# SenSpeed: Sensing Driving Conditions to Estimate Vehicle Speed in Urban Environments

Jiadi Yu, *Member, IEEE*, Hongzi Zhu, *Member, IEEE*, Haofu Han, Yingying (Jennifer) Chen, *Senior Member, IEEE*, Jie Yang, *Member, IEEE*, Yanmin Zhu, *Member, IEEE*, Zhongyang Chen, Guangtao Xue, *Member, IEEE*, and Minglu Li

**Abstract**—Acquiring instant vehicle speed is desirable and a corner stone to many important vehicular applications. This paper utilizes smartphone sensors to estimate the vehicle speed, especially when GPS is unavailable or inaccurate in urban environments. In particular, we estimate the vehicle speed by integrating the accelerometer's readings over time and find the acceleration errors can lead to large deviations between the estimated speed and the real one. Further analysis shows that the changes of acceleration errors are very small over time which can be corrected at some points, called *reference points*, where the true vehicle speed can be estimated. Recognizing this observation, we propose an accurate vehicle speed estimation system, SenSpeed, which senses natural driving conditions in urban environments including *making turns*, *stopping*, and *passing through uneven road surfaces*, to derive reference points and further eliminates the speed estimation deviations caused by acceleration errors. Extensive experiments demonstrate that SenSpeed is accurate and robust in real driving environments. On average, the real-time speed estimation error on local road is $2.1\,\mathrm{km/h}$, and the offline speed estimation error is as low as $1.21$ km/h. Whereas the average error of GPS is $5.0$ and $4.5$ km/h, respectively.

**Index Terms**—Sensing, driving conditions, vehicle speed, urban environments

## 1 Introduction

The smartphone-based vehicular applications become more and more popular to analyze the increasingly complex urban traffic flows and facilitate more intelligent driving experiences including vehicle localization [1], [2], enhancing driving safety [3], [4], driving behavior analysis [5], [6] and building intelligent transportation systems [7], [8]. Among these applications, the vehicle speed is an essential input. Accurate vehicle speed estimation could make those vehicle-speed dependent applications more reliable under complex traffic systems in urban environments.

Generally, the speed of a vehicle can be obtained from GPS. However, GPS embedded in smartphones often suffers from the urban canyon environment [9], which could result in low availability and accuracy. In addition, the low update rate of GPS is not able to keep up with the frequent change of the vehicle speed in urban driving environments. Moreover, continuously using GPS drains the phone battery quickly. Thus, it is hard to obtain accurate vehicle speed relying on GPS for applications requiring real-time or high-accuracy speed estimations. Besides vehicle speed estimation based on GPS, there are a couple of alternatives by using either the OBD-II interface [3] or smartphone's cell tower signals [10] [11]. Although the speed obtained from OBD-II is quite accurate, this approach relies on an additional OBD-II adapter. Using cell tower signal changes on smartphones to perform vehicle speed tracking, [10], [11] show a promising direction that the smartphone on the vehicle can be employed to facilitate vehicle speed estimation. However, the existing studies utilizing Derivative Dynamic Time Warping (DDTW) algorithm introduces large overhead on collecting offline trace and prevents large-scale deployment. Also, the speed estimation accuracy of DDTW suffers from the coarse-grained signal information. In this paper we consider a sensing approach, which uses smartphone sensors to sense natural driving conditions, to derive the vehicle speed without requiring any additional hardware. The basic idea is to obtain the vehicle's speed estimation by integrating the phone's accelerometer readings along the vehicle's moving direction over time. While the idea of integrating the acceleration values over time seems simple, a number of challenges arise in practice. First, the accelerometer readings are noisy and affected by various driving environments. Second, the speed estimation should be real-time and accurate. Finally, the solution should be lightweight and computational feasible on smartphones.

We first show the vehicle speed estimation using the integral of accelerometer's readings through real road driving experiments in two different cities. We find that directly performing integration over acceleration results in large deviations from the true speed of the vehicle. The interesting observation is that the error between the integral value and true speed increases almost linearly over time, and is independent of different phone types. This indicates that the changes of the acceleration error are very small over

- *J. Yu, H. Zhu, H. Han, Y. Zhu, Z.Chen, G. Xue, and M. Li are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {jiadiyu, hongzi, hanhaofu, yzhu, wodoo2474, gt_xue, mlli}@sjtu.edu.cn.*
- *Y. Chen is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030. E-mail: yingying.chen@stevens.edu.*
- *J. Yang is with the Department of Computer Science, Florida State University, Tallahassee, FL 32306. E-mail: jie.yang@cs.fsu.edu.*

*Manuscript received 11 July 2014; revised 19 Jan. 2015; accepted 23 Feb. 2015. Date of publication 9 Mar. 2015; date of current version 1 Dec. 2015.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TMC.2015.2411270*

time which can be corrected if we can derive the speed errors at some time points. Based on this simple yet useful finding, we develop a vehicle speed estimation system, SenSpeed, which utilizes smartphone sensors (accelerometer and gyroscope) to sense the practical driving conditions, which can be exploited to eliminate the acceleration errors and estimate vehicle speed accurately.

In particular, our system, SenSpeed, identifies unique *reference points* from the natural driving conditions to infer the vehicle's speed at each reference point grounded on different features presented by these reference points. Such reference points include making turns, stopping (at a traffic light or stop sign or due to road traffic) and passing through uneven road surfaces (e.g., speed bumps or potholes). Based on the speed inferred from the reference points, SenSpeed measures the acceleration error between each two adjacent reference points and eliminates such errors to achieve high-accuracy speed estimation. The main advantage of SenSpeed is that it senses the unique features in natural driving conditions through simple smartphone sensors to facilitate vehicle speed estimation. Furthermore, SenSpeed is easy to implement and computational feasible on standard smartphone platforms. Our extensive experiments in both Shanghai, China and New York City, USA validate the accuracy and the feasibility of using our system in real driving environments.

We highlight our main contributions as follows:

- We propose to perform accurate vehicle speed estimation by sensing natural driving conditions using smartphone sensors. We study the impact of the acceleration error on the speed estimation results obtained from the integral of the phone's accelerometer readings.
- We exploit three kinds of reference points sensed from natural driving scenarios to infer the vehicle speed at each reference point, which could be utilized to reduce the acceleration error that affect the accuracy of vehicle speed estimation.
- We develop a vehicle speed estimation system, SenSpeed, which utilizes the information obtained from the reference points to measure and eliminate the acceleration error and achieves high accuracy speed estimation.
- We conduct extensive experiments in two cities, Shanghai, China and Manhattan in New York City, USA. The results show that, in representative urban environments, SenSpeed can estimate the vehicle speed in real-time with an average error of 2.12 km/h, while achieving 1.21 km/h during the offline estimation.

The rest of the paper is organized as follows: The related work is reviewed in Section 2. We describe basic idea in Section 3. Section 4 presents the design details of our speed estimation system, SenSpeed. Section 5 presents energy optimization of SenSpeed. We evaluate the performance of our system and present the results in Section 6. Finally, we give conclusive remarks in Section 7.

## 2 Related Work

In this section, we review the existing work on vehicle speed estimation, which can be categorized as follows.

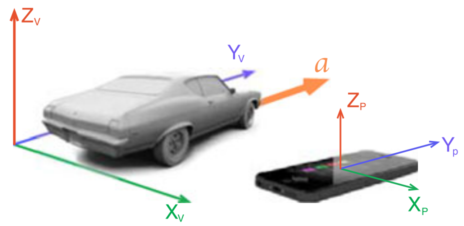



Fig. 1. Illustration of the vehicle's coordinate system and the smartphone's coordinate system.

*Estimation using pre-deployed infrastructures*. In the existing work, there are two vehicle speed estimation mechanisms deployed on highways or main roads. One is employing the loop detectors [12], [13], and the other is using traffic cameras [8]. These solutions all rely on pre-deployed infrastructures that incur installation cost. The traffic camera could be installed in urban environments, but it suffers low accuracy, bad weather conditions and high maintenance cost.

*Estimation using additional devices*. OBD-II adapter [3] is a popular interface to provide the vehicle speed in real-time. Acoustic wave sensors [14], [15] are utilized to estimate the vehicle speed in open environments. Furthermore, traffic magnetic sensors are also employed to capture the vehicle speed [16]. These approaches need to install additional hardware to perform speed estimation.

