

## Research Article

# A Low-Cost Vehicle Anti-Theft System Using Obsolete Smartphone

Bang Liu <sup>1</sup>, Nianbo Liu <sup>1</sup>, Guihai Chen,<sup>2</sup> Xili Dai <sup>1</sup>, and Ming Liu <sup>1</sup>

<sup>1</sup>University of Electronic Science and Technology of China, Chengdu, China

<sup>2</sup>Shanghai Jiao Tong University, Shanghai, China

Correspondence should be addressed to Nianbo Liu; liunb@uestc.edu.cn

Received 28 June 2017; Revised 11 November 2017; Accepted 11 December 2017; Published 29 March 2018

Academic Editor: Daniele Riboni

Copyright © 2018 Bang Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In modern society, vehicle theft has become an increasing problem to the general public. Deploying onboard anti-theft systems could relieve this problem, but it often requires extra investment for vehicle owners. In this paper, we propose the idea of PhoneInside, which does not need a special device but leverages an obsolete smartphone to build a low-cost vehicle anti-theft system. After being fixed in the vehicle body with a car charger, the smartphone can detect vehicle movement and adaptively use GPS, cellular/WiFi localization, and dead reckoning to locate the vehicle during driving. Especially, a novel Velocity-Aware Dead Reckoning (VA-DR) method is presented, which utilizes map knowledge and vehicle's turns at road curves and intersections to estimate velocity for trajectory computation. Compared to traditional dead reckoning, it reduces accumulated errors and achieves great improvement in localization accuracy. Furthermore, based on the learning of the driving history, our system can establish individual mobility model for a vehicle and distinguish abnormal driving behaviors by a Long Short Term Memory (LSTM) network. With the help of ad hoc authentication, the system can identify vehicle theft and send out timely alarming and tracking messages for rapid recovery. The realistic experiments running on Android smartphones prove that our system can detect vehicle theft effectively and locate a stolen vehicle accurately, with average errors less than the sight range.

## 1. Introduction

Nowadays, quick and easy transport has been an essential part of our daily life. As the dark side of this phenomenon, vehicle theft has become one of the costliest property crimes of modern society. According to the U.S. National Insurance Crime Bureau, a vehicle theft occurs every 33 seconds; that is, roughly one million vehicles each year wind up in the hands of thieves at a cost of nearly 6.4 billion dollars. The indirect charge is even much more expensive. To each and every one of us, it is a numbers game where one has to pay hundreds of dollars each year in higher insurance premiums.

In order to prevent theft, some people install anti-theft systems in their cars, such as LoJack, ProScout, and TravelEyes2. Most systems combine wireless communication and GPS localization techniques. After a car has been stolen, the device in it will pinpoint the location, send radio signals to the owner, and help the owner or the police retrieve the car. However, the GPS-based anti-theft systems have their own

problems. Firstly, with a cost of \$400 to \$1300, they are too expensive to many users. Secondly, they cannot operate in tunnels, garages, underground parking lots, or dense urban areas, where GPS signals are screened or disturbed. Thirdly, they are easy to defeat if the thief knows where the device is. For example, the thief can simply cover it with a metal can and then localization will be impossible.

Besides vehicle theft, we are also suffering from a surfeit of electronic waste. In modern society, people upgrade their mobile phones every 18 months on average, either because of the incentives in provider contracts or to keep up with the fashion trend. Only in the U.S., there are 125 million phones discarded each year, which results in 65,000 tons of waste and represents a significant and growing environmental impact. Since many obsolete phones are smartphones, which still function normally, they can be reused as a vehicle-mounted anti-theft device, instead of expensive special equipment. Today's smartphones are not only programmable but also come with network interfaces and a rich

set of embedded sensors, which enable great sensing and communicating abilities to play the role of protector for vehicle safety.

In a vehicle tracking approach, the difficulty first arises from how to locate a vehicle with a smartphone inside. A naive solution is to mount the smartphone on vehicle dashboard, where it can easily receive GPS signals for positioning. This design is incredibly fragile, for professional thieves will disable all suspicious devices after entering a car. Hiding the smartphone deep in vehicle body could relieve this problem, but it means GPS signals may be unavailable and more positioning methods should be considered. The RF-based methods, such as cellular/WiFi localization, are far less accurate than GPS and can make large errors in the regions without dense deployment of cell towers and access points (APs). With the support of motion sensors such as accelerometer and gyroscope, dead reckoning can be used to estimate a trajectory from a known past position to current position. However, this method suffers from fast error accumulation over time, for small errors in the measurement of acceleration are double integrated into increasingly larger errors in displacement. It is found that smartphone motion sensors have surprisingly low accuracy, generating errors up to 100 meters within a minute, at a rate super-linear with time [1].

Even if the vehicle can be accurately located during the whole driving, the problem of theft detection still exists, as how to decide a driver, or a driving, is unauthorized. Most driver verification methods based on biometric features, such as vision, voice, fingerprint, and iris, cannot be used for a smartphone in vehicle body. On the other hand, it is impossible to ask the owner to manually turn off the system before driving and turn it on after driving. Even a user-defined timetable (e.g., the system is merely turned on at night) will finally bring terrible user experience, for long unprotected periods or tiresome false alarming.

In this paper, we propose the idea of PhoneInside, which does not need a special device but leverages an obsolete smartphone to build a low-cost vehicle anti-theft system. After being fixed in vehicle body with a car charger, the smartphone can detect vehicle movement and adaptively use GPS, cellular/WiFi localization, and dead reckoning to locate the vehicle during driving. Especially, a novel VA-DR method is presented, which utilizes map knowledge and vehicle's turns at road curves and intersections to estimate velocity for trajectory computation. Compared to traditional dead reckoning, it reduces accumulated errors and achieves great improvement in localization accuracy. Furthermore, based on the learning of the driving history, our system can establish individual mobility model for a vehicle and distinguish abnormal driving behaviors by an LSTM network. With the help of ad hoc authentication, the system can identify vehicle theft and send out timely alarming and tracking messages for rapid recovery. The realistic experiments running on Android smartphones prove that our system can detect vehicle theft effectively and locate a stolen vehicle accurately, with average errors less than the sight range.

The original contributions that we have made in the paper are highlighted as follows:

- (i) To the best of our knowledge, we are the first to consider the using of obsolete smartphone in vehicle anti-theft. Furthermore, our scheme provides insights to waste reuse for property safety in a simple and feasible way.
- (ii) We develop a novel dead-reckoning method for vehicle tracking. Road curves and intersections are free and everywhere, which contribute distinct velocity-aware landmarks for trajectory computation.
- (iii) We deploy a novel deep learning-based method as LSTM network to identify vehicle theft. Human trajectories show a high degree of temporal and spatial regularities, while a stolen vehicle does not follow the mobility regularities of its owner, which provides a reasonable basis for abnormal detection.

The remainder of this paper is structured as follows. Section 2 briefly discusses the related work, while Section 3 presents the overview of our scheme. In Section 4, we explain the design of PhoneInside step by step, including orientation, sensor errors, motion detection, matching, positioning, and details of modelling and theft decision. Section 5 evaluates our system through realistic experiments, and Section 6 summarizes the paper.

## 2. Related Work

There have been many studies on vehicle tracking and anti-theft. In this section, we only explore the most relevant work in the areas.

*2.1. Mobile Tracking.* Vehicle tracking is often discussed with mobile tracking, due to the wide deployment of smartphones among drivers and passengers. Some phone tracking services, such as Apple MobileMe, and Samsung Dive, can help a user find a missing smartphone, which supports GPS and cellular/WiFi-based localization. However, cellular localization is not really accurate. The accuracy of this method ranges from 100 meters to several kilometers, mainly depending on the number of cell towers detected [2, 3]. WiFi-based localization suffers from low coverage, especially in rural areas and many developing regions. Thus, dead reckoning has become a popular localization method for mobile tracking.

In [4], vehicle trajectory is estimated by an odometer and a gyroscope, and a single data-fusion process based on particle filter performs both positioning and map matching. In [5], a context-aided Kalman filter for urban vehicle navigation is proposed, where contextual knowledge as sensor quality and driving context is used to carry out a continuous estimation and correction of sensor drift errors. In [6], a cellular network-based vehicle tracking scheme is proposed, in which vehicle velocity and heading direction, obtained by a speedometer and a heading sensor, are directly employed in the optimization for position determination.

vTrack [7] builds a system to estimate a users' trajectory and travel time by WiFi and GPS, in which a Hidden Markov Model- (HMM-) based map matching scheme and travel time estimation method are performed. Since the HMM and

Viterbi algorithm are robust to noise, later studies [1, 8–11] also adopt these techniques to model a vehicle trajectory over a map area. Thiagarajan et al. [8] describe a crowd-sourced cooperative transit tracking system using an individual smartphone. For subway or vehicles in tunnels, it detects vehicle mobility by accelerometer and calculates the most likely vehicle location through dead reckoning. CTrack [9] achieves energy-efficient trajectory mapping using raw position tracks obtained largely from cellular base station fingerprints, which also fuses data from low-energy accelerometer (to detect movement) and magnetometer (to detect turns) on smartphones. WheelLoc [10] provides a continuous outdoor location service without GPS, which discusses driving and cycling patterns, respectively, and generates a rough mobility trace with accelerometer and magnetometer. AutoWitness [11] designs a small INS tag that can be embedded into stolen property. In mobile tracking, the tag establishes a sequence of movements, stops, and turns and estimates trajectory through dead reckoning, and then the trajectory data are sent to a server to complete a map matching process. Bumping [1] proposes an inertial navigation method for driving in parking garages, which exploits smartphone's accelerometer to detect bumping caused by speed bumps to find reference positions. It also proves that such bumping can provide precise velocity estimates for positioning.

Although we adopt accelerometer and gyroscope, our scheme is different from previous approaches. Our scheme tries to enhance dead reckoning, by introducing more measurable landmarks, such as road curves and intersections, to reduce accumulated errors, not simply using advanced algorithm in map matching.

In addition, ad hoc localization of vehicle [12–14] comes as a natural result of vehicular ad hoc networks. Each vehicle estimates intervehicle distance and localizes itself among its neighbors, which aims at accurate relative positioning for driving safety. Generally, these approaches assume onboard device equipped on every vehicle to support ad hoc communication, which may need a long time for wide deployment of hardware.

