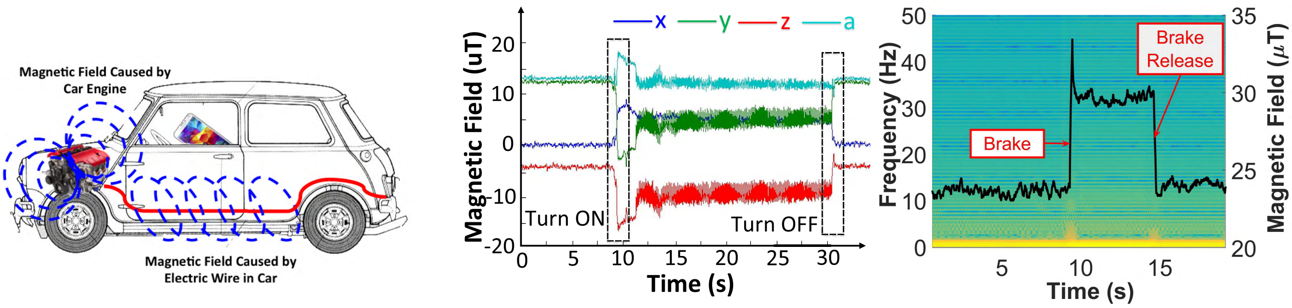
II. BACKGROUND AND OBSERVATIONS

ParkSense relies on the magnetic field variation associated with driver's operating actions, such as hitting a brake, turning the car on and off, etc. When the action happens, the car electrical system causes a certain magnetic changes which can be sensed by magnetometer inside the mobile device. In order to have a deeper understanding of the correlation between the driver action and magnetic field changes, in this section, we present the background of magnetic and electromagnetic field in both general and that in in-vehicle environment. We then discuss our observations on special characteristics of the in-vehicle magnetic field.

A. FUNDAMENTAL OF IN-VEHICLE MAGNETIC FIELD

A *magnetic field* is the magnetic influence of electric currents and magnetic materials. A magnetic field can be produced by either an electrical charge in motion or a permanent magnet. In the former case, it is found that when the electrical charge is in motion, there exists an additional force on the charges that does not appear when the electrical charges are at rest. This additional force, called magnetic force, generated by the presence of the magnetic field in the environment [23]. In the latter case, when there is no moving charges, the magnetic field comes directly from the atoms - from electron spin and orbital states [24]. For in-vehicle environment, there exists a magnetic field generated by both sources. As the magnetic field created by permanent magnet is consistent overtime, we can consider these magnetic field is noise. Note that the electrical and electronics circuits and wires are distributed around the car body, especially at the engine location. The magnetic field changes in the environment is dominant by the charges running in those wires/circuits. In this work, we focus on monitoring the patterns of such magnetic fields changes to identify different stages of the vehicles.

To summary, the electrical and electronic components inside a vehicle are the main sources of in-vehicle magnetic field variations. Let's discuss a few common sources. The magnetic field variations can be generated by the vehicle tires, engine stages, car types, user's movement, location or actions (hit-braking). First, tires are a large contributor to the magnetic field inside a car due to the permanent magnetism in the radial steel bands within the tire [25], [26]. The magnetic field variations from the tires create high magnetic field surges in the driver or co-driver's foot regions. Second, the starter motor and cables cause a high surge in magnetic field when the engine is started. When the user accelerate and decelerate the vehicle, the magnetic field is also changed significantly. Third, different types of car creates different magnetic field variation level [27]. The smaller the car, the higher magnetic field are likely to be captured since the engine is closer to the driver. Forth, different position inside the car can lead to different magnetic field levels. The magnetic field levels are likely to be higher in the front than in the back.



(a) (b) (c)

FIGURE 1. Using a smartphone to sense the magnetic field generated by car and the magnetic readings inside an operating car with some brake-hitting actions. (a) Example illustrating the magnetic field variation around the car body and the phone-based magnetic field variation monitoring. (b) Magnetometer reading from a phone when the car is turned on and turned off. (c) The spectrogram of in-car magnetic field when there is a brake event.

Fifth, user's actions when driving can introduce very high magnetic field variations as well. However, these variations are much smaller than that of magnetic field changes created by the engine. Last but not least important, brake-hitting actions, especially with ABS (Anti-lock Braking System), can also produce high surge in magnetic fields variation.

B. OBSERVATIONS OF IN-VEHICLE MAGNETIC FIELDS

Understanding above fundamental characteristics of the magnetic field of in-vehicle environment, we conduct a set of experiment to validate them. We used smart phone Samsung Galaxy S5 to perform hundreds of magnetic field measurements inside different types of vehicles including cars, buses and light-rails (train). We found that the in-car magnetic fields show some special characteristics and those characteristics, as we will present later, can be utilized to determine the states of the operating car.

Figure 1a illustrates an example of using a smartphone to capture the car magnetic field. The embedded sensors inside the smartphone is used to capture the magnetic field variations generated by different sources inside the car. All of the six sources mentioned above creates the variations in magnetic field that are observable by the mobile phone's sensors. However, we are interested in capturing the magnetic field variations created by different user actions. We found a high correlation between user's actions (or the stages of the car) and the variations in magnetic field. More specifically, Figure 1b shows the magnetometer reading on x-axis when the car is traveling in a straight route and the phone is placed in the driver seat. The magnetic field is recorded starting from when the car is turned on until it is turned off. During the drive, the car is braked three times. As illustrated in the figure, the user actions of turning on/off the car and braking generate surges in the magnetic reading strength. Interestingly, if the phone is placed in other positions inside the car, such as co-driver's seat, backseat, no similar surges is registered on the sensor readings. Also, as the experiment was repeated, we found that the surge caused by the action of turning on/off the car are not really consistent, i.e., the sensor can only capture this type of surge in some of the tests. Meanwhile, the surges generated by brake-hitting actions can always be captured. We consider the results of these experiment as a key motivation to develop a new system that can detect the parking location based on the brake-hitting actions of the users.

Moreover, the magnetic field is also appeared at different range of frequency [25]. We validate this idea by analyzing the magnetic field readings obtained from the phone's sensor at the frequency domain. It is proved that the brak-hitting event is also captured by observe the signal at frequency domain. Figure 1c shows the spectrogram of the magnetic field inside the car when the brake is hit and then released. As can be seen, the magnetic field inside car is mostly at the low frequency range (less than 10Hz) with low power distribution as well. It also implies that detecting the brake-hitting event on frequency domain is very challenging as the difference on power distribution of the surge and noise floor is too small.

III. SYSTEM OVERVIEW

We present an automatic parking positioning system, namely *ParkSense*. *ParkSense* benefits from the fact that common driver actions trigger electrical systems in the car, leading to the aforesaid variation in measured magnetic fields. This enables automatic logging of parking location without requiring any additional hardware. While existing approaches rely on hardware add-ons (e.g. OBD and Bluetooth [2], [3], [5], [6], [28]) for detecting driver actions, *ParkSense* eliminates the need for any additional hardware by utilizing the magnetometer on the smartphone for parking detection. Through our evaluation, *ParkSense* obtains high performance across a diverse set of smartphones and cars in the real world. Also, *ParkSense* consumes less energy than existing applications that need to check GPS signal frequently.

A. CHALLENGES

The utilization of magnetometer and inertial sensor data in a moving vehicle is acutely affected by several factors. Let's discuss some key factors as below:

- **Phone orientation.** Magnetometer readings exhibit sharp variation with sudden changes in phone orientation. Noisy measurements can mask the brake-hitting event that we are interested in. To overcome this challenge, we develop a conversion method to fix the coordinate system of the sensor readings (e.g., Earth Coordinate System) (Section IV-A).