*Estimation using phones*. To eliminate the need of pre-deployed infrastructures and additional hardware, recent studies concentrate on using cell phones to measure the vehicle speed. In particular, [17], [18] use GPS or sub-sampled GPS to drive the vehicle speed. Although GPS is a simple way to obtain vehicle speed, the urban canyon environment and the low update frequency of GPS make it difficult to accurately capture the frequent changing vehicle speed in urban environments. And continuously using GPS causes quicker battery drainage on smartphones. Knowing the drawbacks of using GPS, [10], [11] estimate the vehicle speed by warping mobile phone signal strengths and [19], [20] use the handovers between base stations to measure the vehicle speed. These solutions need to build a signal database which may incur high labor cost and cannot achieve high estimation accuracy.

Obtaining the vehicle speed becomes more and more important in supporting large amounts of vehicular applications. Our work is different from the previous studies in that we explore a smartphone-enabled sensing approach based on natural driving conditions without the need of GPS or additional hardware.

## 3 Basic Idea

We first describe how to obtain the vehicle speed from smartphone sensors. The vehicle's acceleration can be obtained from the accelerometer sensor in the smartphone when a phone is aligned with the vehicle. Suppose the accelerometer's $y$-axis is along the moving direction of the vehicle as shown in Fig. 1. We could then monitor the vehicle acceleration by retrieving readings from the accelerometer's $y$-axis. The vehicle speed can then be calculated from the integral of the acceleration data over time:

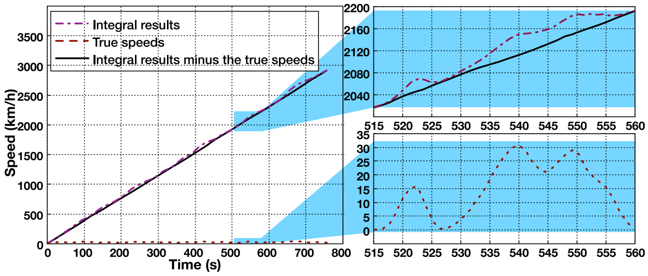



Fig. 2. The true speed, integral value of the accelerometer's readings and their difference in a real driving environment.

$$Speed(T) = S\ peed(0) + \int_0^T acc(t)\,\mathrm{d}t, \tag{1}$$

where $Speed(T)$ is the vehicle speed at time $T$ and $acc(t)$ is the vehicle acceleration function of each time instant $t$.

Instead of producing a continuous function $acc(t)$, the accelerometer in practise takes a series of the vehicle acceleration samples at a certain sampling rate. Thus the vehicle speed can be transformed as

$$Speed(T) = Speed(0) + \sum_{i=0}^{T*k} \frac{1}{k} acc_y(i), \tag{2}$$

where $k$ is the sample rate of the accelerometer and $acc_y(i)$ is the $i$th sample, i.e., the $i$th received reading from the accelerometer's $y$-axis. Therefore, in order to obtain the vehicle speed, we take a series of the acceleration samples by monitoring the accelerometer continuously.

Although the basic idea of using smartphone sensors to estimate vehicle speed is simple, it is challenging to achieve high-accuracy speed estimations. The most obvious problem is that the noise from sensor readings cause serious errors in the estimation results. Such sensor readings are affected by various noise encountered while driving such as engine vibrations, white noise, etc. And the estimation errors are accumulated when integrating the accelerometer's readings over time.

To study the impact of the accumulative error on the speed estimation's accuracy, we conduct experiments about 1,200 kilometers driving at different urban regions with three different smartphones (Galaxy Nexus by Samsung, Nexus4 by LG and iPhone4s by Apple) for over two weeks. Fig. 2 shows the results of a 12 minutes driving that compare the integral value of readings from the accelerometer's $y$-axis with the true vehicle speed collected from an OBD-II adapter. It can be seen from Fig. 2 that the integral results (i.e., the purple curve) grows rapidly over time. This is because the accumulative errors cause large deviations between the speed estimation from the integral value and the true speed. Therefore, in order to estimate the vehicle speed accurately, the accumulative error must be eliminated.

One important observation is that the black curve of the difference between the integral value from Eq. (2) and the true speed increases almost linearly over time, which indicates that the changes over time of the acceleration error are very small. These results are consistent during our experiments at different urban regions with three different smartphones. Thus, if we can derive techniques to measure the acceleration error, the integral value of the accelerometer's readings can be corrected to get close to the true vehicle speed. Since the difference curve between the integral value and the true speed is an approximate linear function of time, the acceleration error is strongly related to the slope of the curve. If we can obtain the true speeds at two time points along the difference curve, the slope of the curve could then be calculated and the acceleration error could be derived accordingly. However, the difference curve is not exactly linear, and slight changes of the slope (i.e., the acceleration error) would affect the accuracy of the speed estimation. To sense the slight changes over time of the acceleration errors, we should capture as many as possible time points, called *reference points*, where the true speed is known, then calculate acceleration errors between each two adjacent points. After knowing these acceleration errors, the integral values can be corrected to get closer to the true speeds.

## 4 Design of SenSpeed

In this section, we present the design of our proposed system, SenSpeed, which estimates vehicle speed accurately through sensing driving conditions in urban environments. SenSpeed does not depend on any pre-deployed infrastructure and additional hardware.

### 4.1 System Overview

The vehicle speed can be estimated by integrating of acceleration data over time. However, the accumulative error from the biased accelerations causes large deviations between the true speed and the estimated speed. In order to realize an accurate vehicle speed estimation, SenSpeed senses the natural driving conditions to identify the reference points, then uses the information of the reference points to measure the acceleration error and further eliminates accumulative error.

Our system identifies three kinds of references points, *making turns*, *stopping*, and *passing through uneven road surfaces*, by sensing natural driving conditions based on smartphone sensors. 1) *making turns*. A vehicle usually undergoes plenty of turns in urban environments. The vehicle speed can be inferred according to a principle of the circular movement when a vehicle makes a turn. 2) *stopping*. A vehicle stops frequently in urban environments because of stop signs, red traffic lights or heavy traffic. When a vehicle stops, the vehicle speed is determined to be zero. 3) *passing through uneven road surfaces*. Speed bumps, potholes, and other severe road surfaces are common on urban roads. The accelerometer's readings from smartphones can be utilized to infer the vehicle speed, when a car is passing over uneven road surfaces.

The workflow of SenSpeed is shown in Fig. 3. SenSpeed uses two kinds of sensors in smartphones, accelerometers and gyroscopes, to estimate the vehicle speed. The accelerometer is used to monitor the vehicle acceleration and the gyroscope is used to monitor the vehicle angular speed. Getting the readings from the accelerometer and the gyroscope, SenSpeed first performs *Coordinate Reorientation* to align the phone's coordinate system with the vehicle's. After that, the raw speeds are obtained by calculating the integral of the aligned readings from the

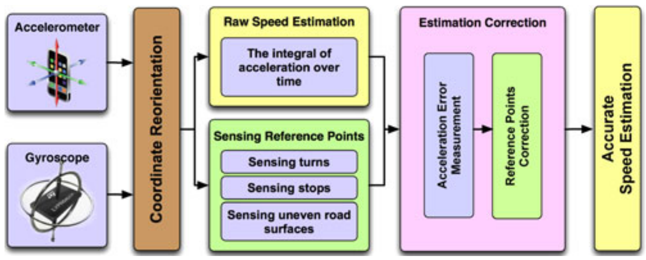



Fig. 3. System architecture.

accelerometer in *Raw Speed Estimation*. Meanwhile, SenSpeed senses reference points by analyzing the aligned readings from the accelerometer and the gyroscope in *sensing reference points* and infers the vehicle speed at each reference point. Next, in *acceleration error measurement*, the acceleration errors between each two adjacent reference points are calculated and then used to correct the raw speed estimations in *reference points correction*. Finally, SenSpeed outputs high-accuracy speed estimations. In order to achieve accurate speed estimations, the speeds at the two adjacent reference points need to be known. However, the speed at the next reference point is unknown on the real-time speed estimation, so the acceleration error between two reference points can not be calculated. Since we know the changes of the acceleration error over time are very small, *acceleration error measurement* uses the exponential moving average to derive the current acceleration error from recent histories. Therefore, SenSpeed can provide real-time speed estimation of vehicles.

## 4.2 Sensing Reference Points

To correct speed estimation from the integral of the accelerometer's readings, the acceleration error should first be measured. If we know the speed at reference points, the acceleration error can be inferred. SenSpeed senses natural driving conditions to identify reference points including *making turns*, *stopping* and *passing over uneven road surfaces*.