**2.2. Vehicle Anti-Theft.** Many recent research schemes have been demonstrated to achieve vehicle electronic immobilizers. Guo et al. [15] designed an automotive security system to disable an automobile and its key auto systems through remote control when it is stolen, in which four layers of security features are written in the form of firmware and embedded on the electronic control units. Sadagopan et al. [16] presented an anti-theft control system use of an embedded chip that has an inductive proximity sensor, which senses the key during insertion and sends a text message to the owner's mobile device stating that the car is being accessed. Hongzhi et al. [17] proposed an auto-guard system which combined RFID and the global mobile communication network. The system could identify the car owner quickly and then realize the function of keyless entry and keyless start-up at the same time. The infrared sensors and vibration sensors completed the monitoring function. Although they deploy different hardware equipment, they

exploit cellular communication and GPS localization and have similar disadvantages as the anti-theft devices discussed above.

At the same time, the use of biometric measures is heavily promoted for driver identification purpose. POLLUX [18] is proposed as an anti-theft system based upon machine vision technology, which can locate and recognize the driver's face and send the unauthorized driver's image to car owner or police through CDMA or GPRS networks. In CMAC [19], driver profiles are created using the cerebellum model articulation controller feature map taking inputs from the brake and gas pedals pressure signals. These features can be used to verify drivers using multilayered perceptron as classifiers. Some researches [18, 20] try to deploy smartphone sensors to evaluate driving behaviors for driving safety, but there is no evidence to support that such driving recognition is accurate enough to identify the driver.

Alternatively, approaches based on ad hoc communication to guarantee vehicle safety have also been considered. In SVATS [21], each vehicle has a wireless sensor node, and each node is monitored by its neighbors, for identifying possible vehicle thefts by detecting unauthorized vehicle movement. In SPARK [22], some sensors are employed as parking lot infrastructure, to surveil and manage the whole parking lot through vehicular ad hoc networks. Once a vehicle is illegally leaving the parking lot, the infrastructure can quickly detect the anomaly. Similarly, these approaches require onboard device on every vehicle, which is still an obstacle for current users.

### 3. System Overview

To any vehicle anti-theft system, there are two basic tasks: alarming and tracking. The former is triggered when theft happens, while the latter usually lasts a long period to locate the moving vehicle. Thus, the PhoneInside system carries on tracking and theft detecting in each driving and sends alarming and locating messages to the owner once the vehicle is thought stolen. We list several overall considerations that meet these requirements, as the basis of our research.

**3.1. Hardware Installation.** As shown in Figures 1(a) and 1(b), an obsolete smartphone and a car charger (which is priced at \$0.99 in ebay) are installed in one corner of vehicle trunk as an onboard anti-theft device. The car charger is connected to the wires of taillight, so that the smartphone can stay in this secret place with sufficient power supply. Of course, the device can be embedded into any other place in vehicle body, where the power supply is available.

#### 3.2. Design Principles

**3.2.1. The System Should Be Cost-Efficient.** Since the hardware discussed above is very cheap, the spending mainly comes from the daily using, including communication and energy costs. Although most smartphones support 3G/4G communication, the subscription of 3G/4G data plan is often not trivial. The short message service (SMS) is the better



FIGURE 1: Hardware installation. (a) Smartphone and recharger. (b) Smartphone in vehicle body.

choice for the communication for low-probability theft event, which lessens the economic burden of vehicle owners. In addition, false alarming should be minimized, for each alarming brings the cost of SMS. On the other hand, energy consumption should be minimized, due to the long-time running of the system. It means the whole alarming and tracking system should avoid those sensors and interfaces that have high power consumption. In addition, map data and other involved data can be previously stored in the smartphone, for avoiding extra communication and energy costs.

Of course, keeping a valid SIM card for SMS is not free, which depends on different service providers in different countries. For example, it is no more than one dollar per month in China but seems much higher in the U.S. Generally, this cost is indispensable for anti-theft systems that support remote tracking, including our system.

**3.2.2. The System Should Be Self-Adaptive.** After being installed, the smartphone will check the feasibility of GPS. In driving, the system deploys cellular localization and dead reckoning to generate a vehicle trajectory, as an energy-efficient approach. If GPS and roadside APs are possible, the trajectory can be improved by introducing more accurate landmarks from GPS or WiFi localization.

In fact, with careful installation, GPS signals can be guaranteed for our system. For example, a small antenna can be put out a hole near the trunk where the smartphone is hidden, or the smartphone can be put under the bumper where it will receive reflected GPS signals. However, vehicle tracking is not an application requiring very high accuracy. An object, as big as a car, can be easily found by people, if it is located within the sight range or 50 meters. In this case, GPS is not very important, if other energy-saving localization methods, such as cellular localization and dead reckoning, can provide a location with average errors less than 50 meters. Thus, we allow vehicle owners to hide their smartphones as they want to, not requiring any special installation to guarantee GPS.

Another possible solution is to install an OBDII bluetooth adaptor in the car. Once connected to the obsolete smartphone, such an adaptor can sense and upload real-time velocity and travelling distance for locating the car accurately. Similar to GPS, we only use an obsolete smartphone

to fulfill our task, not considering OBDII adaptor or other extra sensors.

**3.2.3. The System Should Be Convenient.** Since people always carry a smartphone, driver verification can be performed through ad hoc authentication between the driver-carrying the device and the embedded one. It means our system can open WiFi interface to monitor the WiFi signals emitting from the owner's smartphone, for checking the unique MAC address. However, many users close their smartphones' WiFi to save energy in daily using, while ad hoc authentication requires them to open WiFi before driving. If the vehicle owner forgets this operation or forgets the phone, a false alarm is triggered. Thus, single ad hoc authentication may lead to bad user experience with a high probability of false alarm.

Thus, we need a second theft detection mechanism, which does not need any operation of user but automatically recognizes unauthorized driving. Since the system can record all driving trajectories, a history-based mobility model can be established to distinguish abnormal driving behaviors from regular ones. Combining ad hoc authentication for driver verification and automatic abnormal recognition for driving verification, our system will meet the conflicting requirements for a high probability of theft detection and a low probability of false alarm.

**3.2.4. The System Should Be Memory-Efficient and Time-Efficient.** Our system does not deploy any back end server but achieves real-time tracking and alarming with the limited computational resources of a smartphone. It requires that the algorithms for theft detection and mobile tracking are simple enough and fast enough to be running at a smartphone.

**3.3. Overall Design.** According to the above principles, we can determine the necessary parts in a smartphone in our system: a GSM modem, a WiFi interface, a 3-axis accelerometer, a 3-axis gyroscope, and electronic maps including cellular base station data. As shown in Figure 2, the obsolete smartphone communicates with the vehicle owner's phone through short messages or ad hoc connections, after embedded into a vehicle. The owner can turn on or turn off the

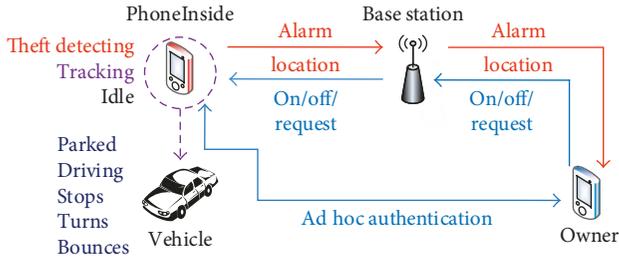


FIGURE 2: The PhoneInside system.

system or directly ask the system reporting vehicle location, if he/she finds the vehicle to be stolen.

After being installed, activated, and matched to the vehicle owner’s phone, the system enters an Idle mode. In this mode, it uses accelerometer measurements to detect whether the vehicle moves. If the readings exceed some predefined limits, for example, a driving starts, the system tries to verify the MAC address of the signals from its WiFi interface. If the MAC address of the owner’s smartphone is found, for example, ad hoc authentication is passed, it enters a Tracking mode and computes a trajectory through cellular localization and dead reckoning. If GPS signals are available, periodic records are used to enhance the trajectory. Similarly, roadside APs, if detected, are also recorded.

If the MAC address of the owner’s smartphone is not found, for example, ad hoc authentication fails, the system switches to a theft detecting mode. In this mode, the system not only computes trajectory but also uses a historical model based on previous trajectories to detect abnormal driving. If abnormal driving is recognized, the system immediately sends an alarming message to the vehicle owner. If the owner confirms vehicle theft and sends a request message, the system would periodically send location messages until it is turned off by the owner.

**3.4. Challenges and Solution.** There are several challenges that emerge when meeting the task of tracking and theft detecting, especially if the performance is to be guaranteed. From the viewpoint of the engineering application, we briefly describe our solution to guide the system design.

- (1) Orientation: as shown in Figure 1(b), the smartphone is fixed inside the vehicle body in an arbitrary orientation. It means that we cannot get correct sensor readings in vehicle axis, unless we first resolve an unconformity between smartphone axis and vehicle axis.
- (2) Sensor errors: in most practical systems, the primary error sources are related to the errors of measurement. Since current smartphone sensors are quite noisy and suffer from large drifts, it is more difficult to obtain robust and reliable estimation of vehicle movement.
- (3) Motion detection: to a vehicle, it may start to move, stop for traffic lights and traffic jam, turn at intersections and road curves, bounce at road bumps,

or park at some places. Such vehicle motions should be captured, classified, and measured by sensor readings, as accurately as possible.

- (4) Matching: with motion data, the distance and direction together produce a meaningful trajectory, which is further fitted into a map topology using a map matching algorithm. A major problem is the performance of dead reckoning, which often introduces serious errors into trajectory estimation.
- (5) Positioning: if travel trajectory has matched map data successfully, the real-time location can be estimated from the last known landmark. The challenge here is that the vehicle may break in an unknown road or a parking garage that is not included in the map data.
- (6) Modelling: it is well known that human trajectories show a high degree of temporal and spatial regularity. In this step, the goal is to find regularity from driving trajectories and to establish a proper model based on such regularity.
- (7) Theft decision: with the given mobility model, we need some rules to decide whether a driving is unauthorized, or “abnormal enough.” Such rules are often context-aware, vehicle-specific, and multifactored.