- **Indirect informing vehicle stages from user actions**. To capture the parking event, we identify the combination of different user actions including brake-hitting, and leaving vehicle to detect the parking event. So, it is important to capture the movement when the user takes the phone and moves out the vehicle. We detect this event by analyzing the phone orientation and velocity using IMU data. (Section IV-B, IV-C, IV-D).
- **Tracking user's vehicle but not bus or train.** The mobile phone might be placed inside different types of vehicles such as car, bus, light-rail or even subway. The phone needs to know when to turn on the its positioning algorithm to store the parking location of the user's vehicle. We develop a detection algorithm to identify whether the phone is inside a tracking vehicle or not by recognizing the patterns of the magnetic field variations at different types of vehicles (Section IV-E).

B. PROPOSED SYSTEM

We design a set of algorithms allowing *ParkSense* to addresses above challenges. We outline the overview of ParkSense system as in Figure 2. ParkSense constitutes an *Phone Pose Estimation* module to sense if the user is in a vehicle, followed by *In-Car Verification* to ensure that the user is driving her/his vehicle (she/he is not in a train). Once the system has determined that the smart phone user is indeed the driver, it detects brake-hitting events using the smart phone magnetometer (applying *Brake-Hitting Detection* algorithm) to monitor user braking behavior and find the last brake before user changes to parking mode. Then, ParkSense tracks the walk-away movement by a *Phone-Picking-Up Detection* algorithm. Once the parking and walk-away events are detected, the user location at this instance is recorded as the car parking location.

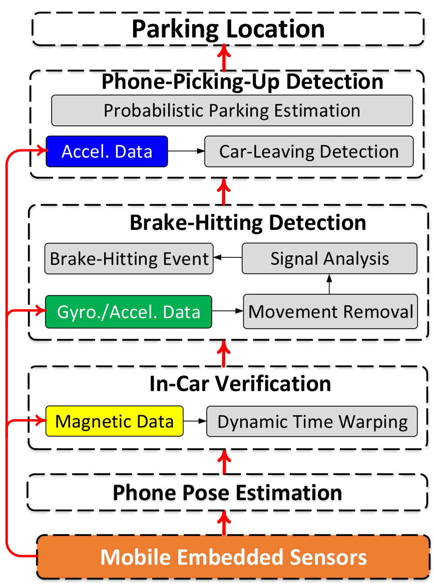


FIGURE 2. System overview.

We define three main states of the phone including *stay_still*, *in_vehicle* and *on_hand* where the phone is placed inside the vehicle and carried by user, respectively. When the *Phone Pose Estimation* component detects that the device in the vehicle or not (*in_vehicle* state), the *In-Car Verification* component will be activated to identify whether the device is in the car or on the train. If it is confirmed that the device is in car, the *Brake-Hitting Detection* (BHD) component is triggered. This component is designed to detect the brake-hitting event signature in the data sequence. When a brake is detected successfully, two components - *Phone-Pick-Up Detection* (PPD) and *Probabilistic Parking Estimation* - are activated. The *Phone-Pick-Up Detection* component is for determining the moment the driver parks and gets out of the car. When a Phone-Pick-Up event is found, the GPS location is recorded. Importantly, ParkSense will only register this as the parking location after the device changes its state from *in_vehicle* to *on_hand* (the driver leaves the vehicle) within a specific time window using *Car-Leaving Detection*. The *Probabilistic Parking Estimation* component is designed to address the problem that the BHD component is unable to detect the last brake-hitting event for parking or the PPD algorithm does not identify the phone-pick-up action. In this case, the position that our ParkSense records is the coordinates where the phone changes its state to *on_hand*.

1) PHONE POSE ESTIMATION

It is important to clearly define when to start and stop collecting data from sensor due to limitation of storage and energy on the mobile device. To implement this requirement, we adopt existing solutions, *i.e.*, `Activity Recognition API` on Android OS and `CMMotionActivity` on iOS. We want to take advantage of these optimized solutions to reduce phone's processing power and storage memory. The energy consumption and resources usage for each state depend on optimization of the provided API. Note that the processing power and storage energy are proportional to the processing delays. Table 1 shows the delay of each state, which can be used to identify the power consumption as well storage memory required.

TABLE 1. Average delays (in seconds) of the state transitions by Activity Detection APIs, where *S*, *INV*, *ONH* stand for *stay_still*, *in_vehicle*, *on_hand* state of the device.

	$S \Rightarrow ONH$	$ONH \Rightarrow INV$	$INV \Rightarrow ONH$
Android	21.97	23.93	21.33
iOS	11.87	6.25	9.00

2) IN-CAR VERIFICATION

In large cities, the metro system is an important public transportation. These electrically operated vehicles, while in operation, generate similar magnetic surges to those in personal vehicles. Thus it is necessary to differentiate between riding the metro and driving a car. Through experiments, we found that the surges of magnetic field inside the train have bigger variances than those in cars. Understand this characteristic, we adopt Dynamic Time Warping [29] to measure the difference of a selected magnetic sequence with ones from each vehicle. If the difference is lower than a specific threshold (0.8 out of 1 in this case), the phone is determined to be on car and vice versa.

3) BRAKE-HITTING DETECTION

To detect the brake hitting events for inferring user's behavior, ParkSense needs to remove unwanted magnetic surge caused by phone's movement our approach. To solve this problem, we propose to use the data collected from accelerometer and gyroscope to isolate unwanted device movements. After removing unwanted surges from the magnetic field data, we normalize the data by using the approximate derivative method. The derivation of the magnetic field data helps to remove most noises from sensor measurement.

4) PHONE-PICK-UP DETECTION

To store the parking location, we design a mechanism to trigger the GPS properly. Note that a wrong trigger would not only create in accurate of parking location detection but also waste energy of retrieving GPS data. We observe that when the driver walks out of the car and pick-up the phone, a distinct signal is registered on accelerometer's readings. We can take advantage of this to trigger the GPS to save the parking location.

5) PROBABILISTIC PARKING ESTIMATION

We employ the Bayesian probabilistic model to classify who is the phone's holder: *the car driver, the passenger on a car* or *the passenger on a bus*. Given the observations that some brake-hitting actions are detected, the model computes the probabilities of each case. Assume that the driver is identified, our system can always record the position of the driver when the phone changes its state from *on_vehicle* to *on_hand* (the phone now knows that the user is using his vehicle). According to our observation and experiment results, using this position as the car parking position, we obtained less accuracy ratio than the position at which the phone-pick-up is detected.

IV. THE PARKSENSE SYSTEM

A. PHONE POSE ESTIMATION

Most of the today's smartphones equip with an inertial sensor which normally outputs data including acceleration $\boldsymbol{a}^t$ from its Accelerometer, angular rate $\boldsymbol{w}^t$ from its Gyroscope, Earth magnetic field $\boldsymbol{o}^t$ from its Magnetometer, and quaternion $\boldsymbol{q}^t$.