### 4.2.1 Sensing Turns

When a vehicle makes a turn, it experiences a centripetal force, which is related to its speed, angular speed and turning radius. Thus, by utilizing the accelerometer and the gyroscope, we can derive the tangential speed of a vehicle. Suppose a car is turning right, as is shown in Fig. 4, then $v = \omega R$, $a = \omega^2 R$, and $\omega = \omega'$, where $a$ is the centripetal

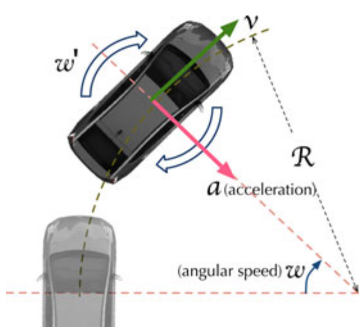



Fig. 4. Illustration of the circular movement when a car makes a turn.

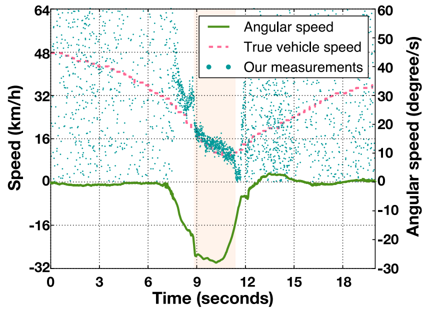



Fig. 5. The speed measurement at a turn reference point using centripetal acceleration and angular speed.

acceleration, $\omega'$ is the angular speed of the car, $R$ is the turning radius and $\omega$ is the angular speed that is related to the center of the orbit circle. Thus, we obtain

$$v = \frac{a}{\omega'}. \tag{3}$$

Since the centripetal acceleration $a$ and the angular speed $\omega$ can be obtained from the accelerometer and the gyroscope respectively, the speed can be calculated based on Eq. (3).

Fig. 5 plots the angular speed obtained from the gyroscope, the speed measurement from Eq. (3) and the speed from an OBD-II adapter when a vehicle makes a turn, i.e., at a turn reference point. It can be seen that the change of the angular speed is very clear at the turn reference point. If the readings from the gyroscope exceeded a trained threshold [3], SenSpeed determines the vehicle is making a turn. In addition, the values of the speed measurement from Eq. (3) at the turn reference point are very close to the ground truth.

Then, we analyze the speed measurement error at turn reference points. A series of experiments are conducted in real driving environments. Fig. 6 plots the CDF of the speed estimation errors at turn reference points. From this figure, we observe that $80$ percent of measurement errors are lower than $3.5$ km/h and the average error is about $1.8$ km/h, which indicates that the speed measurements at turn reference points are accurate. We also find that drivers tend to use a small angular speed to avoid an exorbitant centripetal acceleration when turning under high speed, but a small angular speed is more easily affected by noise. Thus, the accuracy decreases under the higher speed in Fig. 6. However, drivers usually make turns under $30$ km/h for driving safety, thus accurate vehicle speed can be inferred by using turns as reference points.

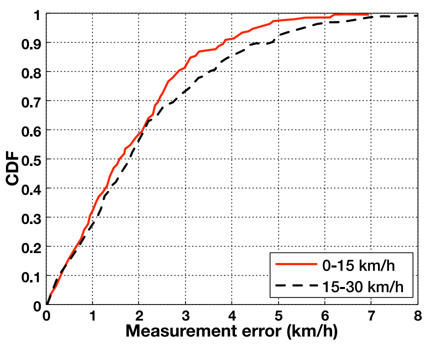



Fig. 6. CDF of the speed measurement errors at turn reference points.

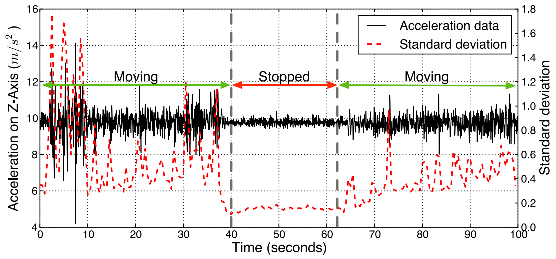



Fig. 7. Illustration of the acceleration on the vehicle's $z$-axis and the corresponding standard deviation when a vehicle stops.

### 4.2.2 Sensing Stops

The vehicle speed decreases to zero when a vehicle stops, so we can obtain the exact speed at a stop reference point. Based on our observation, the data pattern of the acceleration on the vehicle's $z$-axis for stop is remarkably different from that of moving. Fig. 7 plots the readings from the accelerometer's $z$-axis when the vehicle is moving and stops. From Fig. 7, it can be seen that the jitter of the acceleration on $z$-axis is almost disappeared and the standard deviation of the acceleration on $z$-axis remains low while the vehicle stops. Thus, the standard deviation of the acceleration on $z$-axis can be used to detect stop reference points. The standard deviation of the acceleration collected by smartphone is calculated in a small sliding window (usually the size of the window is set as 1 s, which is determined through empirical studies).

### 4.2.3 Sensing Uneven Road Surfaces

Speed bumps, potholes, and uneven road surfaces are common in urban environments. When a car is passing over uneven road surfaces, the accelerometer's readings from smartphones can also be utilized to infer the vehicle speed. Fig. 8 shows the accelerations on the car's $z$-axis, when a car is passing over a speed bump. The front wheels hit the bump first and then the rear wheels. In Fig. 8, the first peak is produced when the front wheel is passing over the bump and the second peak is produced by the rear wheels. Suppose we know the time interval $\triangle T$ between these two peaks, as well as the wheelbase $W$ of the vehicle, then the vehicle speed can be measured as $v = \frac{W}{\triangle T}$.

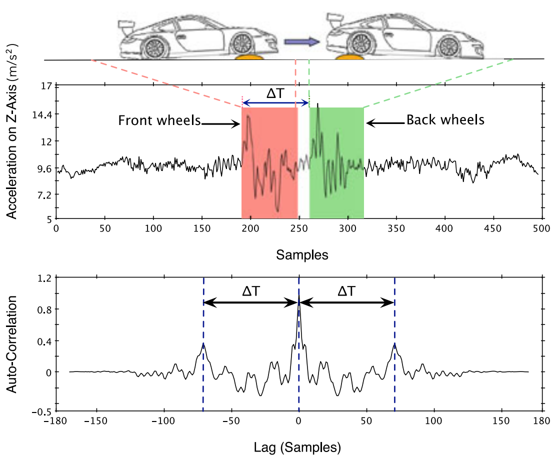



Fig. 8. Illustration of the acceleration on the vehicle's $z$-axis and the corresponding auto-correlation results when a car is passing over a bump.

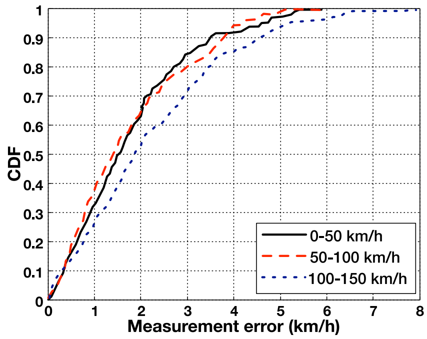



Fig. 9. CDF of the speed measurement errors at uneven road surface reference points.

Considering the similarity between the two peaks, we use *auto-correlation* analysis to find $\triangle T$. Given an acceleration sequence on z-axis, $\{Acc\}$, auto-correlation of lag $\tau$ is:

$$R(\tau) = \frac{E[(Acc_i - \mu)(Acc_{i+\tau} - \mu)]}{\sigma^2}, \tag{4}$$

where $\mu$ is the mean value of $Acc$ and $\sigma$ is the standard deviation. Fig. 8 also shows the auto-correlation results of the accelerometer's readings on z-axis. Obviously, $R(\tau)$ is an even function, so $R(\tau) = R(-\tau)$. To get the $\triangle T$, we need to find the maximum peak value except the one at $\tau = 0$, and the horizontal distance from the maximum peak to $\tau = 0$ equals to $\triangle T$. The time $\tau = 0$ can be obtained by checking the value of $Acc$ and when it is larger than a threshold which can be learned through empirical studies, such a time point is regarded as R(0). And for the wheelbase, we can get it from vehicle's product specifications.