The rest of this study is meant as a step towards a deeper understanding of these fundamental issues, especially in the step of motion detection, matching, modelling, and theft decision.

## 4. System Design

For the idea of PhoneInside, we first investigate the orientation problem and the errors of motion sensors. Later, we classify typical driving events by understanding different sensor readings, explaining the method of VA-DR for accurate matching, and then giving the details of final positioning. Finally, a historical model of vehicle mobility is established, and an abnormal detection algorithm-based history model is given.

**4.1. Orientation.** Since the smartphone is fixed in the vehicle body in any orientation, its axis is massively more likely to be different with vehicle axis. As depicted in the left part of Figure 3, a smartphone has its coordinate  $(x, y, z)$  in three virtual directions. As an object, it is under the force of gravity  $g$ , no matter in what state it is. By reading the data of accelerator and comparing it to the value of gravity acceleration, we can then measure the angle  $\theta$  between the  $x$ - $y$  plane and the gravity direction.

We assume that the rotation process is at an angle  $\alpha$  around  $x$ -axis, then at an angle  $\beta$  around  $y$ -axis, and then at an angle  $\gamma$  around  $z$ -axis. Hence, the rotation matrix can be explained as follows:

$$V^* = R(\alpha)R(\beta)R(\gamma)V, \tag{1}$$

where

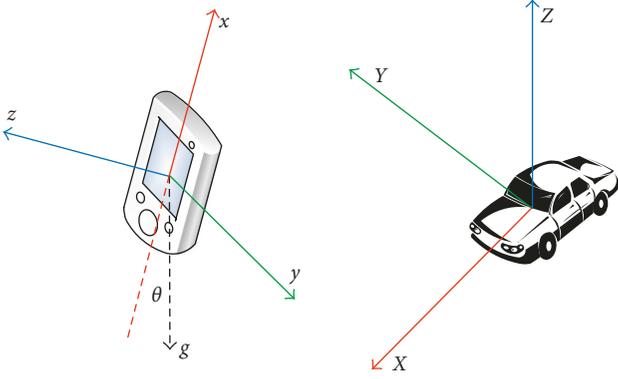


FIGURE 3: Phone axis and vehicle axis.

$$V^* = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad (2)$$

$$V = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

$$R(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix},$$

$$R(\beta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}, \quad (3)$$

$$R(\gamma) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

According to the rotation matrix, we can calculate the vehicle's axis via having three angles between the smartphone coordinate and the vehicle coordinate. That is to say, the smartphone will go through a self-learning process to complete reorientation. After reorientation, vehicle movement can be obtained by reading the data of smartphone sensors.

**4.2. Sensor Errors.** An INS is an autonomous system that provides information about position, displacement, and attitude based on the measurements by inertial sensors and applying the dead reckoning principle, which may have different designs. In many off-the-shelf mobile devices, magnetometer (electronic compass) is equipped, as well as accelerometer and gyroscope. Some studies [9, 10, 23] also use magnetometer to indicate the direction of north during driving. However, if placed in vehicle body, the magnetometer may have large dynamic errors in measurement, due to a local magnetic field from moving metal components in vehicle [24]. Thus, we use a combination of accelerometer

and gyroscope for motion detection and discuss the measurement errors of them, respectively.

The errors of MEMS sensors include those caused by constant bias, thermomechanical white noise, temperature effects, calibration errors, and bias instability. For accelerometer on the smartphone, the uncorrected bias errors and white noise are typically the primary error sources [25].

A preprocessing method [11] is proposed to correct acceleration errors for vehicle tracking. At first, the signals of accelerometer readings are passed through a second-order Butterworth Filter [26], to remove jitters and noise. Furthermore, the obtained signals are smoothed by taking a median of 20 samples and then taking the mean of 10 such obtained medians, to reduce high amplitude variation caused by high sampling rate and jerks in driving. In our system, we adopt this method to yield average acceleration in dead reckoning.

The gyroscopes generally suffer from white noise and bias. The gyroscope bias is the output when it is not experiencing any rotation. A constant bias error of  $\varepsilon$ , when integrated, causes an angular error which grows linearly with time, as  $\theta(t) = \varepsilon * t$  [27]. Thus, we can correct this error in the matching process. If some routes (parts of the whole trajectory) are matched with the map, the turns in these routes are picked up and measured. Thus, the angles measured by gyroscope and that by map knowledge of the same turns can tell us the value of  $\varepsilon$ . In later matching, this value can help us to reduce the gyroscope errors, for generating more accurate trajectory.

**4.3. Motion Detection.** The spatial movement of a rigid body can be described as a combination of translation and rotation in space. In our system, the movement of vehicle is measured by a 3-axis accelerometer and a 3-axis gyroscope, which is represented in six parameters in vehicle axis, as X-, Y-, Z-acceleration and X-, Y-, Z-angular velocity. Any vehicle motion, including movement, stops, turns, and bounces, is captured and classified, according to these six-parameter readings. Especially, velocity estimates are calculated when specific motions occur.

- (1) **Movement:** in the Idle mode, we use X-acceleration to decide whether a vehicle moves. During this mode, the accelerometer is sampled at 200 Hz for 1 second in every 5 seconds, and the peak of X-acceleration is kept in a threshold  $|T|$ . If  $|T_{i+1}| \geq 10 * |T_i|$  happens, the value of  $|T_i|$  is saved and the sampling is performed in every second. If half readings of the next 20 samples are more than  $10 * |T_i|$ , a decision of "a driving starts" is made. Then, the system carries out ad hoc authentication and switches to the Tracking mode or the Theft Detecting one. Otherwise, the sampling is restored and the system keeps in the Idle mode. As shown in Figure 4, the X-acceleration values show the changes from parking to moving. During the interval between 50 and 80 samples, the obvious positive values indicate a vehicle movement.

In the Tracking/Theft Detecting mode, the sampling rate is set at 200 Hz, and the six-parameter readings, including X-, Y-, Z-acceleration and X-, Y-, Z-angular velocity, are recorded in time order for further processing.

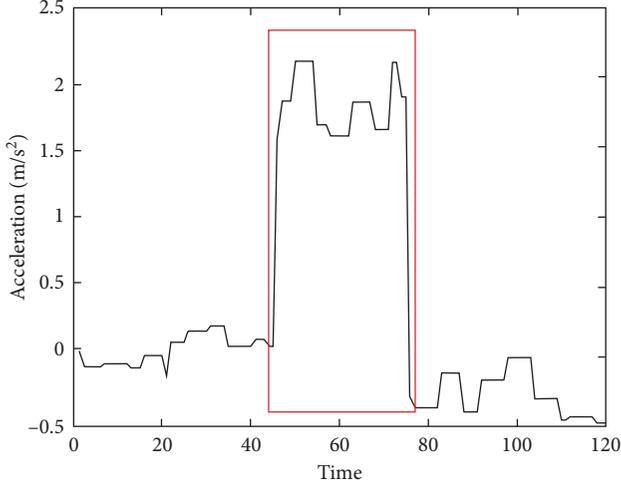


FIGURE 4: Motion detecting.

- (2) Stops: in realistic driving, a vehicle is not always moving but stops for traffic lights or traffic jam occasionally. Compared to the parking state, a stop in *driving* is often called “live parking,” for the engine is still running. In Figure 4, a stop appears between 20 and 45 samples, which is slightly different from the parking state from 0 to 20 samples. We do not distinguish the two states clearly but think a vehicle enters a stop when average acceleration and angular velocity are both zero.

Although a preprocessing method is used to correct acceleration errors (Section 4.2), the accelerometer still produces acceleration values with high oscillation. Since a vehicle may have successive stops in driving, for example, waiting for red traffic light twice, the average acceleration during the interval is zero. Thus, the mean acceleration between two successive stops can be subtracted from all recorded acceleration values, to eliminate the drift from the measurements [28].

- (3) Turns: in our system, the vehicle motion of the turn is represented by  $Y$ -angular velocity. A clockwise rotation generates positive reading, which indicates the vehicle is making a right turn; otherwise, the gyroscope generates negative reading, indicating the vehicle is making a left turn. Since a turn is given by the resultant of the two opposing forces, namely, the tangential force and the centripetal force,  $X$ -acceleration and  $Y$ -acceleration, respectively, indicate the tangential acceleration and the centripetal acceleration. Since turns can be accurately measured with a high sampling rate, an interesting question is whether we can estimate vehicle velocity at turns, with the support of parameters as angular velocity, acceleration, and passed distance.

There are two kinds of turns that frequently appear in driving: turns at intersections and turns at road curves. As shown in Figure 5, a turn at an intersection is often sharp, where  $Y$ -angular velocity has large changes in a short time interval between  $t_0$  and  $t$ . Even we have  $X$ -acceleration records as the tangential acceleration, a velocity estimate is very hard to get, because

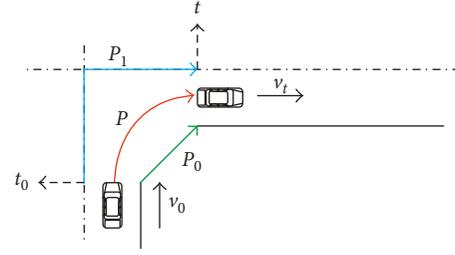


FIGURE 5: A sharp turn at intersection.

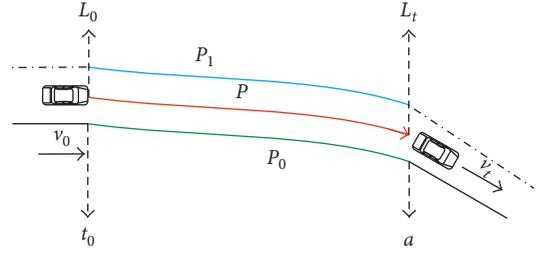


FIGURE 6: A large, gentle turn at road curve.

- (1) the trajectory may not be known deterministically, which may range from  $P_0$  to  $P_1$ ;
- (2) the distance estimate of  $P$  obtained from map data could be difficult, for it is very trivial in comparison to street length;
- (3) small errors in the estimates of the distance of  $P$  and the time points  $t_0$  and  $t$  could result in large errors in velocity estimation, for the turning is sharp and short.