The Inertial's quaternion is a vector $\boldsymbol{q}^t(x, y, z, w)$, and the conjugation $(\boldsymbol{q}^t)^*$ of the $\boldsymbol{q}^t$ is a vector obtained as follows:

$$(\boldsymbol{q}^t)^* = (-\boldsymbol{q}^t(x), -\boldsymbol{q}^t(y), -\boldsymbol{q}^t(z), \boldsymbol{q}^t(w)). \tag{1}$$

The Euler angles of rotation, *Roll* (α^t), *Pitch* (β^t) and *Yaw* (γ^t) rotate along the X, Y, Z axis, respectively, and they can be obtained as follows:

$$\begin{bmatrix}\alpha^t\\ \beta^t\\ \gamma^t\end{bmatrix} = \begin{bmatrix}f_1(\boldsymbol{q}^t(x))\\ f_2(\boldsymbol{q}^t(x))\\ f_3(\boldsymbol{q}^t(x))\end{bmatrix} \tag{2}$$

where,

$$\begin{aligned}
f_1(\boldsymbol{q}^t) &= atan2[2(\boldsymbol{q}^t(x)\boldsymbol{q}^t(y) + \boldsymbol{q}^t(z)\boldsymbol{q}^t(w)),\\
&\qquad 1 - 2((\boldsymbol{q}^t(y))^2 + (\boldsymbol{q}^t(z))^2)],\\
f_2(\boldsymbol{q}^t) &= arcsin[2(\boldsymbol{q}^t(x)\boldsymbol{q}^t(z) - \boldsymbol{q}^t(w)\boldsymbol{q}^t(y))],\\
f_3(\boldsymbol{q}^t) &= atan2[2(\boldsymbol{q}^t(x)\boldsymbol{q}^t(z) + \boldsymbol{q}^t(y)\boldsymbol{q}^t(z)),\\
&\qquad 1 - 2((\boldsymbol{q}^t(z))^2 + (\boldsymbol{q}^t(w))^2)].
\end{aligned}$$

The local velocity of the phone motion in its body frame can be obtained as follows:

$$\boldsymbol{v}^t = \int \boldsymbol{a}^t \, dt. \tag{3}$$

From (3), we can compute the local position of the phone by taking the integral of $\boldsymbol{v}_b^t$. However, we are interested in localizing the phone in the global Earth coordinate system such as the North East Down (NED) system [30], [31]. Hence, we can obtain the acceleration of the phone motion in the NED system as:

$$\boldsymbol{a}^t_{NED} = \boldsymbol{q}^t \cdot \boldsymbol{a}^t \cdot (\boldsymbol{q}^t)^*. \tag{4}$$

Then the real motion of acceleration in the NED system can be obtained as follows:

$$\boldsymbol{a}^t_{NED} = \boldsymbol{a}^t_e - \boldsymbol{g}_e, \tag{5}$$

where the $\boldsymbol{g}_e$ is the gravitational acceleration vector:

$$\boldsymbol{g}_e = (0.0, 0.0, 9.8m/s^2).$$

Similar to Equ. (3), the velocity of the phone motion in the NED system can be obtained as follows:

$$\boldsymbol{v}^t_{NDE} = \int \boldsymbol{a}^t_{NDE} dt. \tag{6}$$

From Equ. (6), 3D position of the phone motion in the NED system in real-time can be obtained as follows:

$$\boldsymbol{p}^t = \int \boldsymbol{v}^t_{NED} dt. \tag{7}$$

The phone localization (7) may return accumulative errors along the travel. We can combine with the GPS and human foot or car motion model via Extended Kalman Filter (EKF) to improve the accuracy even in GPS outage environments [32], [33].

B. BRAKE-HITTING DETECTION ALGORITHM

Our Brake-Hitting Detection Algorithm aims to correctly detect the moment when the driver hits the brake using the change in magnetic field caused by the action. As mentioned, the magnetic strength (measured at the driver seat area, inside a running car) surges when the brake is pressed. However, sudden movements of the phone can also cause similar magnetic surge to happen. To solve this problem, we use the *approximate derivative* method which is normally used to measure the variations of discrete signals. For the outputs to be independent of the phone's orientation, we combine the derivatives of all three axes, i.e., $|diff_{mag}| = \sqrt{diff^2_{mag_x} + diff^2_{mag_y} + diff^2_{mag_z}}$. Where, $diff_{mag_x} = \frac{mag_x(t)-mag_x(t-1)}{\Delta_t}$, $diff_{mag_y} = \frac{mag_y(t)-mag_y(t-1)}{\Delta_t}$, and $diff_{mag_z} = \frac{mag_z(t)-mag_z(t-1)}{\Delta_t}$, and the sampling interval, Δt, is selected empirically.

To detect those surges, we compare the derivative value outputs $diff_{mag}$ with a threshold. If any output surpasses this threshold, our algorithm will count the data sample corresponding to that output as a signature of one brake-hitting event. We apply this technique for a sequence of magnetic data collected in a driver-seat area inside our Toyota Camry 2000 with smart phone Samsung Galaxy S5. As shown in Figure 3, a surge at second 65^{th} corresponds to the moment that the driver pressed the brake of the car.

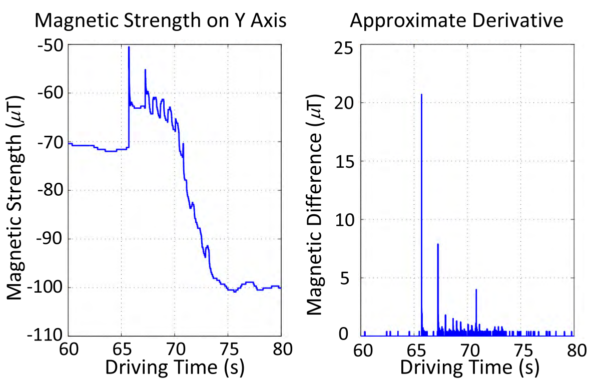


FIGURE 3. An example of approximate derivative calculation for magnetic sequence in our Toyota Camry 2000 with one brake-hitting event.

There are three problems with using the approximate derivative method to detect the magnetic surges. Firstly, there can be a burst of surges in magnetic strength due to the phone shaking when the brake is engaged. As can be seen in Figure 3, there are two surges with magnitude 18μT (at the second 65^{th}) and 6μT (at the second 70^{th}) close together that match with one brake-hitting event. Secondly, device movements such as tilt, shake, rotation, or swing also produce sudden magnetic changes. Thirdly, because this threshold depends on the accuracy of the phone's magnetometer and the strength of electrical currents inside the car, it must be dynamically set.

To address the first problem, we count a burst of surges within a time frame as a one surge to match with one brake-hitting. This time frame is started from the first detected surge and lasts for few seconds. For the second problem, we apply a technique to smooth out the magnetic reading, i.e., to remove the data sections resulted by some unwanted device movements. The technique computes the values of phone's gyroscope and accelerometer readings. Whenenver the gyroscope and accelerometer readings are both greater than a set of pre-determined thresholds, our technique removes that corresponding magnetic data section. For the third problem, we choose the threshold by an algorithm called Threshold Learning Algorithm (TLA).

More specifically, TLA runs once before the ParkSense application is used in the smartphone. The driver puts the phone in the driver area and drives the car in a smooth and straight route. From the start to the end, he creates N brake-hitting events. He enters this number into the app. TLA computes the derivative of the obtained magnetic sequence. It sets a large threshold value and then reduces this value gradually until it counts N brake-hitting events. It reduces threshold more until it counts $N+1$ brake-hitting events. TLA takes the average of those two found values as a threshold that is used for the whole program lifetime (assume that the car is still the same). For example, in Figure 4 (Middle), there are 6 brake-hitting events and the threshold is found at 10 μT by LTA.

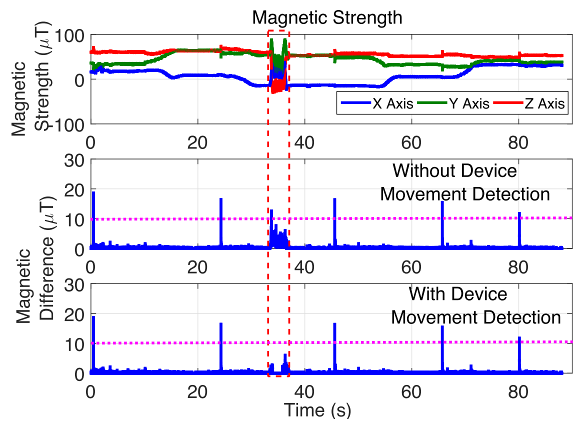


FIGURE 4. An experiment with Honda Accord 2003: the BHD algorithm removes the data section in which the phone is shaken while keeping the surges representing brake-hitting events at second 1^{st}, 24^{th}, 46^{th}, 66^{th}, and 80^{th}.