Additionally, according to [21], the phone's location in the car has an influence on the shape of the two acceleration peaks. Specifically, If the phone is located in the front row, then the first peak has a larger amplitude than that of the second peak and vice versa. However, in our experiment, the phone's location is not changed during driving. Thus whatever the shapes of the two acceleration peaks, they are always similar to each other and can be detected by auto-correlation. Therefore, the phone positions will not affect the detection of $\triangle T$.

Fig. 9 depicts the accuracy of speed measurement at reference points including speed bumps, potholes, and other uneven road surfaces. It can be seen that 80 percent of measurement errors are lower than 2.7 km/h under the low speed (i.e., 0-50 km/h), 80 percent of measurement errors are lower than 3.4 km/h under the high speed (i.e., 100-150 km/h), and the average error is about 1.9 km/h. Also, we find that the vehicle speed affects the measurement accuracy, i.e., the accuracy slightly increases as the speed decreases. This is because that the accuracy is affected by the sampling rate. For example, suppose the vehicle speed is 30 km/h, the sampling rate of the accelerometer is 200 Hz and the wheelbase is 3 m, then the samples between the two wheels passing over a bump or pothole is $\frac{wheelbase}{speed} \cdot frequency \approx 72\ \ samples$. By contrast, when the vehicle speed is 130 km/h, the number of the samples decreases to $17 samples$. A smaller number of samples causes slightly worse accuracy. However, the average vehicle speed in urban area is relatively low (under 80 km/h).

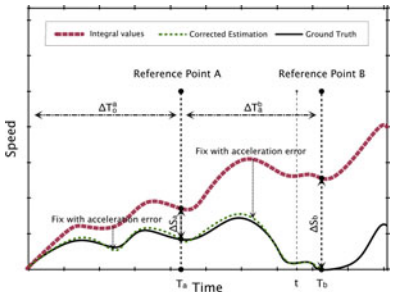



Fig. 10. Illustration of the acceleration error measurement using reference points.

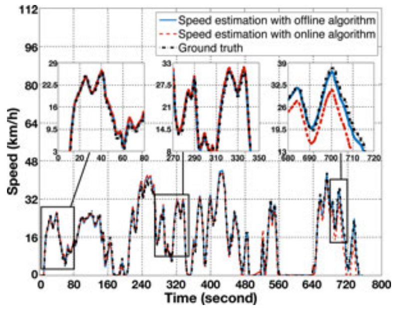



Fig. 11. Results of the offline and online vehicle speed estimation using SenSpeed.

Thus the vehicle speed at uneven road surfaces can be accurately measured in real driving environments.

### 4.3 Eliminating Accumulative Errors

With the above sensed reference points, once a vehicle makes turns, stops or passes over uneven road surfaces, SenSpeed is able to estimate the instant vehicle speed. In order to realize an accurate vehicle speed estimation, SenSpeed utilizes reference points to qualify the acceleration error and eliminate accumulative error.

In Fig. 10, the vehicle starts with zero speed, and there are two reference points $P_A$ and $P_B$ (i.e., the vehicle passes the reference point A and B at time $T_a$ and $T_b$ respectively). Suppose the integral value of the accelerometer's readings from zero to time $t$ is $S(t)$ and the measured speed at the reference point $x$ is $RPS_x$, the errors of the vehicle speed at the reference point $a$ and $b$ are $\triangle S(T_a) = S(T_a) - RPS_a$ and $\triangle S(T_b) = S(T_b) - RPS_b$ respectively. Since the value of acceleration error is nearly a steady constant and strongly related to the slope of the $\triangle S(t)$ curve, the acceleration error between $P_A$ and $P_B$ can be calculated as:

$$\tilde{A} = \frac{\triangle S(T_b) - \triangle S(T_a)}{\triangle T_a^b}. \tag{5}$$

where $\triangle T_a^b$ is the interval time between the reference points A and B. Thus, the accumulative error from $T_a$ to $t$ is $\int_{T_a}^{t} \tilde{A}\, \mathrm{d}t$, i.e., $\tilde{A} \times (t - T_a)$. Furthermore, the corrected speed estimation $S'(t)$ between $A$ and $B$ is:

$$S'(t) = S(t) - \triangle S(T_a) - \tilde{A} \times (t - T_a). \tag{6}$$

We then apply this algorithm to the same data used in Fig. 2, and the corrected estimation results are shown in Fig. 11. It can be seen that the corrected speeds match the ground truth closely. As a result, the mean estimation error after speed correction by using the reference points is $1.05$ km/h.

The above algorithm uses the information of two adjacent reference points to correct the speed estimations between these two points. However, it is an *offline algorithm* that can not be used for real-time speed estimations, because the information about the next reference point is unknown on real-time speed estimations. In order to achieve a real-time speed estimation, an *online algorithm* is proposed to estimate the current acceleration error. Since we know that the acceleration error changes slightly over time, thus the current acceleration error can be derived from the recent reference points. In particular, we utilize the *exponential moving average* to estimate the current acceleration error by using the recent reference points. When the $i$th reference point is sensed, the current acceleration error $\tilde{A}_i$ between the $i$th and $(i+1)$th reference point is updated through:

$$\tilde{A}_i = \alpha \cdot \tilde{A}_{i-1} + (1 - \alpha) \times \frac{\triangle S(T_i) - \triangle S(T_{i-1})}{\triangle T_{i-1}^i}, \tag{7}$$

where $\alpha$ is the weight coefficient. In SenSpeed, $\alpha$ is set to be $0.5$. $\alpha$ can determine which of the two portions is more important. The real-time speed estimation between the $i$th and the $(i+1)$th reference point is corrected by:

$$S'(t) = S(t) - \triangle S(T_i) - \tilde{A}_{i+1} \times (t - T_i). \tag{8}$$

We also apply this online algorithm to the same data used in Fig. 2, and present the corrected speed estimation in Fig. 11. We observe that there are some small differences between the online estimation and the ground truth, which indicates the online algorithm has a comparable accuracy when compared with the offline algorithm. Although the differences exist, they are very small and the mean estimation error of the online speed estimation algorithm is $1.74$ km/h.

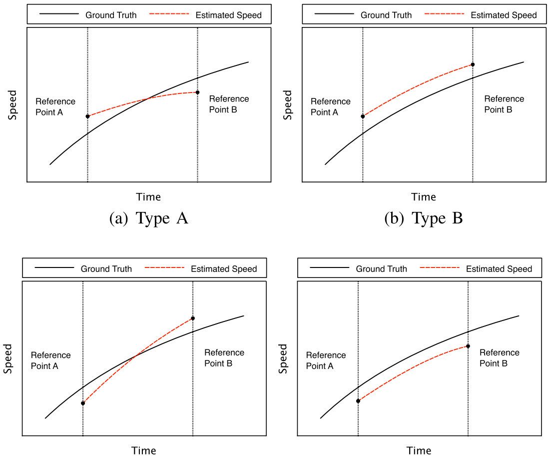
### 4.4 Complexity Analysis

We assume that the amount of sensing data from accelerometer and gyroscope is $N$ and the amount of reference points is $M$. The computational overhead of SenSpeed can be divided into the following parts.

*Raw vehicle speed calculation*. The raw vehicle speed is calculated from the integral of acceleration data over time. Each time an input from accelerometer is read, the system calculates the current vehicle speed, the last vehicle speed and the time interval between the adjacent calculations. It is clear that only the reading from accelerometer is a relevant factor for computational complexity, which is $O(N)$.

*Rotation matrix calculation*. SenSpeed only needs to calculate the rotation matrix when it starts. Thus the complexity of this calculation is $O(1)$. (see Section 4.6.1)

*Sensing reference points*. For turns, SenSpeed needs to monitor readings from gyroscope continuously and calculates vehicle speed when a turn is sensed. So the complexity of calculation is linearly associated with the number of the reading from gyroscope, which is $O(N)$. For stops, SenSpeed needs to monitor the $z$-axis of the readings from accelerometer and calculates its standard deviation to decide whether the car stops. Although it is complex to calculate the standard deviation of all data, there is one




(a) Type A (b) Type B (c) Type C (d) Type D

Fig. 12. Illustration of four types of reference points combination.

optimizations available. We can divide the input into several equal size windows and only calculate standard deviation within each window. In this way, the complexity of calculation can be reduced to $O(N)$. For uneven road surfaces, we need to calculate the auto-correlation function of the wave, which needs readings from accelerometer within a window of a certain length. However, since this calculation is only triggered when hit bump, the complexity of autocorrlation function is $O(1 \times M)$. Thus, the complexity is $O(M+N)$ for sensing reference points.