**4.3.1. Curve-Based Velocity Estimation.** As shown in Figure 6, a turn at the road curve is often designed large and gentle for safety driving. In this case, the curve length  $P$  has no much difference from  $P$  and  $P_1$ . At the same time, we can get the value of  $P$  from map data, if the corresponding road curve is found in map. Finally, measurement errors in  $P$ ,  $t_0$ , and  $t$  will not affect velocity estimation very much, for the turning is slow and long.

Suppose the interval of sampling is  $\Delta t$  and  $t_0$  and  $t$  denote the beginning time point and the ending time point of a turn, then the total sampling number  $N$  is given by

$$N = \frac{t - t_0}{\Delta t}. \quad (4)$$

Therefore,  $X$ -acceleration records during the turn can be write as  $\sum_{k=1}^N a_k$ . The travelling distance  $P$  can be expressed as Newton's law of motion as

$$\begin{aligned} P &= \sum_{k=1}^N \nabla f(k) \\ &= \sum_{k=1}^N \left( V_{k-1} \Delta t + \frac{a_k \Delta t^2}{2} \right). \end{aligned} \quad (5)$$

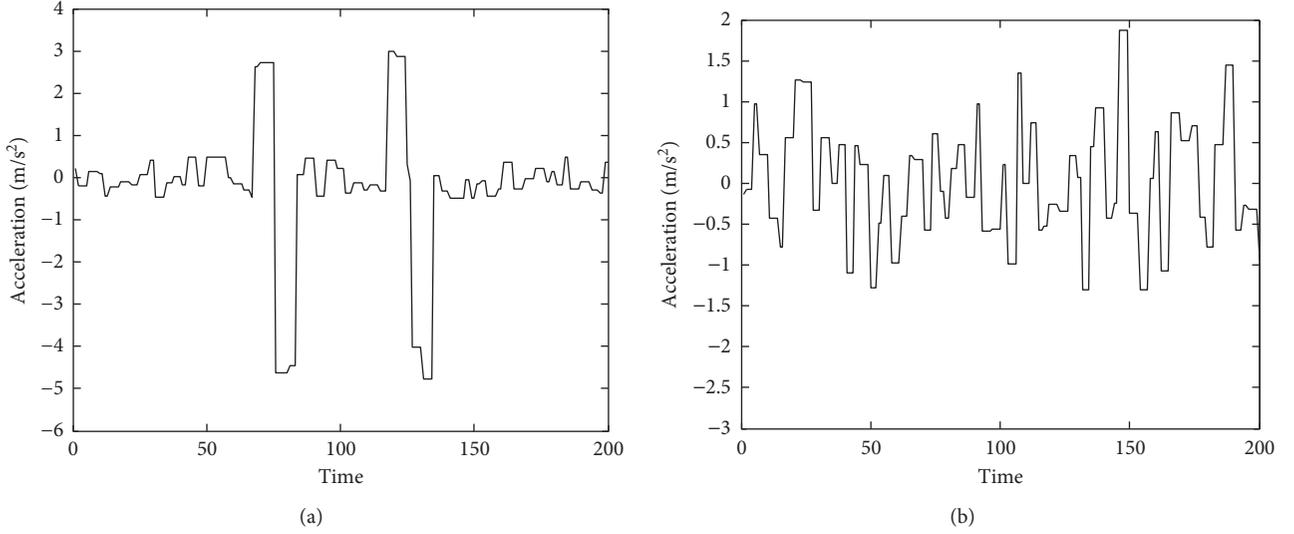


FIGURE 7: Bounces at road obstacles. (a) Good bounces. (b) Bad bounces.

With the entering velocity  $V_0$  at the time point  $t_0$ , we can write the travelling distance  $P$  as

$$\begin{aligned}
 P &= \sum_{k=1}^N \left[ \left( V_0 + \sum_{i=1}^{k-1} a_i \Delta t \right) \Delta t + \frac{a_k \Delta t^2}{2} \right] \\
 &= NV_0 \Delta t + \sum_{k=2}^N \sum_{i=1}^{k-1} a_i \Delta t^2 + \sum_{k=1}^N \frac{a_k \Delta t^2}{2}.
 \end{aligned} \tag{6}$$

Then, we have the velocity estimation function of  $V_0$  as

$$\begin{aligned}
 V_0 &= \frac{P - \sum_{k=2}^N \sum_{i=1}^{k-1} a_i \Delta t^2 - \sum_{k=1}^N (a_k \Delta t^2 / 2)}{N \Delta t} \\
 &= \frac{P}{N \Delta t} - \sum_{k=2}^N \sum_{i=1}^{k-1} \frac{a_i \Delta t}{N} - \sum_{k=1}^N \frac{a_k \Delta t}{2N}.
 \end{aligned} \tag{7}$$

Now, we only need to calculate the leaving velocity  $V$  at the time point  $t$ , and we consider

$$\begin{aligned}
 V_t &= V_0 + \sum_{k=1}^N a_k \Delta t \\
 &= \frac{P}{N \Delta t} - \sum_{k=2}^N \sum_{i=1}^{k-1} \frac{a_i \Delta t}{N} - \sum_{k=1}^N \frac{a_k \Delta t}{2N} + \sum_{k=1}^N a_k \Delta t \\
 &= \frac{P}{N \Delta t} - \sum_{k=2}^N \sum_{i=1}^{k-1} \frac{a_i \Delta t}{N} + \sum_{k=1}^N \frac{(2N-1)a_k \Delta t}{2N}.
 \end{aligned} \tag{8}$$

Using (11) and (12), we can figure out the entering and leaving velocities of the vehicle, if it happens to pass a road curve that is matched in map. With the velocity estimates at two ends of a road curve, the displacement estimation before and after the curve will be largely improved, for the accumulated errors in dead reckoning are removed.

According to our test, some turns at intersections, for example, those on bypass road or flyover, are long enough and slow enough to make velocity estimates, while some turns at sharp curves are useless. Some road curves are too plain to generate a clear beginning or ending of a turn. Sometimes, lane changing may be misleading. We simply make some rules in turn choosing for velocity estimation:

- (1) Turning should last at least 5 seconds.
- (2) Angle of turn is larger than 15 degrees.
- (3) Centripetal displacement of turn is larger than 8 meters (since the average lane width is 3.0–3.6 meters [29], most lane-changing can be removed).

**4.3.2. Two-Turn-Based Velocity Estimation.** More importantly, let us consider the vehicle travelling between two sharp turns, which is common in realistic driving. If we take the time of the first turn and that of the second one as  $t_0$  and  $t$  and the distance between two turns as  $P$ , we can compute the entering velocity  $V_0$  at the first turn as (11) and the leaving velocity  $V$  at the second turn as (12), respectively. At the same time, the limits discussed above are not effective. It means the paths between two successive turns can be used to estimate velocity, which makes all turns, no matter sharp or slow, no matter at road curves or intersections, velocity-aware. Thus, VA-DR is feasible, and the accuracy of localization is enhanced.

- (4) Bounces: during driving, a vehicle may encounter road bumps, which can cause unexpected bounces of the vehicle body. Using such bounces, also called *bumping*, to estimate driving speed is first proposed in [1]. As shown in Figure 7(a), when a vehicle is passing a speed bump, two bounces can be observed from  $Z$ -acceleration values, which indicate that the front and rear wheels get on top of the bump, respectively. The distance between front and rear

TABLE 1: The roles of vehicle motions.

Motion type	Landmark	Matched location	Accelerometer readings	Gyroscope readings
Movement	No	None	Continue	Continue
Stops	Yes	Possible	Restart	Restart
Turns at interscetions	Yes	Possible	Restart	Continue
Turns at curves	Yes	Possible	Restart	Continue
Bounces	Yes	None	Restart	Continue

wheels is called *wheelbase*, which is known to a vehicle owner. The details of velocity estimation are very similar to those of turns at road curves and intersections, which are discussed above.

However, we test this bounce-based velocity estimation method, and the results are not very good. There are bumps, flaws, potholes, manhole covers, or others on roads, but most obstacles are not straight enough and slim enough to generate good bounces, as man-made speed bumps. For example, a tilted road flaw can cause complex bounces on four wheels and two speed bumps placed near to each other will bring four bounces. Diving in bad roads also makes too many bounces. As shown in Figures 7(a) and 7(b), respectively, good bounces are two peaks of  $Z$ -acceleration, while bad ones often involve too many peaks that cannot be used to calculate velocity. In realistic driving, the chance of hitting a good bump on roads is low. Another problem is that people often slow down for speed bumps in driving, which increases errors in velocity computation. In addition, the wheelbase is a short travelling distance, so that small errors in measurement will cause large errors in estimation.

Thus, we only use this method to compute velocity at the beginning and the end of driving. At these stages, vehicles are often moving slowly in parking lots, where man-made speed bumps exist (as the case in [1]). Similarly, we set some rules in bounce choosing for velocity estimation:

- (1) Only the bounces happened during the first 100 seconds and the last 100 seconds of a driving.
- (2) Only two successive bounces are observed, and no bounce can be found in 5 seconds ahead and 5 seconds later.

**4.4. Matching.** A landmark can be any fixed location reference, such as a bridge. Its physical location is not necessary to be known, and we only take it as a unique identity for labelling the vehicle movement. Since a series of vehicle motions are classified in the last subsection, we list their roles in matching in Table 1.

As shown in Table 1, all vehicle motions are recorded in order of time, for example, known in time points. Unclassified movement cannot be matched to any location but generates distance estimation through dead reckoning. Since a vehicle may stop or turn at intersections, such motions are possible to be matched to intersections in map data. At the same time, stops are restarting points of dead reckoning, where accelerometer and gyroscope measurements can be flushed out, so

that accumulated errors are removed. In this study, new landmarks with velocity estimates, such as bounces and turns, are considered, which introduce extra restarting points in dead reckoning. Thus, a more accurate trajectory can be achieved with more velocity-aware landmarks.

Once a travel trajectory based on a sequence of displacements, turns, stops, and bounces has been obtained, the next challenge is to search a map of streets to identify the sequence of road segments that is most likely to result into the observed trajectory. The matching process is especially challenging, for many factors will affect the performance:

- (1) Not all trajectory data are meaningful in map, especially those in the early and last stages. For example, a vehicle may start from or drive into a small road not included in map.
- (2) Sensor errors and dead reckoning may result in large errors in trajectory.
- (3) The resemblance of roads is also an obstacle. Some road segments may have similar lengths or curves.