The detail discussion is shown in Algorithm 1. The system computes the *gravity* vector on three axes x, y, z by applying Low-Pass Filter (LPF) to the raw accelerometer readings (line 1). In line 2, the algorithm computes the magnitude of the phone's acceleration created by the user. Line 3, the rotation rate of the phone around the Z axis of the earth coordinates is found. The algorithm determines the combined magnetic derivative on all three phone's axes in line 4. Given the acceleration and the rotation rate are both greater and pre-set thresholds, the algorithm *smooth out* the corresponding magnetic data section in lines 5 and 6. Otherwise, whenever the derivative is greater than a threshold, thus a surge created by brake-hitting action is found and the algorithm outputs the time of the event in lines 7 and 8.

Figure 5 shows the output of our algorithm for a case the phone is in the driver-seat and passenger-seat areas inside our running car (Honda Accord 2003). The car was driven following a rectangular route. We plot five magnetic surges that greater than the threshold value are presented the Figure 5a at

Algorithm 1: Brake-Hitting Detection algorithm

Input: Realtime raw sensor readings including magnetometer, gyroscope, and accelerometer mag_{raw}, $gyro_{raw}$, a_{raw} and magnetic, acceleration and rotation threshold values: TH_M, TH_A, TH_G (obtained by TLA)

Output: Brake-hitting event Timestamp T_{BH}

Use Low-Pass Filter to extract the gravity vector from magnetometer readings $gravity_{(x,y,z)} \leftarrow LPF(a_x, a_y, a_z)$

Compute the acceleration of the device's movement by subtracting the fixed gravity: $a_{linear} \leftarrow a_{raw} - gravity$
$|a_{usr}| \leftarrow \sqrt{a^2_{linear_x} + a^2_{linear_y} + a^2_{linear_z}}$

Calculate the direction Z of the gravity vector and the device's rotation rate around that vector:
$unit_z \leftarrow gravity_{(x,y,z)}$ $rr_z \leftarrow gyro_{raw} * (unit_z^T * unit_z)$

Get the combined derivatives of the raw magnetometer readings on all three axes:
$diff_{mag(x,y,z)} \leftarrow mag_{raw(x,y,z)}(t) - mag_{raw(x,y,z)}(t-1)$
$|diff_{mag}| \leftarrow \sqrt{diff^2_{mag_x} + diff^2_{mag_y} + diff^2_{mag_z}}$

if $|a_{usr}| > TH_A \wedge |rr_Z| > TH_G$ **then**
    Smooth out the magnetic surges created by unwanted device movements: $|diff_{mag}| = 0$

if $|diff_{mag}| > TH_M$ **then**
    Detect a brake-hitting even if there is a surge: $T_{BH} \leftarrow t_{current}$

second 10^{th}, 24^{th}, 45^{th}, 63^{th} and 85^{th} corresponds to 5 brake-hitting events. The last brake-hitting event indicates that the driver stopped and parked the car. We omit the engine's turned off event in these figures. On the other hand, Figure 5b shows the data for the passenger-seat area. In this case, there isn't any spike of magnetic data that is higher than the threshold. Note that fine-grained localization to identify the location of the phone inside a car is not needed. We only need to implement a simple binary classification algorithm to identify whether driver or passenger is using the phone. In addition, implementing fine-grained localization requires more computational cost as well as power consumption. In short, we want to implement a simple solution but yet sufficient to solve the problem.

In another testing, we let the driver shake the phone intensively within a few seconds while driving. As shown in Figure 4, the magnetic readings have big variations at the time frame from second 34^{th} to 38^{th} that correspond with the period the driver shakes the phone. Our Brake-Hitting Detection algorithm is able to remove this magnetic data section so that the output does not indicate the surge in that time frame as a brake-hitting event.

C. PHONE-PICK-UP DETECTION (PPD) ALGORITHM

When the driver stops the car then walks out of the car with a phone in his/her hand or pocket, the built-in accelerometer captures the acceleration of the device's movement. We call this movement as phone-pick-up movement and we are interested in an approach to detect that action. We quantify the action in a simple way to compute an approximate value of the acceleration. Assume that it takes a period of time Δt for the driver to get out of the parked car and S is the distance the phone is moved from a place inside the car to out of the car, the acceleration of the device can be computed approximately as: $|a| = (2 * (S/2))/\Delta t^2 = S/\Delta t^2$.

This acceleration depends on how quickly the driver gets out of the car, the height of both the car and the driver. Figure 6 illustrates a simplified phone movement out of the car. In the first and second half of the movement, the phone has acceleration vector $\overrightarrow{a}$ and $\overrightarrow{-a}$, respectively. More specifically, the magnetometer reading on the acceleration axis varies from $0(m/s^2)$ to $0.4(m/s^2)$ and then from $-0.4(m/s^2)$ to $0(m/s^2)$. In our algorithm, we use $0.2(m/s^2)$ as the threshold.

PPD, as presented in Algorithm 2, utilizes only accelerometer readings to identify this action. More specifically, the algorithm isolates the fixed component of the acceleration vector, while computing the acceleration on the Z axis. Within a time frame $[T_{BH}, T_{BH}+\delta]$, if the maximum variation of accelerometer readings are larger than a desired threshold, the algorithm will decide that this event is a phone-pick-up event. Note that the variation is calculated from a maximum to a minimum obtained accelerometer readings, consecutively.

Algorithm 2: Phone-Pick-Up Detection Algorithm

Input: Realtime raw accelerometer readings acc_{raw}; time window size $[T_{BH}, T_{BH} + \delta]$; and acceleration threshold value TH_{PPU}

Output: Phone-Pick-Up Event Timestamp T_{PPU}

Use Low-Pass Filter to extract the gravity vector from magnetometer readings: $gravity_{(x,y,z)} \leftarrow LPF(a_x, a_y, a_z)$

Compute the acceleration on Z axis of the device's movement by subtracting the fixed gravity:
$a_{linear} \leftarrow a_{raw} - gravity$; $unit_z \leftarrow gravity_{(x,y,z)}$;
$a_z \leftarrow a_{raw} * (unit_z^T * unit_z)$

Find maximum and minimum acceleration values on Z axis in the time window: Find $max_{a_z[T_{BH},T_{BH}+\delta]}$; Find $min_{a_z[T_{BH},T_{BH}+\delta]}$

if $((max_{a_z} - min_{a_z}) > TH_{PPU}) \wedge (index_{max} > index_{min})$ **then**
    Output the current time $T_{PPU} \leftarrow t_{current}$

return T_{PPU}

D. PROBABILISTIC PARKING EVENT ESTIMATION

To further improve our algorithm, we develop a probabilistic parking event estimation model. Let $P(DS)$, $P(PS)$, and $P(B)$ denote belief probabilities that the device is in the driver-seat (DS) area in cars, in the passenger-seat (PS) area in cars, and on the bus. We are interested in finding the probability that the phone is in the driver-seat area given some