*Eliminating accumulative errors*. It can be seen from Equations 7 and 8 that the complexity is linearly associated with the reading from accelerometer, which is $O(N)$.

Based on the above analysis, the calculation complexity of SenSpeed is $O(M+N)$. Further, since the number of reference points is much smaller than the number of readings from sensors in practice. The complexity of SenSpeed is thus $O(N)$.

### 4.5 The Impact of Reference Point Accuracy

Since SenSpeed uses reference points to eliminate the speed estimation deviations caused by acceleration errors, the accuracy of vehicle speed will be degraded if there are errors on the instant speed estimation at reference points. Thus, it is necessary to discuss the impact of errors on the instant speed estimation at reference points.

The instant speeds at adjacent reference points are used to eliminate accumulative error. There are four possible relationships between the estimated and real speed at adjacent reference points, as shown in Fig. 12. We assume the time when a car passes adjacent reference points $A$ and $B$ are $T_A$ and $T_B$, respectively. Further, $S(t)$ represents the real vehicle speed at time $t$ and $S'(t)$ represents the speed estimated by SenSpeed. So the difference between the estimated and real vehicle speed at reference point $A$ and $B$ are $\Delta S_A = S'(T_A) - S(T_A)$ and $\Delta S_B = S'(T_B) - S(T_B)$ respectively. If we assume all systematic errors come form the error incurred from reference points, the average systematic error between $T_A$ and $T_B$ is:

$$AVG_{error} = \frac{1}{T_B - T_A}\int_{T_A}^{T_B} |S'(t) - S(T)|\mathrm{d}t, \tag{9}$$

*Type A*. Relationship between the real speed and estimated speed at reference points are $S'(T_A) > S(T_A)$ and $S'(T_B) < S(T_B)$ respectively. Thus, $AVG_{error}$ is:

$$AVG_{error} = \frac{\Delta S_A^2 + \Delta S_B^2}{2(S_A - S_B)} < \frac{|\Delta S_A| + |\Delta S_B|}{2}. \tag{10}$$

*Type B*. Relationship between the real speed and estimated speed at reference points are $S'(T_A) > S(T_A)$ and $S'(T_B) > S(T_B)$ respectively. Thus, $AVG_{error}$ is:

$$AVG_{error} = \frac{\Delta S_A + \Delta S_B}{2} = \frac{|\Delta S_A| + |\Delta S_B|}{2}. \tag{11}$$

*Type C*. Relationship between the real speed and estimated speed at reference points are $S'(T_A) < S(T_A)$ and $S'(T_B) > S(T_B)$ respectively. Thus, $AVG_{error}$ is:

$$AVG_{error} = \frac{\Delta S_A^2 + \Delta S_B^2}{2(S_B - S_A)} < \frac{|\Delta S_A| + |\Delta S_B|}{2}. \tag{12}$$

*Type D*. Relationship between the real speed and estimated speed at reference points are $S'(T_A) < S(T_A)$ and $S'(T_B) < S(T_B)$ respectively. Thus, $AVG_{error}$ is:

$$AVG_{error} = \frac{-\Delta S_A^2 - \Delta S_B^2}{2(S_A - S_B)} < \frac{|\Delta S_A| + |\Delta S_B|}{2}. \tag{13}$$

Therefore, the error is $AVG_{error} \leq \frac{|\Delta S_A| + |\Delta S_B|}{2}$, no matter what relationship between the real speed and estimated speed at adjacent reference points. Since it is nearly impossible that all estimated speeds at reference points are greater/smaller than the real speed, the above equation could be further extended for multiple reference points, as:

$$AVG_{error} < \frac{1}{N}\sum_{i=1}^{N} |\Delta S_i|. \tag{14}$$

It can be inferred from the equation above that the accuracy of SenSpeed is not significantly affected even if the speed error is relatively large at some reference points. That is because the speed error at a certain reference point should be less than the average error at all reference points. Therefore, SenSpeed is robust enough to cope with the errors at reference points.

### 4.6 Practical Issues

In the implementation of SenSpeed, we are facing several practical issues as follows.

#### 4.6.1 *Reorienting the Coordinate Systems*

SenSpeed can not derive meaningful vehicle speed estimations from sensors' readings unless the phone's coordinate system is aligned with the vehicle's. Since the pose of a smartphone in a vehicle could be arbitrary, we should first align the motion sensors' readings in the phone's coordinate system with the vehicle's before it can be utilized to estimate the vehicle speed.

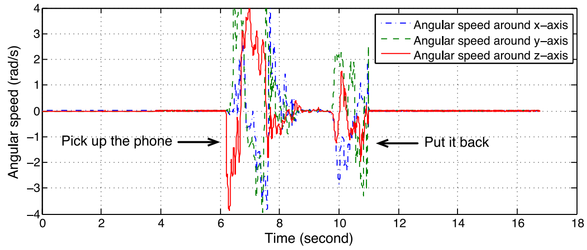



Fig. 13. Illustrating the change of the gyroscope's readings while a driver or passenger picks up a phone and then puts it back.

In our previous work [3], a rotation matrix $R = [\hat{i} \quad \hat{j} \quad \hat{k}]$ (where $\hat{i}$, $\hat{j}$ and $\hat{k}$ are three-dimensional coordinate vectors that represent the $x$, $y$ and $z$-axis direction of the vehicle coordinate system in the phone's respectively) is used to align the sensors' readings with vehicle's coordinate system. In the method, we first apply the low pass filter on the acceleration data to get $\hat{k}$, then extract $\hat{j}$ by observing the vehicle acceleration and deceleration when driving straight, and finally calculate $\hat{i} = \hat{j} \times \hat{k}$. We can obtain the rotated sensor reading aligned with vehicle's coordinate system by applying the rotation matrix.

However, [3] does not consider the transform accuracy on $z$-axis of the accelerometer. Consider the scenario that the gravity direction does not align with the $z$-axis of the vehicle when the vehicle is running on a slope. In order to keep the orthogonality of the vectors in rotation matrix, we recalibrate the $z$-axis vector by $\hat{k} = \hat{i} \times \hat{j}$.

After obtaining the corrected rotation matrix $R'$, a sensor reading $\hat{r}_s$ in the smartphone's coordinate system could be rotated by $\hat{r}_v = \hat{r}_s \cdot R'$. Now, a sensor reading in the smartphone's coordinate system can be aligned exactly with the vehicle's coordinate system.

### 4.6.2 Allowing Usage of Phone

The coordinate alignment uses a rotation matrix to align the phone's coordinate system with the vehicle's. Once the pose of phone is changed, the rotation matrix needs to be re-calculated. In order to solve the problem, SenSpeed first needs to detect the change of the phone's pose, then recalculates a new rotation matrix to align the phone's coordinate system with the vehicle's. Fig. 13 shows the readings from the gyroscope while a driver or passenger picks up a phone and then puts it back. It can be seen that the gyroscope's readings have large fluctuation on all three axis when the pose of phone changes. As a result, SenSpeed is able to detect the change of the phone's pose by monitoring the gyroscope's readings continuously. Once a change of the pose is detected, SenSpeed conducts coordinate alignment again to calculate a new rotation matrix. Specifically, both after a pick-up and a put-down of the phone is detected, once the gyroscope's readings return to a flat state, SenSpeed conducts coordinate alignment again to calculate a new rotation matrix.

### 4.6.3 Acquiring the Wheelbase Information

When SenSpeed uses uneven road surfaces as reference points, the information of wheelbase is needed to infer the speed at the reference points. Although we can get the wheelbase of a vehicle from the product specifications, it requires extra user operations to input the wheelbase into the system. To solve this problem, SenSpeed first needs to estimate the vehicle speed $v$ and the delay $\triangle T$ between the two peaks caused by the front wheels and the back wheels. Then, the wheelbase can be calculated as $W = v \cdot \triangle T$. For the time delay $\triangle T$, we can measure it with the method presented in Section 4.2.3. For the vehicle speed, SenSpeed can obtain it either with GPS or itself:

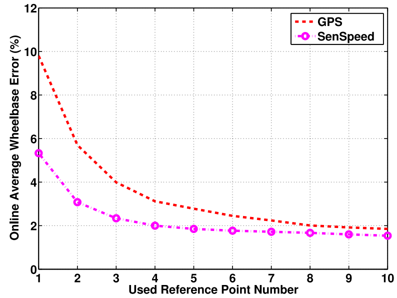



Fig. 14. The estimation error of wheelbase using GPS or SenSpeed itself.