Note that, velocity estimates at road curves and intersections are impossible, unless we know the curve lengths or road lengths between two turns at first. It means that we need to have a two-step matching, which first matches road segments with the support of VA-DR, and finally constructs a path, as a sequence of road segments. Based on a basic map matching algorithm, known as the nearest segment matching (NSM) [7], we develop a VA-DR matching algorithm to support our system. Algorithm 1 shows the overall procedure of VA-DR matching, in which those road segments with curves and those between turns are first estimated and matched.

In Algorithm 1, we have a road segment set as map data. A road segment is a portion of road between two points of interest, which could be intersections, not necessarily neighboring. An error region contains a number of segments where the vehicle is travelling. Since the vehicle can be located through cellular localization, an error region may range from 100 meters to several kilometers [2, 3]. Thus, to a road segment or a trajectory, we can use the NSM to select most matched segment candidates within error region.

As shown in Algorithm 1, a greedy strategy is adopted to find the matched segment where turns at curves and intersections happen. If a turn at road curve is found, the segment candidates set  $S$  are found through calling NSM. In Lines 5, 6, and 7, each candidate  $S_i$  provides its curve length, estimates velocity as (9) and (10), then estimates the distances before and after the curve as (8), and finally forms

```

Require: NSM: select most matched segment candidates
within error region;
MotionDetect: detect vehicle motions from
sensor readings.
Ensure: A sequence of road segment.
(1) if MotionDetect finds Curve then
(2)   set a segment  $S_{cur}$ ;
(3)   call NSM for  $S_{cur}$ , get  $S = (S_1, S_2, \dots, S_n)$ ;
(4)   for  $i = 1; i \leq n; i++$  do
(5)     calculate velocity of  $S_i$  with CurveLength;
(6)     calculate distance of  $S_i$  with CurveLength;
(7)     set  $S_{cur}$  values;
(8)     if  $S_{cur}$  equals  $S_i$  then
(9)       return  $S_i$ 
(10)    end if
(11)   end for
(12) else
(13)   if MotionDetect finds Two-Turns then
(14)     set a segment  $S_{tur}$ ;
(15)     call NSM for  $S_{tur}$ ,  $S = (S_1, S_2, \dots, S_n)$ ;
(16)     for  $i = 1; i \leq n; i++$  do
(17)       calculate velocity of  $S_i$  with Distance
(18)       -BetweenTurns;
(19)       calculate distance of  $S_i$  with Distance
(20)       -BetweenTurns;
(21)       set  $S_{tur}$  values;
(22)       if  $S_{tur}$  equals  $S_i$  then
(23)         return  $S_i$ 
(24)       end if
(25)     end for
(26)   end if
(27) else
(28)   call NSM to get a most matched  $S_1$ ;
(29) end if

```

ALGORITHM 1: VA-DR matching.

a segment estimate in  $S_{cur}$ . If  $S_{cur}$  equals the candidate  $S_i$  in length, the candidate  $S_i$  can be regarded as the real segment where the vehicle passed by. Similarly, if two successive turns are encountered, the segment candidates  $S = (S_1, S_2, \dots, S_n)$  are found through calling NSM. In Lines 17, 18, and 19, each candidate  $S_i$  provides its distance between two turns, estimates velocity as (9) and (10) and then estimates the distances before the first turn and after the second turn as (8), and finally forms a segment estimate in  $S_{tur}$ . If  $S_{tur}$  equals the candidate  $S_i$  in length, the candidate  $S_i$  can be regarded as the real segment where the vehicle passed by.

Generally, the idea behind Algorithm 1 is that VA-DR can provide accurate velocity and distance estimates, so that those road segments with turns at curves and intersections should be initially estimated and matched, and those road segments without turns should be estimated and matched later. Since the incremental matching is beginning from single segment with turns, it is very efficient, for there are no global considerations. The drawback is a result of *local-best*, which means the generated path may be incomplete or incorrect. However, this approach effectively trades accuracy for speed of computation, which fits the running in the

smartphone. In addition, if cellular localization is accurate enough or the segment candidates are few, the performance of Algorithm 1 will be good, for there are no much matching errors.

**4.5. Positioning.** Positioning is the last step of vehicle tracking, which directly decides the localization performance. If travel trajectory has matched map data successfully, the real-time vehicle location can be estimated from the last known landmark. Besides the landmarks in Table 1, a GPS location is also a landmark, if GPS signals are feasible. In addition, a WiFi AP can be a landmark, if the system assures that vehicle theft happens and gets WiFi localization results from owner's phone through SMS.

If the last known landmark happens to be a GPS location, a road curve, or a turn in map, VA-DR will provide distance estimation as (12), with a known starting velocity. Otherwise, traditional dead reckoning is used to estimate the final location. Especially, bounces at the end of driving are recorded (Section 4.3.2), and bounce-based velocity estimation is used to compute vehicle trajectory. It may provide a better localization for vehicle entering an unknown road or a parking garage at a low speed.

**4.6. Modelling.** The researches on human mobility were developing rapidly in recent years. Some of them noticed the relationship between human social activities and geographic movements. The time-variant community mobility model [30] captures two properties of human mobility via empirical WLAN traces: skewed location visiting preferences and periodical reappearance of nodes at the same location. Another study [31] investigates the trajectories of 100,000 anonymous mobile phone users and finds that human trajectories show a high degree of temporal and spatial regularity: each individual can be characterized by a time-independent characteristic length scale and a significant probability to return to a few highly frequented locations.

According to NHTS [32], the majority of individual daily trips (87 percent) are taken by personal vehicle. The daily activities of an individual person, including "going to work," "having lunch," and "shopping," often show regular features. As a kind of human activity, driving is controlled by individual drivers and follows their respective social activities, partially. Although some unexpected driving occurs, individual or household driving in a certain vehicle usually yields the same spatial and temporal features. For instance, a commuter always drives his/her car from home to office at 9:00 and from office to home at 17:00. Research in the field of transportation also validates the regularities, both in human mobility and vehicle mobility. Mobidrive [33, 34] monitors the trajectories of private cars by collecting their trajectories. A mobility pattern is constituted by the spatial distribution of those locations where a traveler has had six weeks of personal experience. The spatial regularities indicate two to four main locations (such as home and office) and cover

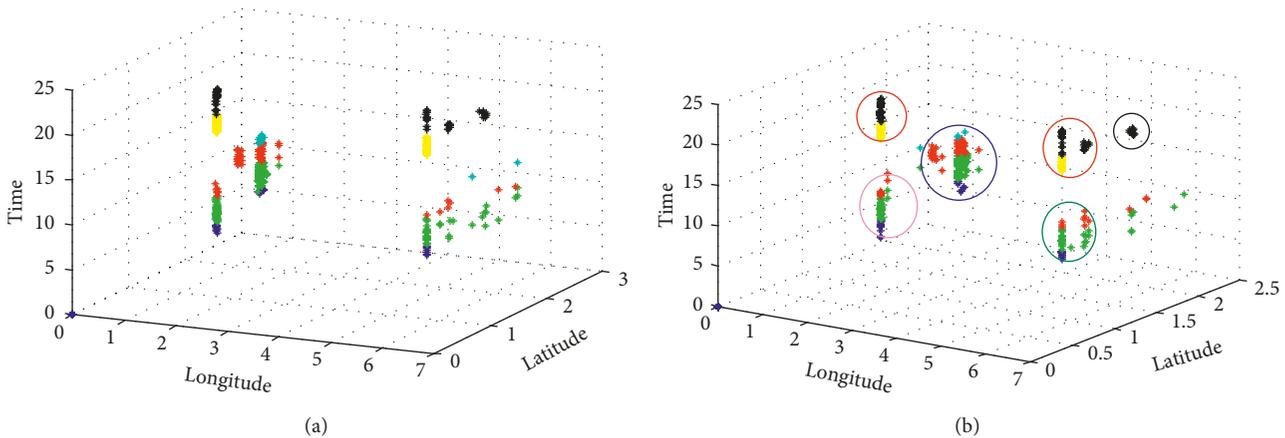


FIGURE 8: Visiting place sets. (a) Before clustering. (b) After clustering.

more than 70% of the overall trips. However, this model does not involve the temporal regularities.

In this study, we first collected realistic driving trajectories from a 16-car 3-month test (Section 5.1). Then, we can establish a new driving mobility reflecting spatial and temporal regularities. Suppose a driving history data set is  $D = \{d_i \mid i \in N\}$  and  $N$  is the capacity, a driving trajectory  $i$  can be expressed as  $d_i = \{(L_{i0}, T_{i0}), (L_{i1}, T_{i1}), \dots, (L_{in}, T_{in})\}$ , in which  $L_{ij}$  and  $T_{ij}$  are the location (including latitude and longitude) and the time of the  $j$ th measurement in the driving  $i$ , respectively.

More specifically, we use a weighted graph  $G(V, E)$  to represent one’s driving behaviors.  $V = \{v_1, v_2, \dots, v_p\}$  is the set of visiting places and  $E = \{w_{ij} = (v_i, v_j) \mid v_i, v_j \in V\}$  is the set of driving trajectories, in which  $w_{ij}$  represents all trajectories from visiting place  $v_i$  to visiting place  $v_j$ . In such a weighted graph, a vertex represents a visiting place and a line represents a driving trajectory.

**4.6.1. Visiting Places.** We first consider the starting point and the ending point of each driving and depict these points in their spatial and temporal contexts, including latitude, longitude, and time of day. The typical starting and ending point sets of a vehicle can be depicted as Figure 8(a), which represent the visiting places of a vehicle owner. Then, we use  $K$ -means clustering to process these points, which is a commonly used data clustering for performing unsupervised learning tasks. As shown in Figure 8(b), some sets of visiting places, as  $V = \{v_1, v_2, \dots, v_p\}$ , are found, by eliminating the noise data.

**4.6.2. Driving Trajectories.** With a driving history data set  $D$ , we can find  $E = \{w_{ij} = (v_i, v_j) \mid v_i, v_j \in V\}$  as one’s driving trajectories. To a record in  $D$ , we put the starting point and the ending point into the visiting place sets  $V = \{v_1, v_2, \dots, v_p\}$  and find the values of  $i$  and  $j$  in  $w_{ij}$ . Thus, we can classify the driving history data in  $D$  into the line sets  $E$  of the weighted graph  $G(V, E)$ .