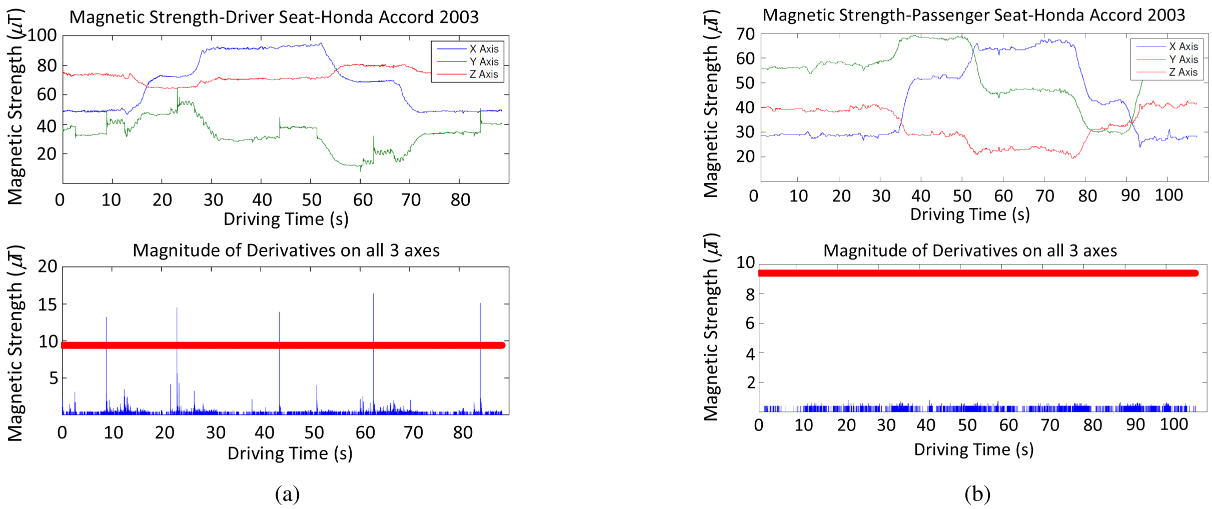


FIGURE 5. Experiments with Honda Accord 2003: Magnetic strength and its derivative on the driver-seat and passenger-seat area. The signals respect to the threshold (the solid bold line). (a) Driver-seat area. (b) Passenger-seat area.

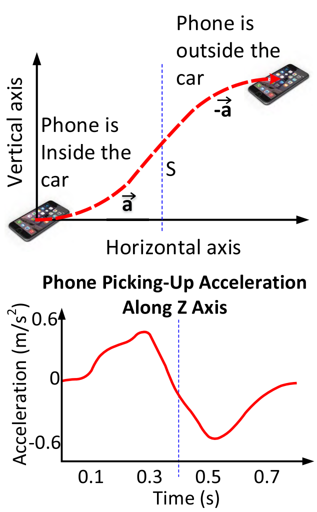


FIGURE 6. A simplified phone movement out of car and its acceleration measured on one axis by the accelerometer of the phone.

observations of N_S magnetic surges.

$$P(DS|N_S) = \frac{P(N_S|DS)P(DS)}{P(N_S|DS)P(DS) + P(N_S|PS)P(PS)}$$

Given the priori probabilities for the devices $P(DS)$, $P(PS)$, and $P(B)$, by comparing the posteriori probabilities. If $P(DS|N_S) > P(PS|N_S)$ and $P(DS|N_S) > P(B|N_S)$, the divice is in the driver-seat area. Once the device position in cars is identified, the moment when the state of the device changes from IN_CAR mode to ON_FOOT mode is the parking moment and the GPS ouputs will be recorded.

E. IDENTIFYING ON_TRAIN STATUS

We use the Dynamic Time Warping algorithm (DTW) to distinguish *on_train* status and *on_vehicle* status. The current approach is to calculate the DTW for the signals that obtained when the user is on the light rails or on the car. The referenced point is decided through experiment. In our scenarios, we collect the signal from both light rail and cars. The reference signal is achieved when we run algorithm to identify the value that could separate the two data set precisely. Again, we avoid to use machine learning or other complicated solution to reduce the computational cost and energy consumption. The results show that finding the references from the data set is sufficient for all scenarios.

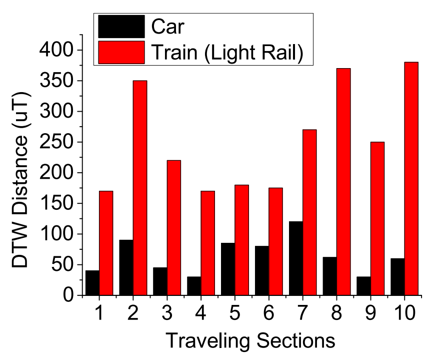


FIGURE 7. DTW distance between a reference sequence and the sequences by *on_train* status is much bigger than one by *on_vehicle* status.

The DTW algorithm is designed to measure the minimum distance between two given sequences of signal data [29]. This minimum distance indicates the alignment of two sequences: the smaller the distance is, the more similar the two sequences are. In particular, we compare the distance of one selected reference sequence with the derivative sequences of the raw magnetic data measured on cars and on light rails. In our *Parksense* system, we select one sequence $diff_{mag_{ref}}$ of *on_vehicle* magnetic data as a reference sequence. Figure 7 illustrates that the distances of the referenced sequence $diff_{mag_{ref}}$ with the *on_train* magnetic data is much bigger than the *on_vehicle* magnetic data.

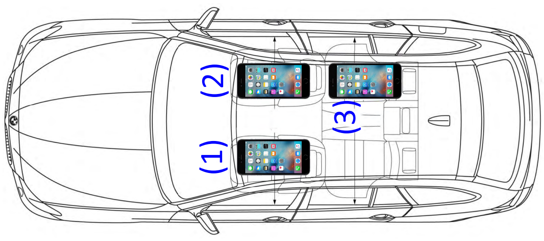


FIGURE 8. Phone's position inside the car.

V. PERFORMANCE EVALUATION

A. EXPERIMENTAL SETUP

We perform experiments using three different phones: Samsung Galaxy S5, LG G3, and iPhone 6 on several cars: Toyota Camry 1999, Toyota Camry 2005, Honda Accord 2003, and Solara Coupe 2004. The phone is placed in different places inside cars: driver seat, co-driver seat and backseat (Figure 8). Testing is done in different parking locations and routes in Denver, CO, U.S. The experiment was conducted on a limited number of cars due to the available of our current resource. However, we believe that the system works across different type of cars. We aim to conduct experiment later on electric cars, we think that ParkSense would obtain even higher accuracy due to the increase of magnetic field sources. We evaluate our ParkSense in these following test cases:

1) TEST CASE #1

We test our system in the driver-seat area of different cars with different phones as mentioned above. In this scenario, we are interested in measuring the accuracy of the ParkSense system to correctly detect the parking event given the driver having the phone, by the following formula: $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, where TP, TN, FP, and FN represent the number of true positives, true negative, false positive, and false negative.

2) TEST CASE #2

In this scenario, we consider the cases in which the passenger has the phone and the ParkSense system should not be able to detect the parking events. In other words, the metric to measure in this scenario is the accuracy of ParkSense that does not detect the parking event. We use the same formula above to evaluate our system.

3) TEST CASE #3

The system is tested on buses and light-rail. Similar to test case #2, the ParkSense system should not be able to detect the parking or stopping events of the vehicle, i.e., the bus or light-rail.

Testing is done for more than 200 tests with a total of more than 150 miles of driving both on highway and in-town. For each session, the driver hits the brake pedal at the stop signs, the red traffic lights, the turning points (i.e, left turn, right turn, or U-turn), and the parking spot. The parking moment is when the driver hits the brake to stop the car, turns off the car, picks up the phone and walks out of the car.