*Measure with GPS*. The position of a smartphone can be determined by using GPS. We can get the average vehicle speed by utilizing the position and time interval from adjacent GPS data.

*Measure with self-estimated speed*. SenSpeed first uses stops and turns as reference points to estimate the vehicle speed, then uses the estimated speed to learn the wheelbase. After a couple of wheelbase measurements, the accurate wheelbase information can be obtained.

The advantage of the latter solution is that it needs neither user inputs nor the GPS, and only involves in SenSpeed itself. It is a self-learning process to obtain the wheelbase. In order to further compare the efficiency and accuracy of the two methods above, we perform a series of experiments to illustrate the average error, which is shown in Fig. 14. It can be seen that the error of either method decreases when the number of reference points increases. The reason is the elimination of white noise and improvement of accuracy by using the average of many measurements. Besides, the accuracy of wheelbase estimation by using GPS is not as good as using SenSpeed itself. That is because the low sampling rate of GPS is not good enough to make prompt responses to the change in speed, which leads to a accuracy degradation. Also, the figure shows that the error of wheelbase can be reduced to around 2 percent when the number of reference points is larger than 5 using the GPS method or larger than 3 using the SenSpeed method. This accuracy is good enough to support the above wheelbase measurement method by utilizing bumps of the vehicle. Finally, in terms of accuracy, using SenSpeed itself is a better choice than using GPS.

To further evaluate the impact of unknown wheelbase, we carry on another experiment which compares the performances of SenSpeed using known and unknown wheelbase, respectively. Fig. 15 shows the CDF of the speed estimation errors using known and unknown wheelbase. It can be seen the estimation errors of both offline and online algorithms show virtually no drop when the wheelbase become unknown. Thus, unknown wheelbase does not affect the speed estimation much.

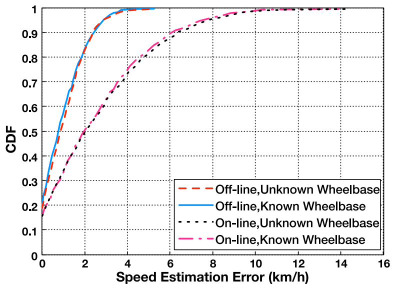



Fig. 15. CDF of the speed estimation errors using known and unknown wheelbase.

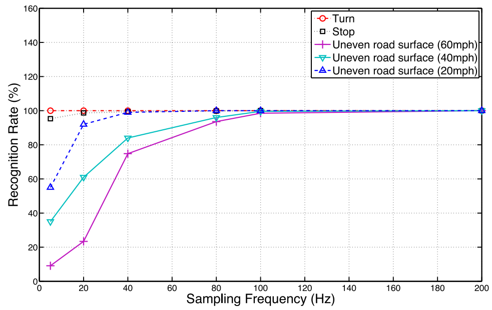



Fig. 17. The relationship between the recognition rate of reference points and sampling frequency.

## 5 Energy Optimization

Energy consumption is an important issue on smartphones. In this section, we first analyze the energy consumption of SenSpeed. Then, an optimized SenSpeed is proposed, which could effectively reduce the energy consumption of SenSpeed by automatically adjusting the sampling rate based on the density of the reference point.

### 5.1 Energy Consumption Model of Sensors

In order to optimize the energy consumption of SenSpeed, we first introduce the operation mode of smartphone motion sensors. Modern motion sensors usually have 6 power modes, i.e., *IdleMode*, *Accelerometer Low Power Mode*, *Accelerometer Mode*, *Gyroscope Mode* and *Gyroscope Accelerometer Mode*. Energy consumption is the lowest when the motion sensor is running under Idle Mode, which is only $5\mu A$. Besides, energy consumption is also relatively low when Accelerometer Low Power Mode is on, which is 10-140$\mu A$. The specific energy consumption of this mode is related with sampling rate, which is shown in Fig. 16. It is obvious that the power consumption of accelerometer is approximately linearly associated with the sampling rate. There is no low power mode for gyroscope. Once the gyroscope starts running, it consumes much more energy than accelerometer, which is 3.6 mA. Therefore, there are two important principles in reducing power consumption of Senspeed: Since the power consumption of gyroscope is much higher than that of accelerometer, SenSpeed needs to try to avoid using gyroscope in order to reduce the power consumption; Since the low power mode is available when the sampling rate is low, a significant amount of energy could be saved by reducing the sampling rate of SenSpeed.

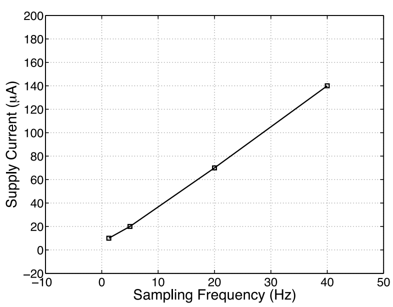



Fig. 16. The relationship between sensor supply current and sampling frequency.

### 5.2 Frequency Adaptation Optimization of SenSpeed

#### 5.2.1 Impact of Sampling Rate on Accuracy

In order to make SenSpeed more energy efficient, the sampling rate of accelerometer need to be reduced. A low sampling rate, however, could result in reference point recognition problem, which degrades the accuracy of SenSpeed. We perform some experiments on the impact of sampling rate on reference point recognitions. Fig. 17 shows that the percent of reference point recognition rate under different sampling rates.

*Turning reference point*. It can be seen from Fig. 17 that the sampling rate does not affect the recognition rate of turning reference points. That is because a car usually takes more than 1s to make a turn. Even if the sampling rate of motion sensors is reduced, SenSpeed could still sense a significant change in readings from gyroscope while the car is turning. So the sampling rate of motion sensors does not affect the recognition of turning reference points.

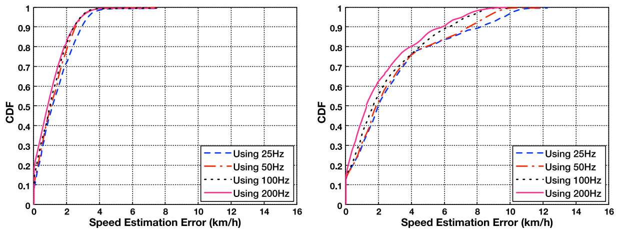
*Stopping reference point*. Compared with turning reference points, a low sampling rate has a minor effect on the recognition of stopping reference points. Since we use the standard deviation of the vehicle's acceleration along $z$-axis when sensing stopping reference points, a low sampling rate could bring in errors in recognition. However, we can also see from Fig. 17 that when sampling frequency is above 20 Hz, the recognition rate does not changed.

*Uneven road surface reference point*. Different with the two kinds of reference points above, the low sampling rate could significantly degrades the recognition of uneven road surface. The recognition rate is below 10 percent if the low sampling rate is used. With the reduced sampling frequency, the characteristics of the original wave gradually fade out. When the sampling rate is below 25 Hz, it is hard for SenSpeed to recognize bumping from the wave.

Also, we analyze the impact of the sampling rate on the whole SenSpeed system. Fig. 18 shows that the CDF of the speed estimation errors using different sampling rates. It can be seen the estimation errors of the offline and online algorithm raise slightly when the sampling rate drops from 200 to 25 Hz.

#### 5.2.2 Impact of Sampling Rate on Power Consumption

SenSpeed incurs two kind of power consumption while it is running. The first is the power consumption from the motion sensor. A low sampling rate could significantly reduce the power consumption of the motion sensors. The



(a) Offline (b) Online

Fig. 18. CDF of the speed estimation errors using different sampling rates, i.e., 200, 100, 50, 25 Hz.

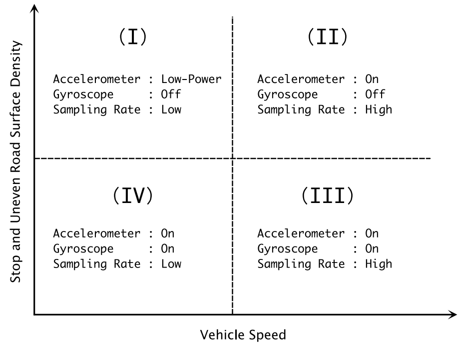



Fig. 19. Four sampling strategies under different combinations of vehicle speed and reference point density.

second is power consumption of CPU. In order to infer vehicle speed form motion sensor readings, SenSpeed needs use the computation power of CPU, whose energy consumption is linearly associated with its workload. From the discussion in Section 4.4, the computation complexity of SenSpeed is $O(N)$, where $N$ is the number of readings from the motion sensors. Thus, a low sampling rate can reduction data size generated by motion sensors and finally lead to the reduce of energy consumption of CPU.