At the same time, the location records in  $E$  are converted into some road IDs, through map matching. It means that repeating driving can be integrated, and the data size of  $E$  can be reduced. In addition, there are some trajectories that cannot be classified into the line set  $E$ , which can be stored for later learning. People may change their visiting places and paths and form new mobility models.

**4.7. Theft Decision.** After building a mobility model based on driving history, we need to decide under what conditions a driving behavior can be regarded *abnormal*. Since such abnormal may disobey the temporal and spatial regularities of the mobility model of the vehicle, we mainly consider some driving features with obvious temporal and spatial regularities, as the most important factors in theft decision.

Here, we discuss three factors in driving, such as current location, current driving direction, and the distance between current driving and historical driving. Correspondingly, we set three rules based on these factors, for determining whether theft happens as follows.

**4.7.1. Location Rule.** As discussed above, more than 70% of the overall driving is to two to four main visiting places, for example, a regular driving largely happens between some of the most visiting places. Furthermore, a driver is usually familiar with those roads to those places, which makes the shortest path or the most convenient path to be chosen at first. Thus, we can make a location rule as follows: if a vehicle is on the roads it most travelled in history, the chance of vehicle theft is low.

Suppose  $(x_c, y_c)$  is current location and  $ID(x_c, y_c)$  represent the ID of the road that current location belongs to. To a line set  $E = \{w_{ij} = (v_i, v_j) \mid v_i, v_j \in V\}$ , the location rule can be written as follows:

$$\text{sgn}(x_c, y_c) = \begin{cases} 1, & \text{if } ID(x_c, y_c) \in E \\ 0, & \text{if } ID(x_c, y_c) \notin E \end{cases} \quad (9)$$

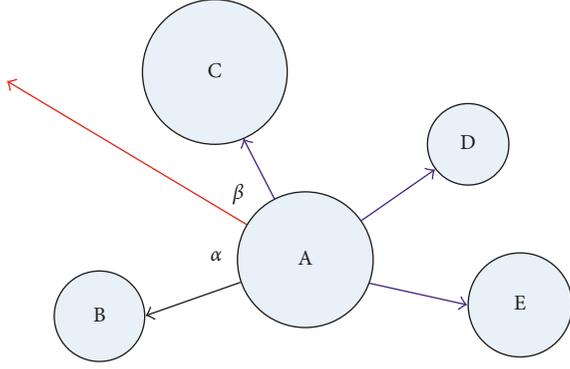


FIGURE 9: Destination rule for theft decision.

**4.7.2. Direction Rule.** Sometimes a driver will take a new path to a familiar place, due to road close, traffic jam, missing highway exit, making a wrong turn, or other reasons. Although the path is new, the travelling destination is still the most visiting place. Thus, we can make a direction rule as follows: if a vehicle is on the roads to its most visiting place in history, the chance of vehicle theft is low.

In this case, with a start location  $L_s(x_s, y_s)$  and current location  $L_c(x_c, y_c)$ , we can get a driving direction vector  $\vec{v}_{sc} = (x_c - x_s, y_c - y_s)$ . Then, we can obtain the direction vectors from the start location to the cluster centers of the visiting place sets, as  $\vec{v}_{s1}, \vec{v}_{s2}, \dots, \vec{v}_{s,s-1}, \vec{v}_{s,s+1}, \dots, \vec{v}_{s,p}$ . Each of them indicates a potential driving path to a known visiting place.

Moreover, we can find the angles between current driving direction and all possible driving directions to known visiting places. With  $i = 1, 2, \dots, s-1, s+1, \dots, p$ , the angles  $\theta_{c,s,i}$  can be written as follows:

$$\theta_{c,s,i} \langle \vec{v}_{sc}, \vec{v}_{si} \rangle = \arccos\left(\frac{\vec{v}_{sc} \cdot \vec{v}_{si}}{|\vec{v}_{sc}| \cdot |\vec{v}_{si}|}\right). \quad (10)$$

Later, we can select two minimum angles as  $\theta_{\min1}, \theta_{\min2}$  from  $\theta_{c,s,i}$ :

$$\{\theta_{\min1}, \theta_{\min2}\} = \min\{\theta_{c,s,1}, \theta_{c,s,2}, \dots, \theta_{c,s,p}\}. \quad (11)$$

Finally, the direction rule, for example, the safety probability based on driving direction, can be written as follows:

$$P_1 = 1 - \frac{\theta_{\min1}}{\theta_{\min1} + \theta_{\min2}} \quad (0.5 \leq P_1 \leq 1). \quad (12)$$

An example of the direction rule is given in Figure 9. In a 2D plane, a vehicle starts from the place A but does not take any history path. Here, we take the most probable destinations as the place B and C, because  $\beta = \theta_{\min1}$  and  $\alpha = \theta_{\min2}$  are the two minimum angles between current direction and possible directions. Then, we have  $\beta < \alpha$ , which indicates that the chance of driving to the place C is bigger.

**4.7.3. Distance Rule.** The time criterion in driving does not appear in the first two rules, but it is always effective. For

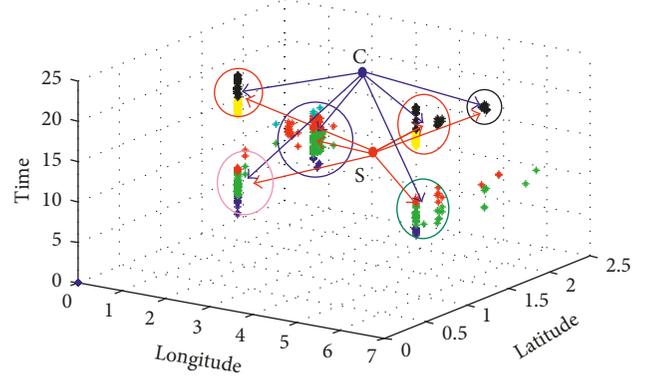


FIGURE 10: Distance rule for theft decision.

example, people often start cars in workday morning, but seldom drive at night. Thus, we can make a distance rule as follows: if a driving is near to the driving in history, the chance of vehicle theft is low.

In the 3D coordination shown in Figure 8(b), we can get the center of each cluster as  $Q_i(x_i, y_i, T_i)$ . With a start point  $Q_s(x_s, y_s, T_s)$  and current point  $Q_c(x_c, y_c, T_c)$ , we can write the distance rule, for example, the safety probability based on the distance between current driving and historical driving, as follows:

$$P_2 = 1 - \frac{\sum_{i=1}^p [d(Q_c, Q_i) - d(Q_s, Q_i)]}{\sum_{i=1}^p \max\{d(Q_c, Q_i), d(Q_s, Q_i)\}}. \quad (13)$$

An example of the distance rule is given in Figure 10. In a 3D space, a vehicle drives from the point S to the point C. Suppose all previous visiting places belong to one's life circle with a time criterion, as L. If the distance of SL is larger than that of CL, it means a vehicle is getting away from the owner's life circle. In this case, the chance of vehicle theft increases.

**4.7.4. Deep Learning-Based Theft Decision.** Combining the location rule, the direction rule, and the distance rule discussed above, we can obtain one triple feature sequence  $\vec{P} = \{(P_0^1, P_1^1, P_2^1), (P_0^2, P_1^2, P_2^2), \dots, (P_0^{n-1}, P_1^{n-1}, P_2^{n-1})\}$  from one GPS sequence  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ .  $P_0^i$  is the result of location rule form as  $P_0^i = \text{sgn}(x_i, y_i)$ . The triple feature sequence  $\vec{P}$  is less than GPS sequence with 1 step ( $\vec{P}$  length of  $n-1$ , GPS sequence length of  $n$ ), since  $P_1^i$  and  $P_2^i$  are vectors which need to be calculated from two GPS points. Moreover, we label each triple feature sequence with a sequence label. For example, we label the  $n-1$  length feature sequence  $\vec{P}$  with  $\{0, 1/n-1, \dots, n-1/n-1\}$  while it corresponds to a theft event (negative sample). It is a label smoothing regularization (LSR) method [35]. Otherwise, we label it with  $\{0, 0, \dots, 0\}$  (positive sample). The elements in each label sequence represent the probability of theft event, and we use LSR, which needs a powerful model, to fit the rule of theft event.

In this study, we choose a special recurrent neural network named LSTM, which is a powerful deep learning model and is flourish in speech recognition [36] and

machine translation [37]. The LSTM takes a sequence of  $h'$ -dimension vectors and transforms each one into  $h'$ -dimension  $v_i$ . The LSTM module can be denoted as follows:

$$\begin{aligned}
 (i_i, f_i, o_i, g_i)^T &= W_x v_i + W_h h_{i-1} + b \\
 (\bar{i}_i, \bar{f}_i, \bar{o}_i, \bar{g}_i) &= (\text{sig } m(i_i), \text{sig } m(f_i), \text{sig } m(o_i), \tan h(g_i)) \\
 c_i &= \bar{f}_i \odot c_{i-1} + \bar{i}_i \odot \bar{g}_i \\
 h_i &= \bar{o}_i \odot \tan h(c_i).
 \end{aligned} \tag{14}$$

We use only one layer LSTM network with hidden size 256, since our dataset is limited. The detail experiment settings and results are shown in Section 5.2.

## 5. Performance Evaluation

In this section, after a brief introduction of testbed, we present the evaluations of various aspects of the PhoneInside system, including distance estimation, map matching, positioning accuracy, theft detection, and system cost.

**5.1. Testbed.** To evaluate the PhoneInside system, we need to test our designs in realistic driving. It is not easy to build suitable testbed to finish experiment and comparison. We found 16 volunteers, most of them are school employees, to attend our experiments. Each volunteer has a private car. We prepared 16 Android smartphones as anti-theft device, including Google Nexus 5, Galaxy S3, and Huawei 3C. Each user embedded one phone in his/her vehicle body and matched it to his/her private phone. In addition, all users were asked to run our application in their private phones during driving, which also collects GPS traces as reference trajectories.

After a 3-month test, we take back data from the embedded phones and the owner's phone and evaluate the performance of the PhoneInside system. A total trace of 4833 trips are found, which involves 48,900 km travelling. The average driving to each volunteer is about 3.36 times per day, with a distance of 34 km one day.