B. RESULTS EVALUATION

1) DETECTION ACCURACY

For *test case* #1, the accuracy of the ParkSense system in detecting the parking event is presented in Figure 9. Results show that ParkSense works well with the accuracy rate around 90% for different types of cars and phones. The cases, in which the system failed, are either because ParkSense misunderstood a temporary stop as a parking event or because ParkSense does not detect the parking event. Even though we only have chance to thoroughly evaluate our system with some four different models as above. The similar results of experiment are obtained with a short distance test on Lexus ES 300 and another Toyota Camry. In addition, as most of Samsung phone use the same compass sensor (*e.g.*, Yamaha YAS532B [34]), the similar results are expected to obtain with other mobile phone's model that use the same sensor chip.

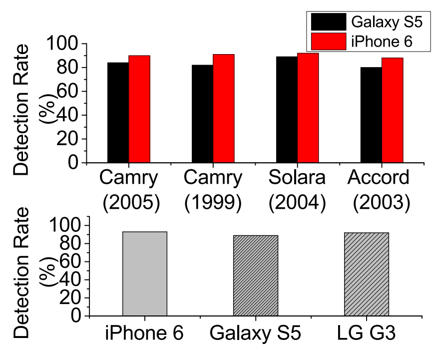


FIGURE 9. Accuracy of correctly detecting the parking events for *test case* #1.

For test case #2 and #3, we have the results presented in a common Table 2. ParkSense performs more than 90% accuracy in average. The experiment are conducted with 4 different light rail lines C, D, F, H [35] starting from Denver downtown, Colorado, U.S. The phone we use to test in both scenarios is Samsung Galaxy S5. The inaccuracy of ParkSense in these cases are due to the similarities in magnetic strength variances of the train during some of the route sections. Because of this, the DTW method is unable to distinguish whether the passenger is in a car or on a train.

TABLE 2. Testing results of the Parksense system in the scenarios that does not need to record the vehicle's parking positions.

Test case	Accuracy (%)
Accord 2003 Passenger Seat	90
Light Rail	94.8
Bus	91.7

2) ESTIMATION DELAY

We are also interested in estimating the delay from the actual time that the car is parked until ParkSense registers the parking event. Figure 10 shows the delays in test cases using Samsung Galaxy S5 on Honda Accord 2003 and Toyota Solara 2004. As can be seen in this figure, the average delay

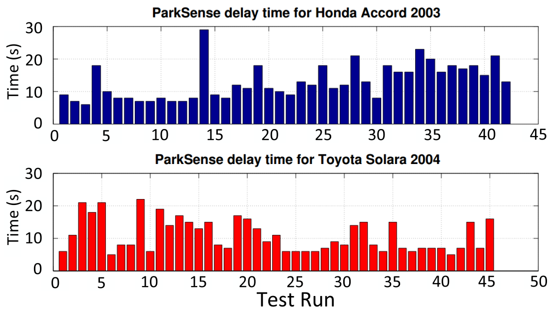


FIGURE 10. ParkSense time delay from actual parking time.

for Toyota Solara is usually less than that of Honda Accord. From the results of our experiment, as Toyota Solara emits more electromagnetic signal than Honda Accord, ParkSense takes less time to detect the signal from Toyota vehicle.

3) SENSITIVITY ANALYSIS

As Brake-Hitting Detection Algorithm is considered as the core component in our design for detecting parking event. We conduct an analysis on the detection accuracy of this method. Two phones (Samsung Galaxy S5 and iPhone6) are used to test on Toyota Solara 2004, two Toyota Camry (1999, 2005). Galaxy S5 can detect up to 99% the number of the brake-hitting event while iPhone 6 can detect up to 95%. Note that the performance of other components inside ParkSense such as Phone Pose Detection, In-car Verification, and Phone-Picking-Up Detection are also very reliable. They contribute to the high overall performance of the system as presented earlier.

4) ENERGY CONSUMPTION

We conduct experiment to evaluate the energy consumption of ParkSense and compare with another automatic parking locator application - Google Now (Launcher) on Samsung Galaxy S5. Figure 11 shows the distribution of power consumption of two applications, where $\mathcal{E}_{\text{GN}}$, $\mathcal{E}_{\text{ParkSense}}$, $\mathcal{E}_{\text{Default}}$ represent the total energy consumed by Google Now, ParkSense, and default applications running on the device, respectively. The experiment duration is 15 minutes and the sampling interval is 0.2 milliseconds. Even though

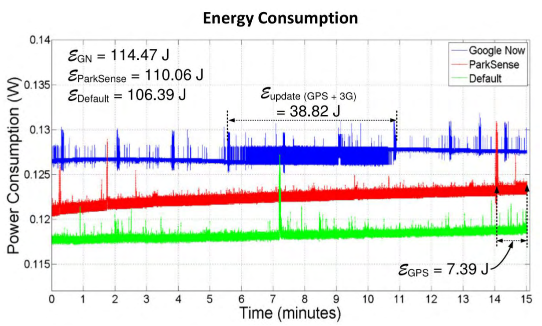


FIGURE 11. Energy consumed by ParkSense and Google Now Launcher.

Google Now consumes less energy by running as a background service, it always requires GPS data for localization function. Google Now also needs to access data network (via WiFi/LTE/3G) to connect to server in order to response to user's requests, which consumes $\mathcal{E}_{\text{update}} = 38.82$ J. ParkSense, on the other hand, only needs GPS information for the last minute (requires $\mathcal{E}_{\text{GPS}} = 7.39$J) to activate its localization feature without accessing to the internet. The power consumption is measured using Monsoon power monitor [36]. We then calculate the energy consumed by those applications. The results of our experiments show that ParkSense consumes only 110.06J, while Google Now consumes 114.47J after the same amount of time. Meanwhile, Samsung Galaxy S5 requires 106.39J ($\mathcal{E}_{\text{Default}}$) to run its default applications and services. Therefore, as a result, the Google Now consumes 1.83 times more energy compared to ParkSense. In short, ParkSense consumes much less energy compared with state-of-the-art approach, Google Now, due to its simplicity. The mobile device doesn't need to run power hungry user interface with real-time internet connectivity. ParkSense utilize the sensor reading on the mobile device and run on the background to compute the parking location.

VI. DISCUSSION

A. ACQUIRING LOCATION

As mentioned in Section I, ParkSense relies on the ability to acquire the current location of the mobile device at parking, either through GPS for outdoor environment or other indoor positioning systems. While we can certainly remove this assumption by integrating an existing positioning system into ParkSense, such as using Wi-Fi finger printing positioning for indoor environments, it is not within the scope of this work.

B. IMPACT FROM ENVIRONMENT NOISES

Technically, the compass readings may be affected by different noise sources such as earth's magnetic field or a loud speaker inside a car. However, the magnetic field noise from environment is usually less than $4\mu T$ and the signature of the magnetic signal generated by the car engine is very distinct, as reported in Figure 7, the noise doesn't affect to the performance of the system. Moreover, the magnetic readings may also be affected from a loud speaker which is operated by vibrating a magnet coil. However, because the vibration of the magnet coil inside a speaker is operating at high frequency (in 10s kHz range) and the car's magnetic signature is generated at low frequency (less than 1Hz), the noise from speaker doesn't affect to the robustness of ParkSense.

C. OTHER APPLICATIONS

We observed that the magnetic field signal is sensitive to its location inside the car. Hence it could be used for in-car localization without any additional hardware component. This application is useful to detect and possibly prevent the use of mobile devices by driver while driving. Furthermore, because the magnetometer, gyroscope and accelerometer consume the least amount of energy compared to alternative sensors such as location sensors (i.e. GPS) or camera, this approach promises the minimum energy usage.

Another possible application leveraging in-vehicle magnetic field variation is for safety reminder for driver when driving. An example of such applications is having an application that reminds users of what to do after a certain action inside the car. For example, the application would remind the user to turn his head to check for blind spot after turning on the turning signal; or remind the user to lock the car after the car is parked.