Therefore, the benefit of reducing sampling rate is two-fold. The power consumption of motion sensors and CPU can both be reduced by reducing sampling rate. Taking the result shown in Fig. 17 into consideration, although the accuracy of uneven road surface reference points recognition is degraded under the low sampling rate, we can still eliminate the accumulative error by using all three kind of reference points. Through the adjustment of motion sensors' sampling rate, we could achieve power consumption optimization without losing speed estimation accuracy.

#### 5.2.3 *Sampling Rate Adaptation*

Based on the analysis above, a novel sampling rate adaptation method for SenSpeed is proposed. In order to reduce power consumption, it utilizes the estimated vehicle speed and reference point density to adjust the sampling rate of accelerometer and gyroscope. The rationale of this method is: when the density of stop reference points and uneven road surface reference points are high, gyroscope is turned off to save energy; when the vehicle speed is low, we could make the accelerometer working at the low power mode; when the density of stop reference points and uneven road surface reference points are too low, we turn on gyroscope to increase the density of those reference points.

For different combinations of vehicle speed and reference points density, SenSpeed employs different sampling strategy, which is shown in Fig. 19. It can be seen that mode (I) is the most power efficient one, for it not only turnoff the gyroscope and make the accelerometer run under low power mode, but also reduce computation by cutting down sampling rate. On the contrary, mode (III) consumes most power, because both the accelerometer and gyroscope use a high sampling rate.

Fig. 20 illustrates the change in sampling rate based on different combinations of vehicle speed and reference points density. First, mode (IV) is used under a low speed and reference point density condition. In this phase, the sampling rate increases with the increase in the vehicle speed. Then, when the vehicle speed is higher than a pre-learned threshold, the sampling mode is changed to mode (III). In order to keep the accuracy of SenSpeed, the sampling frequency of the accelerometer and the gyroscope must be high to sense as much reference points as possible. After the vehicle slows down, the sampling mode changes back to mode (IV), until a increase in reference point density is encountered. According to Fig. 19, we use mode (I) when the vehicle speed is low and reference point density is high. Finally, after the vehicle speed goes up, Senspeed uses mode (II) to sense reference points.

## 6 Evaluation

In this section, we evaluate our speed estimation system, SenSpeed, in real driving environments using two types of smartphones in two different cities.

### 6.1 Prototype

We implement SenSpeed as an open source Android App and install it on smartphones: Galaxy Nexus (Manufactured by Samsung, Android 4.2, 1.2 GHz dual-core, 1 GB RAM, Maximum sampling rate of accelerometer and gyroscope: 100 Hz) and Nexus4 (Manufactured by LG, Android 4.2, 1.5 GHz quad-core, 2 GB RAM, Maximum sampling rate of accelerometer and gyroscope: 200 Hz). SenSpeed senses the natural driving conditions by using both accelerometers and gyroscopes. Meanwhile, the raw data of accelerometers' and gyroscopes' reading are stored on smartphones for off-line data analysis.

### 6.2 Real Road Driving Environments

To evaluate the generality and robustness of SenSpeed, we conduct experiments in two typical urban environments: one is in Shanghai, China with Nexus4, and the other one is in New York City, USA with Galaxy Nexus. Fig. 21 shows

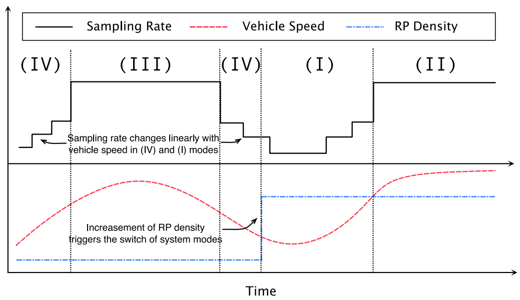
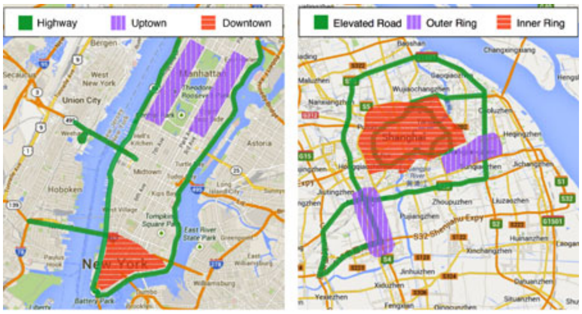



Fig. 20. Four sampling strategies under different combinations of vehicle speed and reference point density.




(a) Shanghai (b) Manhattan

Fig. 21. Areas covered by our experiments in two cities marked by different colors including red, purple, and green to represent different regions and types of roads in urban environments.

the areas that our traces covered in these two cities. In Shanghai, we evaluate our system on different road types including local roads and elevated roads, as well as different regions including the area within Inner Ring (financial districts and shopping centers) and the area outside Outer Ring (living districts). Similarly in Manhattan, two kinds of road types (local road and highway), as well as two regions (the financial district in Downtown and the living district in Uptown), are covered in our experiments. Furthermore, experiments are conducted in both peak time and off-peak time. In addition, three types of cars are involved in our experiments: Volkswagen Lavida and Passat are used in Shanghai, and Nissan Altima is used in Manhattan, New York City. We collect about 2,500 kilometers driving traces in Shanghai for over one month and 1,600 kilometers driving traces in Manhattan for over three weeks.

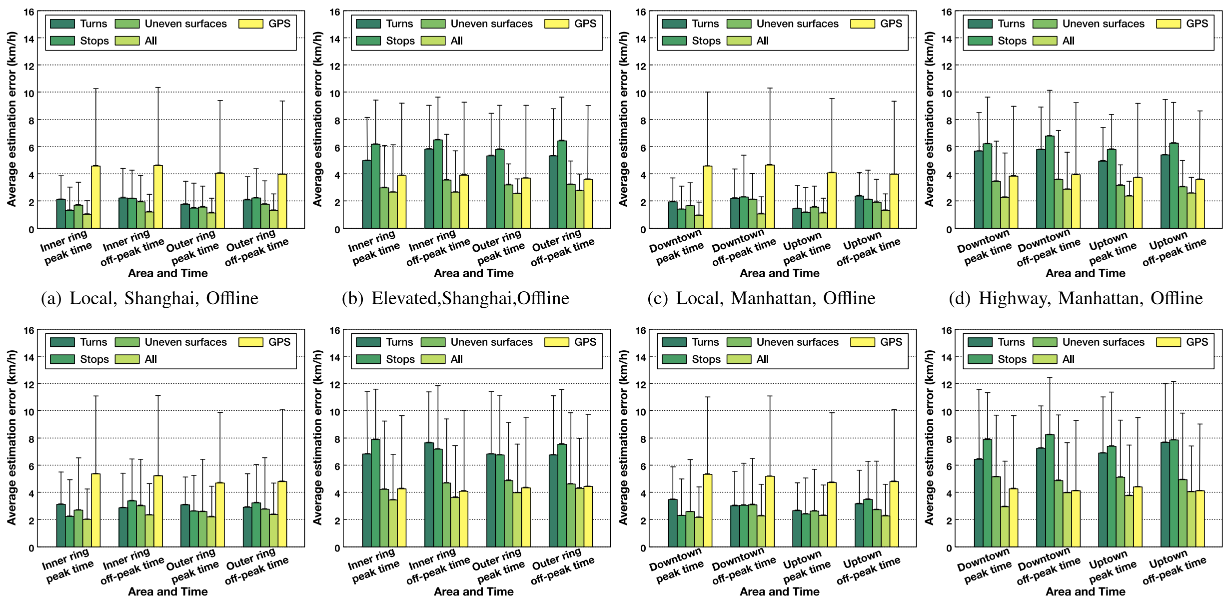
### 6.3 Speed Estimation Accuracy

We evaluate the speed estimation accuracy of our system when driving on different types of road and under different periods of day. We experimented with two type of speed estimations: online and offline speed estimation. We compare the estimated speed by our system with that of ground truth and the GPS. The ground truth is obtained from the calibrated (i.e., with respect to tire pressure and tire worn) OBD-II adapter. Fig. 22 presents the average estimation error in both Shanghai and Manhattan for online, offline and GPS estimations.