In order to train an efficient neural network, sufficient training data are necessary. Among the volunteers, 10 of them are picked up to simulate vehicle theft. Each volunteer drives on each vehicle with three times, one for normal driving, one for abnormal driving, and another one for normal driving on unusual paths. For the collected trips, we use different sample steps on the GPS trajectories to make data enhancement, which can enlarge the dataset by a factor of 100. At last, we label the 3000 trajectories/GPS sequences with LSR (which is discussed in Section 4.7) for further processing.

### 5.2. Performance

**5.2.1. Dead Reckoning.** As shown in Table 1, there are two kinds of landmarks in traditional dead reckoning, which are stops and turns at intersections. Now we mainly use three

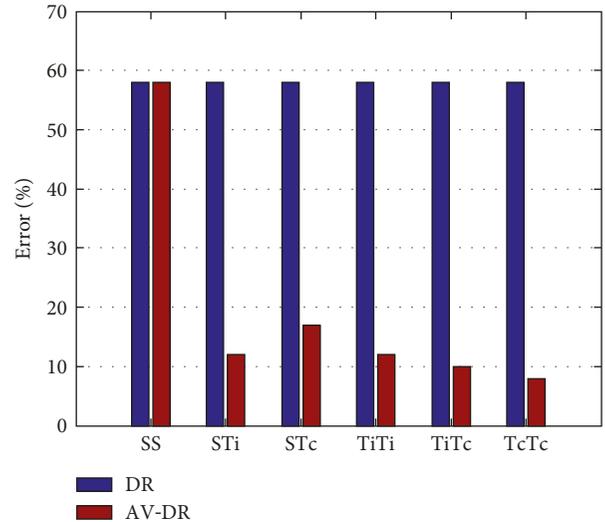


FIGURE 11: Error in distance estimation.

kinds of landmarks in VA-DR, as stops, turns at intersections, and turns at curves (that are marked as S, Ti, and Tc in Figure 11). Bounces are also velocity-aware landmarks but seldom appear in trajectory computation. Since distance estimation between two landmarks is the fundamental step of localization, we can compare the average errors of distance estimation between different landmarks. The GPS traces between these landmarks, which obtained from the owner's phone, are used as accurate distance values.

As shown in Figure 11, the average error in dead reckoning is about 58%. VA-DR greatly improves the estimation that related to velocity-aware landmarks as Ti and Tc, which is not surprising for VA-DR is designed to do that.

**5.2.2. Map Matching.** We mainly consider two factors affecting matching, for example, base station and distance. Since the vehicle is first located through cellular localization, an error region in matching is roughly decided by the number of base station. According to the studies in cellular localization [2, 3], the coverage of base station plays a vital role in localization. If the coverage is small, there are dense tower deployment that represents a small error region in map matching. On the contrary, a larger coverage means fewer cell towers can be used for localization. As shown in Figure 12(a), the coverage is ranging from 500 to 3500 m, which represents an accuracy from 100 m to nearly 1300 m [2]. We test NSM, VA-DR matching, and HMM matching (Section 2.1) and find all error segments in matching process, with the rising of base station coverage. Compared NSM and VA-DR matching, HMM matching show better performance, in which the global knowledge of trajectory is considered. In incremental matching schemes, VA-DR matching is better than NSM, due to more accurate trajectory based on VA-DR.

In Figure 12(b), the impact of travel distance is shown. With an increase of travel distance, HMM matching shows better with longer travel distance and more knowledge. NSM does not show this trend, in which incremental matching

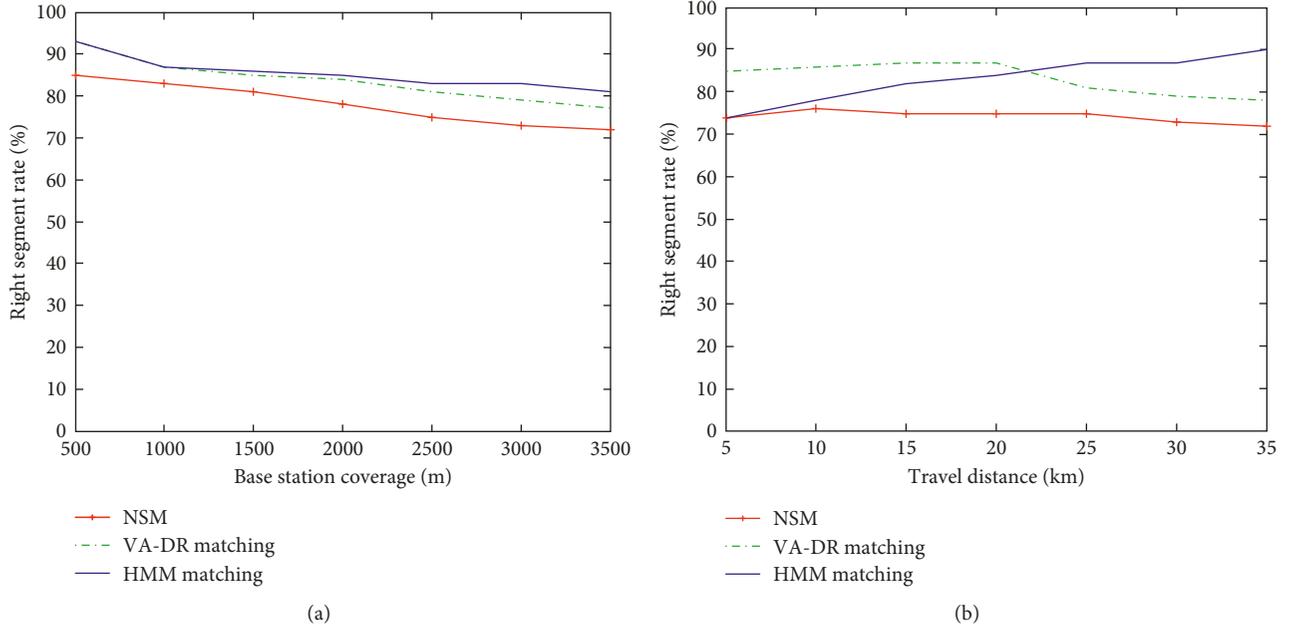


FIGURE 12: Map matching. (a) Base station factor. (b) Distance factor.

TABLE 2: Accuracy of different localization methods.

Scheme	Mean (m)	STD (m)
Cellular + DR	72.4	55.9
Cellular + HMM	41.7	31.2
Cellular + VA-DR	46.8	37.6
GPS + VA-DR	14.7	28.4

cannot fix previous errors in travelling. VA-DR matching rises first and falls later, when a vehicle travels further. It proves that the problem of *local best* exists if there is no global knowledge. However, VA-DR matching has a highest hitting rate in the matching with short distance, in which local characteristics, as accurate trajectory based on VA-DR, play an important role in incremental matching. Thus, an accurate and fast localization is feasible in the early stage of vehicle tracking.

**5.2.3. Positioning Accuracy.** Table 2 summarizes the statistics of positioning accuracy in our test, which provide the final performance for vehicle recovering. The basic localization scheme of our system is cellular localization and VA-DR, with an accuracy of 46.8 m. For those system that can receive GPS signals, the accuracy increases to 14.7 m though GPS and VA-DR. If there is no VA-DR, cellular localization and traditional dead reckoning achieve an accuracy of 72.4 m. If there is a back server to support online HMM matching, the accuracy will be up to 41.7 m.

As discussed in Section 3, a vehicle can be easily found by people within 50 meters. Using cellular localization and VA-DR is an energy-efficient solution to achieve this goal. Moreover, it is also a cost-efficient scheme to achieve this

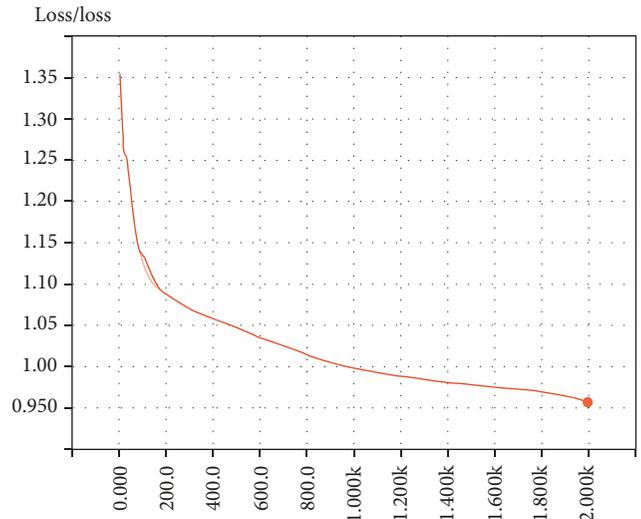


FIGURE 13: The training procedure of the LSTM model.

goal, for a HMM matching process requires the support of back server and 3G/4G connections (Section 2). Generally, the scheme of cellular localization and VA-DR fits our system well and may be useful to many car navigation systems as an auxiliary localization method.

#### 5.2.4. Theft Detection

- (a) Experiment settings: we use one layer LSTM network with 256 hidden units to build our deep learning model. The weights are initialized with truncated normal distribution of 0 mean and 0.1 standard deviation. We choose RMSProp as training optimizer with a learning rate of 0.002 and use cross entropy

TABLE 3: Test of vehicle theft.

	Alarm	No alarm	Average delay time
Normal	0.068	0.93	0.34 (h)
Abnormal	0.932	0.07	0.3 (h)

with the sigmoid function. The implementation of the model uses a public deep learning library TensorFlow and is trained on a server with 4 GTX1080 GPUs and 8 GB memory. The batch size is set to 8, and a valid training have 3750 iterations, which costs about 10 minutes. After that, the LSTM model is deployed on the smartphone to test.

- (b) Training: based on the simulation dataset of theft detection, Figure 13 shows the training procedure of the LSTM model. The loss is stably decreasing with the increasing of iteration number. We also add the dropout method, and the output keeps a probability of 0.8. The model is converged since the loss is less than 1.00.
- (c) Test: the trained model is tested on the simulated vehicle theft dataset discussed in the last subsection, and the test results are shown in Table 3. In Table 3, we can find that the true positive rate is about 0.93, while the false positive rate is 0.068. As a second theft detection mechanism, our system could achieve a very low false positive rate, with the help of phone-based ad hoc authentication. The average delay time of alarm is also low, which indicates that our system could send out alarm in time and track a stolen vehicle quickly.