VII. RELATED WORK

A. PARKING POSITIONING

Parking location saving application for smartphones have attracted much attention nowadays. The most common approach of these applications is to have user manually saving the parking position by pressing a button [2], [28], shaking the phone [3], or taking a picture of the parking position [4]. These all require users' actions, and therefore, become ineffective if users forget to do the required actions. Another approach is to memorize the parking location when the phone's Bluetooth is disconnected [5], [6]. This, however, either requires users to connect the phone to the car or purchase additional hardware. Recently, Google released Google Now application [7] with the feature of automatically saving parking position without any requirement or additional hardware. The idea behind this is to detect the user's action of leaving a parked vehicle; however users still have to manually change the mode of operation to car in order for the app to correctly save the parking position of the car.

B. IN-CAR POSITIONING

Distinguishing the driver with the passengers is important to estimate the proper parking location. There are some techniques to distinguish drivers and passengers that have been presented recently by using stereo infrastructure [37], exploiting accelerometer information with [38], or utilizing a fusion of embedded sensors [39]. In addition, Chandrasekaran *et al.* [40], [41] present a technique to obtain the mobile speed. This information together with the floor plan of the parking space would be combined to approximate the relative parking location.

C. DRIVER'S BEHAVIOR ANALYSIS

Towards the related work in detecting driver's behavior, phone's sensors have been used extensively to analyze the driving styles and behaviors [42]–[45] by observe the acceleration data. With ParkSense, we propose for the first time a technique utilizing the information extracted from phones magnetometer to detect the interested events from the car, which goes beyond the traditional objective of mangetometer (orientation estimation [46]). Following this direction, magnetometer has been used to monitor the change in the magnetic field to detect passing vehicles [47], or to recognize human activities [48].

VIII. CONCLUSION

This paper presents an accurate and autonomous parking positioning system called ParkSense that takes advantage of the unique in-vehicle magnetic field signatures captured by the integrated magnetometer in today's smart phones. We proposed a set of algorithms and probabilistic model to overcome challenges caused by the noisy environment inside vehicles, the movement of the mobile device, and the movement of the vehicle itself. We developed ParkSense app for Android and iOS and evaluated it on real-world setting with different mobile device hardware and platforms. Tested on different car models and manufacturers, our results show the accuracy of 90% in parking detection rate. ParkSense can differentiate a user walking out of a bus from a driver that actually parks his or her car then walks away, which is not possible with existing motion-based technique, such as that found in Google Now.

Since the magnetic field signature are correlated to the position of the mobile device inside a vehicle and also tied in with the driver behaviors, many potential applications can be developed from this observation. For example, it can be used for detecting driver's phone use to prevent distraction during driving, or reminding a driver what to do after a certain action (e.g. reminding the driver to turn his head to the right to check for blind spot after he signals for a right turn). As another potential research direction, we can enable ''car identification'' application since each car generates magnetic field differently. However, given the inaccuracy of current embedded-phone, the dedicated magnetometer is needed to capture the signal at a greater details in order to realize the idea. In addition, when the signal is captured with a higher resolution, machine learning algorithm will be applied to improve the performance of the parking position system. In addition, we will use ParkSense for electronic cars. ParkSense would be more robust with electric cars because the magnetic noises are easier to observed due to the large number of electric components inside the car.

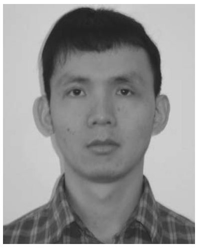
REFERENCES

[1] I. Inc Survey. (2017). *Top Driving Embarrassments*. Accessed: Oct. 24, 2017. [Online]. Available: http://goo.gl/pSeDJj
[2] BitPedal. (2017). *Parking Recall (Version 1.32)*. Accessed: Oct. 24, 2017. [Online]. Available: http://goo.gl/Nepuaj
[3] E. Kim. (2017). *Car Locator (Version 4.22)*. Accessed: Oct. 24, 2017. [Online]. Available: http://goo.gl/umWz80
[4] eLibera. (2016). *Find My Car*. Accessed: Aug. 26, 2016. [Online]. Available: http://goo.gl/oBDUwQ
[5] Hexter Labs Software. (2017). *Car Finder AR (Version 3.2.5)*. Accessed: Oct. 24, 2017. [Online]. Available: http://goo.gl/DTR6Gr
[6] FMC. (2017). *Find My Car Smarter*. Accessed: Oct. 24, 2017. [Online]. Available: http://goo.gl/eRJWIU
[7] Google. (2017). *Google Now*. Accessed: Oct. 24, 2017. [Online]. Available: http://goo.gl/IkoqEv
[8] Google. (2017). *Google Developers*. Accessed: Oct. 24, 2017. [Online]. Available: http://goo.gl/2N6PyM
[9] Z. Yang, C. Wu, and Y. Liu, ''Locating in fingerprint space: Wireless indoor localization with little human intervention,'' in *Proc. ACM MobiCom*, 2012, pp. 269–280.
[10] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, ''802.11 user fingerprinting,'' in *Proc. ACM MobiCom*, 2007, pp. 99–110.

[11] J. Xiong, K. Sundaresan, and K. Jamieson, "ToneTrack: Leveraging frequency-agile radios for time-based indoor wireless localization," in *Proc. ACM MobiSys*, 2015, pp. 537–549.
[12] Z. Zhang *et al.*, "I am the antenna: Accurate outdoor ap location using smartphones," in *Proc. ACM MobiCom*, 2011, pp. 109–120.
[13] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism," in *Proc. ACM MobiSys*, 2011, pp. 141–154.
[14] S. Suksakulchai, S. Thongchai, D. M. Wilkes, and K. Kawamura, "Mobile robot localization using an electronic compass for corridor environment," in *Proc. IEEE SMC*, Oct. 2000, pp. 3354–3359.
[15] J. Haverinen and A. Kemppainen, "A global self-localization technique utilizing local anomalies of the ambient magnetic field," in *Proc. IEEE ICRA*, May 2000, pp. 3142–3147.
[16] D. Navarro and G. Benet, "Magnetic map building for mobile robot localization purpose," in *Proc. IEEE ETFA*, Sep. 2009, pp. 1–4.
[17] S. Nirjon, J. Liu, G. DeJean, B. Priyantha, Y. Jin, and T. Hart, "COIN-GPS: Indoor localization from direct GPS receiving," in *Proc. ACM MobiSys*, 2014, pp. 301–314.
[18] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proc. ACM UbiComp*, 2008, pp. 114–123.
[19] N. Agarwal *et al.*, "Algorithms for GPS operation indoors and downtown," *GPS Solutions* vol. 6, no. 3, pp. 149–160, Dec. 2002.
[20] G. Dedes and A. G. Dempster, "Indoor GPS positioning—Challenges and opportunities," in *Proc. IEEE Semiannu. Veh. Technol. Conf.*, Sep. 2005, pp. 412–415.
[21] F. van Diggelen and C. Abraham, *Indoor GPS Technology*. Dallas, TX, USA: CTIA Wireless-Agenda, 2001.
[22] F. van Diggelen, "Indoor GPS theory & implementation," in *Proc. IEEE Position Location Navigat. Symp.*, Apr. 2002, pp. 240–247.
[23] D. Jiles, *Introduction to Magnetism and Magnetic Materials*, 2nd ed. Boca Raton, FL, USA: CRC Press, 1998.
[24] G. L. Pollack and D. R. Stump, *Electromagnetism*. Reading, MA, USA: Addison-Wesley, 2001.
[25] S. Stankowski, A. Kessi, O. Bécheiraz, K. Meier-Engel, and M. Meier, "Low frequency magnetic fields induced by car tire magnetization," *Health Phys.*, vol. 90, no. 2, pp. 53–148, Feb. 2006.
[26] S. Milham, B. J. Hatfield, and R. Tell, "Magnetic fields from steel-belted radial tires: Implications for epidemiologic studies," *Bioelectromagnetics*, vol. 20, no. 7, pp. 440–445, 1999.
[27] A. Philips and J. Philips. (2014). *Transport, Travel and EMFs*. [Online]. Available: http://goo.gl/ap8ckf
[28] nomadrobot. (2017). *MyCar Locator Free (Version 4.0)*. Accessed: Oct. 24, 2017. [Online]. Available: http://goo.gl/Z7UidE
[29] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2001, pp. 1–11.
[30] G. W. Collins, II, *The Foundations of Celestial Mechanics*. New York, NY, USA: The Pachart Foundation DBA Pachart Publishing House, 2004.
[31] X. Yun, J. Calusdian, E. R. Bachmann, and R. B. McGhee, "Estimation of human foot motion during normal walking using inertial and magnetic sensor measurements," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 7, pp. 2059–2072, Jul. 2012.
[32] L. Van Nguyen and H. M. La, "Real-time human foot motion localization algorithm with dynamic speed," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 6, pp. 822–833, Dec. 2016.
[33] H. M. La *et al.*, "Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1655–1664, Dec. 2013.
[34] Yamaha. (2017). *Yamaha YAS532B*. Accessed: Oct. 24, 2017. [Online]. Available: https://goo.gl/uL4EY2
[35] Denver. (2017). *Denver Light Rail Map*. Accessed: Oct. 24, 2017. [Online]. Available: http://www.rtd-denver.com/LightRail_Map.shtml
[36] (2017). *Monsoon Solution Inc*. Accessed: Oct. 24, 2017. [Online]. Available: http://goo.gl/yHWK7t
[37] J. Yang *et al.*, "Sensing driver phone use with acoustic ranging through car speakers," vol. 11, no. 9, pp. 1426–1440, Sep. 2012.
[38] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin, "Sensing vehicle dynamics for determining driver phone use," in *Proc. ACM MobiSys*, 2013, pp. 41–54.
[39] H. Park, D. Ahn, M. Won, S. H. Son, and T. Park, "Poster: Are you driving?: non-intrusive driver detection using built-in smartphone sensors," in *Proc. ACM MobiCom*, 2014, pp. 397–400.
[40] G. Chandrasekaran *et al.*, "Vehicular speed estimation using received signal strength from mobile phones," in *Proc. ACM Ubicomp*, 2010, pp. 237–240.
[41] G. Chandrasekaran *et al.*, "Tracking vehicular speed variations by warping mobile phone signal strengths," in *Proc. IEEE PerCom*, Mar. 2011, pp. 213–221.
[42] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 1609–1615.
[43] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles, "Driving behavior analysis with smartphones: Insights from a controlled field study," in *Proc. ACM MUM*, 2012, pp. 36:1–36:8.
[44] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2012, pp. 234–239.
[45] J.-H. Hong, B. Margines, and A. K. Dey, "A smartphone-based sensing platform to model aggressive driving behaviors," in *Proc. ACM CHI*, 2014, pp. 4047–4056.
[46] S. Ayub, A. Bahraminasab, and B. Honary, "A sensor fusion method for smartphone orientation estimation," in *Proc. 13th Annu. Post Graduate Symp. Converg. Telecommun., Netw. Broadcast.*, Liverpool, U.K., Jun. 2012.
[47] S. Lee, D. Yoon, and A. Ghosh, "Intelligent parking lot application using wireless sensor networks," in *Proc. IEEE CTS*, May 2008, pp. 48–57.
[48] M. Zhang and A. A. Sawchuk, "A preliminary study of sensing appliance usage for human activity recognition using mobile magnetometer," in *Proc. ACM Ubicomp*, 2012, pp. 745–748.

PHUC NGUYEN received the B.S. degree from Vietnam National University in 2010 and the M.S. degree in computer science from Sungkyunkwan University, South Korea, in 2014. He is currently working toward the Ph.D. degree in computer science at the University of Colorado Boulder, Boulder, CO, USA. His research interests include mobile computing, mobile sensing, and wireless technology. He received an Outstanding Graduate Student award from the Department of Computer Science and Engineering, University of Colorado Denver, in 2016, and the Best Paper Award at ACM MobiCom-S3 2016 and ACM SIGMobile Research Highlights 2017.

HIEU NGUYEN received the B.S. degree in telecommunication engineering from the Hanoi University of Science and Technology, Vietnam, in 2013. He is currently working toward the Ph.D. degree at the DECENT Laboratory, Department of Computer Science and Engineering, University of Colorado Denver, Denver, CO, USA. His research interests include computer networks and high speed communication system utilizing NFV and SDN technologies.



DONG NGUYEN received the B.S. degree in electrical engineering from the Hanoi University of Technology, Vietnam, in 2000, the M.S. degree in computer engineering from Yonsei University, Seoul, South Korea, in 2005, and the Ph.D. degree in electrical and computer engineering from Oregon State University, Corvallis, OR, USA, in 2009. His research interest is mobile computing, mobile sensing, and wireless networking.

THANG N. DINH received the Ph.D. degree in computer engineering from the University of Florida in 2013. He is currently an Assistant Professor at the Department of Computer Science, Virginia Commonwealth University. He has authored or coauthored over 60 papers in conferences and journals, including the top venues such as SIGMOD, MOBICOM, INFOCOM, and the IEEE TRANSACTIONS ON NETWORKING. His research focuses on optimization and security challenges in complex systems, especially social networks, and cyber-physical systems. He serves as an Associate Editor of *Computational Social Networks* journal and the Guest-Editor for *Theoretical Computer Science* and the *Journal of Combinatorial Optimization*. He co-chaired COCOON and CSoNet and served on TPC of several conferences, including the IEEE INFOCOM.

HUNG M. LA received the B.S. and M.S. degrees in electrical engineering from the Thai Nguyen University of Technology, Thai Nguyen, Vietnam, in 2001 and 2003, respectively, and the Ph.D. degree in electrical and computer engineering from Oklahoma State University, Stillwater, OK, USA, in 2011. From 2011 to 2014, he held a postdoctoral position and then was a Research Faculty member at the Center for Advanced Infrastructure and Transportation (CAIT), Rutgers University, New Brunswick, NJ, USA. He was a Key Team Member of the CAIT Team that developed the Robotics Assisted Bridge Inspection Tool for the Federal Highway. He is currently an Assistant Professor and the Director of the Advanced Robotics and Automation Laboratory, Department of Computer Science and Engineering, University of Nevada, Reno, NV, USA.

TAM VU received the B.S. degree in computer science from the Hanoi University of Technology, Vietnam, in 2006 and the Ph.D. degree in computer science from WINLAB, Department of Computer Science, Rutgers University, New Brunswick, NJ, USA, in 2013. He is currently an Assistant Professor and the Director of the Mobile and Networked Systems Laboratory, Department of Computer Science, University of Colorado Boulder. He received the Google Faculty Research Award in 2014 for his work in Chrome browser authentication. He received the Best Paper Award for inventing new form of communication, called capacitive touch communication, at ACM MobiCom 2012. He was also a recipient of the ACM MobiCom 2011 Best Paper Award for his work on driver phone use detection. His research also received wide press coverage, including CNN TV, *The New York Times*, *The Wall Street Journal*, National Public Radio, *MIT Technology Review*, Yahoo News, among other venues.