**5.2.5. System Cost.** In our system, the obsolete smartphone is installed with a recharger, which obtains power supply during driving, and relies on its own battery in parking time. With the help of Android system monitor, we record the average power consumption of PhoneInside app running in different states and find the following results in Table 4.

In the Idle state, the car is parked and the system detects possible vehicle moving with the accelerometer and gyroscope. In this case, the energy consumption is about 343.9 mW, and the system keeps running about 68.2 hours in average. But it does not mean that our system will be out of function after three days. As soon as the car is driving, the system will keep tracking after a restarting period. In the Tracking and Theft Detecting modes, the energy consumption rises to 485.5 mW and 672.5 mW, respectively. There is no energy limit because the car is in driving and the recharger is working. According to NHTS [32], the average driving time is about 55 minutes one day. The power consumption of our system in daily use is about 2297.48 mAh one day, which is very near to a smartphone in standby mode.

Another cost lies in the SMS messages as communication cost in Theft Detecting mode. At a frequency of one message every 3 minutes, the system will send out 40 messages in a two-hour driving. After the vehicle is parked, the system does not send messages, and the last SMS indicates the position of the vehicle. Since vehicle theft is an event with a very low probability, sending some messages is acceptable to most vehicle owners.

TABLE 4: Power consumption in different states.

State	Power (mW)	Lasting time (h)
Idle	343.9	68.2
Tracking	485.5	No limit
Theft detecting	672.5	No limit

Generally, we can draw a conclusion that our scheme is low cost in daily use and longtime tracking, with very low energy cost and communication cost.

## 6. Conclusion and Perspective

Motivated by substantial losses of vehicle theft, we propose PhoneInside to make the best of obsolete smartphones for vehicle safety. The basic idea of PhoneInside is simple: if we have to discard old phones, why not keep them to protect our cars as free anti-theft devices?

In the paper, we first explain hardware installation and describe the system overview. Then, we consider the system design step by step and investigate the challenges in it. Using a novel dead-reckoning method, we achieve an energy-efficient tracking with enough accuracy. At the same time, we use individual driving history to distinguish abnormal driving and construct a convenient and effective anti-theft mechanism, including ad hoc authentication and deep learning-based driving abnormal detection. At last, the experiment results show that our system can detect vehicle theft effectively and locate a stolen vehicle accurately.

Since some implementation issues have not been carefully considered, we list them as future work:

- (1) The algorithm of VA-DR matching
- (2) The rules to distinguish abnormal driving
- (3) More training data for LSTM
- (4) More smartphone OS supports.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] G. Tan, M. Lu, F. Jiang, K. Chen, X. Huang, and J. Wu, "Bumping: a bump-aided inertial navigation method for indoor vehicles using smartphones," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1670–1680, 2013.
- [2] L. Xiong, "A selective model to suppress NLOS signals in angle-of-arrival (AOA) location estimation," in *Proceedings of the Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 1998*, vol. 1, pp. 461–465, Boston, MA, USA, September 1998.
- [3] L. Cong and W. Zhuang, "Non-line-of-sight error mitigation in tdoa mobile location," in *Proceedings of the Global Telecommunications Conference (GLOBECOM'01)*, vol. 1, pp. 680–684, San Antonio, TX, USA, November 2001.
- [4] R. Toledo-Moreo, D. Bétaille, and F. Peyret, "Lane-level integrity provision for navigation and map matching with gnss, dead reckoning, and enhanced maps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 100–112, 2010.

- [5] E. D. Martí, D. Martín, J. García, A. de la Escalera, J. M. Molina, and J. M. Armingol, "Context-aided sensor fusion for enhanced urban navigation," *Sensors*, vol. 12, no. 12, pp. 16802–16837, 2012.
- [6] K. Yu and E. Dutkiewicz, "Geometry and motion-based positioning algorithms for mobile tracking in nlos environments," *IEEE Transactions on Mobile Computing*, vol. 11, no. 2, pp. 254–263, 2012.
- [7] A. Thiagarajan, L. Ravindranath, K. LaCurts et al., "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 85–98, Berkeley, CA, USA, November 2009.
- [8] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, "Co-operative transit tracking using smart-phones," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pp. 85–98, Zurich, Switzerland, November 2010.
- [9] A. Thiagarajan, L. Ravindranath, H. Balakrishnan et al., "Accurate, low-energy trajectory mapping for mobile devices," in *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, p. 20, Boston, MA, USA, March–April 2011.
- [10] H. Wang, Z. Wang, G. Shen, F. Li, S. Han, and F. Zhao, "Wheelloc: enabling continuous location service on mobile phone for outdoor scenarios," in *Proceedings of the INFOCOM, 2013*, pp. 2733–2741, Turin, Italy, April 2013.
- [11] S. Guha, K. Plarre, D. Lissner et al., "Autowitness: locating and tracking stolen property while tolerating gps and radio outages," *ACM Transactions on Sensor Networks*, vol. 8, no. 4, p. 31, 2012.
- [12] V. Kukshya, H. Krishnan, and C. Kellum, "Design of a system solution for relative positioning of vehicles using vehicle-to-vehicle radio communications during gps outages," in *Proceedings of the Vehicular Technology Conference (VTC'05)*, vol. 2, pp. 1313–1317, Stockholm, Sweden, May–June 2005.
- [13] R. Parker and S. Valaee, "Vehicle localization in vehicular networks," in *Proceedings of the Vehicular Technology Conference (VTC'06)*, pp. 1–5, Melbourne, Australia, May 2006.
- [14] H. Gong, L. Yu, and X. Zhang, "Social contribution-based routing protocol for vehicular network with selfish nodes," *International Journal of Distributed Sensor Networks*, vol. 10, no. 4, p. 753024, 2014.
- [15] H. Guo, H. Cheng, Y. Wu et al., "An automotive security system for anti-theft," in *Proceedings of the Eighth International Conference on Networks (ICN'09)*, pp. 421–426, Cancun, Mexico, March 2009.
- [16] V. K. Sadagopan, U. Rajendran, and A. J. Francis, "Anti theft control system design using embedded system," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES'11)*, pp. 1–5, Beijing, China, July 2011.
- [17] O. Hongzhi, W. Xinlin, Z. Weihua, and L. Yuehua, "Design of auto-guard system based on rfid and network," in *Proceedings of the International Conference on Electric Information and Control Engineering (ICEICE'2011)*, pp. 1292–1295, Wuhan, China, April 2011.
- [18] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC'2011)*, pp. 1609–1615, Hamburg, Germany, September 2011.
- [19] M. Khalid, A. Wahab, and N. Kamaruddin, "Real time driving data collection and driver verification using CMAC-MFCC," in *Proceedings of the 2008 International Conference on Artificial Intelligence (ICAI'08)*, pp. 219–224, Las Vegas, NV, USA, July 2008.
- [20] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *Proceedings of the Intelligent Vehicles Symposium (IV'2012)*, pp. 234–239, Alcal de Henares, Madrid, Spain, June 2012.
- [21] H. Song, S. Zhu, and G. Cao, "SVATS: a sensor-network-based vehicle anti-theft system," in *Proceedings of the 27th Conference on Computer Communications IEEE INFOCOM 2008*, pp. 2128–2136, Phoenix, AZ, USA, April 2008.
- [22] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: a new vanet-based smart parking scheme for large parking lots," in *Proceedings of the IEEE INFOCOM 2009*, pp. 1413–1421, Rio de Janeiro, Brazil, April 2009.
- [23] R. Bhoraskar, N. Vankadhara, B. Raman, and P. Kulkarni, "Wolverine: traffic and road condition estimation using smartphone sensors," in *Proceedings of the Fourth International Conference on Communication Systems and Networks (COMSNETS'12)*, pp. 1–6, Bangalore, India, January 2012.
- [24] E. Abbott and D. Powell, "Land-vehicle navigation using GPS," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 145–162, 1999.
- [25] O. J. Woodman, "An introduction to inertial navigation," Tech. Rep. UCAMCL-TR-696, Computer Laboratory, University of Cambridge, Cambridge, UK, 2007.
- [26] A. Lawrence, *Modern Inertial Technology: Navigation, Guidance, and Control*, Springer, Berlin, Germany, 1998.
- [27] V. Marotto, M. Sole, T. Dessi, A. Manchinu, D. Carboni, and A. Serra, "Orientation analysis through a gyroscope sensor for indoor navigation systems," in *Proceedings of the Fourth International Conference on Sensor Device Technologies and Applications*, pp. 85–90, Barcelona, Spain, August 2013.
- [28] D. M. Boore, "Analog-to-digital conversion as a source of drifts in displacements derived from digital recordings of ground acceleration," *Bulletin of the Seismological Society of America*, vol. 93, no. 5, pp. 2017–2024, 2003.
- [29] Lane width, chapter 3: The 13 controlling criteria," 2013, [http://safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/chapter3/3\\_lanewidth.htm](http://safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/chapter3/3_lanewidth.htm).
- [30] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications IEEE (INFOCOM'2007)*, pp. 758–766, Anchorage, AK, USA, May 2007.
- [31] M. González, C. Hidalgo, and A. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.
- [32] National household travel survey," 2001, [http://www.bts.gov/programs/national\\_household\\_travel\\_survey](http://www.bts.gov/programs/national_household_travel_survey).
- [33] S. Schönfelder and S. Raumplanung, "Some notes on space, location and travel behaviour," in *Proceedings of the 1st Swiss Transport Research Conference*, Ascona, Switzerland, March 2001.
- [34] S. Schönfelder and U. Samaga, "Where do you want to go today?—more observations on daily mobility," in *Proceedings of the 3rd Swiss Transport Research Conference*, Ascona, Switzerland, July 2003.
- [35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 2818–2826, Las Vegas, NV, USA, June–July 2016.
- [36] D. Amodei, R. Anubhai, E. Battenberg et al., "Deep speech 2: end-to-end speech recognition in english and mandarin," *Computer Science*, 2015.
- [37] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *Computer Science*, 2015.