*Overall performance.* From Fig. 22, we observe that our speed estimation (both online and offline) leveraging all the reference points (i.e., All) has low errors and achieves better accuracy than that of GPS under all types of roads and different periods of day. For example, on local road in Manhattan, the average error for the offline and online speed estimation is only 1.1 and 2.1 km/h respectively, whereas it is up to 4.5 and 5.0 km/h for GPS respectively (Due to the next sample from GPS is unknown, the online estimation using GPS has lower accuracy). Further, we find that the offline estimation is slightly better than that of the online estimation, and this is because the value of acceleration error is not exactly accurate due to the lack of the next reference point information.

*Accuracy versus reference points.* We next evaluate the estimation accuracy of our system by using only one type of reference points. We find that the average estimation error on local road is still lower than of GPS even if only one type of reference points is used in both cities. However, the speed estimation using turns or stops is worse than that of GPS under elevated road and highways due to the fact that there are less turns and stops can be used as reference points. Still, we find that by using uneven road surfaces only, we can achieve comparable or better accuracy when comparing with GPS under all types driving roads.

*Accuracy versus type of roads.* Fig. 22 shows the road type affects the speed estimation accuracy. In particular, the average speed estimation errors on the elevated road or highway are higher than that on the local road (e.g. in Shanghai, the average error of the offline and online speed estimation is 1.1 and 2.0 km/h respectively on local roads, but it is up to 2.7 and 4.0 km/h respectively on the elevated road). This is because there are less reference points on the




(a) Local, Shanghai, Offline (b) Elevated,Shanghai,Offline (c) Local, Manhattan, Offline (d) Highway, Manhattan, Offline

(e) Local, Shanghai, Online (f) Elevated,Shanghai,Online (g) Local, Manhattan, Online (h) Highway, Manhattan, Online

Fig. 22. The average estimation error of the vehicle speed in Shanghai and Manhattan.

TABLE 1
Density of Reference Points in Shanghai and Manhattan

| Type & Period | | Shanghai | | Manhattan | |
|---|---|---|---|---|---|
| | | Local Road | Elevated Road | Local Road | Highway |
| All | peak | 6.81/km | 3.96/km | 9.05/km | 5.60/km |
| | Off-peak | 5.59/km | 3.06/km | 7.02/km | 5.58/km |
| Stop | peak | 2.67/km | 1.13/km | 4.12/km | 0.26/km |
| | Off-peak | 1.45/km | 0.23/km | 2.09/km | 0.24/km |
| Turn | | 2.42/km | 0.19/km | 1.41/km | 0.13/km |
| uneven road surfaces | | 1.72/km | 2.64/km | 3.52/km | 5.21/km |

elevated road and highway than those on local road. However, the average estimation error on elevated road and highway is still lower than that of GPS. Further, for GPS, we can observe the average estimation error on local road is higher than the error on highway due to the urban canyon environment (i.e., local road) causes lower GPS availability and accuracy.

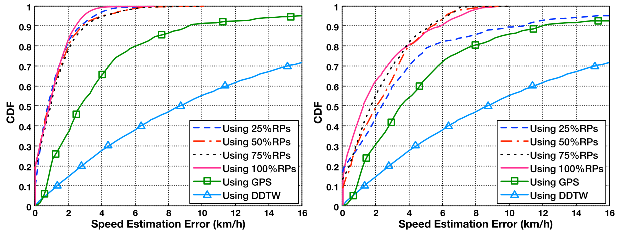
Finally, we find that the period of day and various districts slightly affect the estimation accuracy. The average estimation error at the peak time in financial district is slightly lower than at the off-peak time in living district respectively. It is the heavy traffic that causes more stops and increases the density of stops. Since only the density of stops is affected by traffic, overall performance of SenSpeed is not affected evidently by various districts and the period of day.

### 6.4 Impact of Reference Points Density

Our accurate vehicle speed estimation is built upon the identified reference points (i.e., turns, stops, and uneven road surfaces) from the natural driving conditions. We thus first statistically analysis the reference point density in urban environments using all the data collected in these two cities. The details are presented in Table 1.

Our overall observation from Table 1 is that the reference point is very dense in both Shanghai and Manhattan. For local road, there are about six reference points per km (rps/km) in Shanghai and around 7 rps/km in Manhattan. Whereas we have about 3 rps/km on elevated road in Shanghai and about 6 rps/km on highway of Manhattan on average. Further, we find that the density of reference points is affected by road types and period of day. Specifically, the density of stops nearly doubled on peak time in both Shanghai and Manhattan due to different traffic conditions. And the density of turns and stops on the local road is much higher than that on highway or elevated road. Moreover, one surprising finding is that the density of uneven road surfaces on highway or elevated road is much higher than that on local road. This is because highway and elevated road have lots of road joints which causes high density of uneven road surfaces. Due to the density of turns and uneven road surfaces only depends on the travel path, there is no density difference of these two types between peak-time and off-peak time periods.

To further evaluate the accuracy and robustness of SenSpeed, we analyze the speed estimation errors using



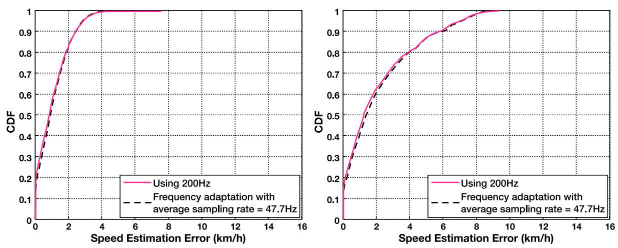

(a) Offline (b) Online

Fig. 23. CDF of the speed estimation errors using different percentages of reference points, i.e., 25, 50, 75, and 100 percent.

different percentages of reference points and compare the estimated speed with the ground truth from OBD-II.

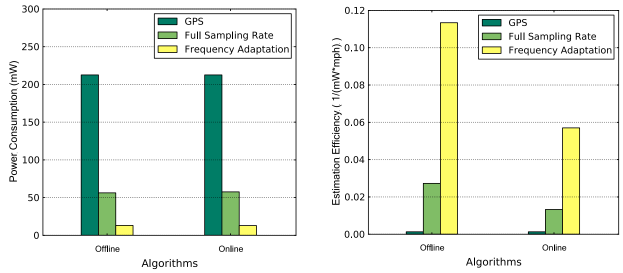
Fig. 23 shows the CDF of the speed estimation errors using different percentages (i.e., 25, 50, 75 and 100 percent) of reference points. As we have seen, we can always get high accurate speed estimations for the offline speed estimation regardless how many percent of reference points are used. For example, 80 percent of estimation errors are lower than 1.9 km/h if all reference points are used for the offline speed estimation, and the accuracy shows no obvious change when reference points are reduced from 100 to 25 percent. For the online speed estimation, 80 percent of estimation errors are lower than 3.9 km/h if all reference points are used, and also the accuracy shows no obvious change when reference points are reduced from 100 to 50 percent. Even if the reference points are reduced to 25, 65 percent of estimation errors are still lower than 3.7 km/h. Thus, the proposed online speed estimation is highly accurate and robust to different densities of reference points in urban environments. Although the accuracy of SenSpeed is affected by the density of reference points, excessive reference points do not contribute much to the estimation accuracy. For example, in the online speed estimation, the speed estimation errors using 50 percent reference points are very close to the estimation errors using 100 percent reference points. Thus, SenSpeed is robust when facing a decline of reference point density in urban environments.

Meanwhile, we compare SenSpeed with GPS and Derivative Dynamic Time Warping [10]. From Fig. 23, it can be seen that SenSpeed significantly outperforms DDTW in both offline and the online speed estimation. Compared with GPS, SenSpeed still has a higher accuracy. For example, 80 percent of GPS's estimation errors are lower than 8 km/h. By contrast, 85 percent of SenSpeed's estimation errors are lower than 8 km/h only when 25 percent of the reference points are used for the online speed estimation.




(a) Offline (b) Online

Fig. 24. CDF of the speed estimation errors using different sampling rates, i.e., 200 Hz and frequency adaptation.



(a) Power Consumption (b) Power Efficiency

Fig. 25. Energy Analysis on speed estimation using GPS, SenSpeed with full sampling rate and with frequency adaptation.

### 6.5 Impact of Sensor Sampling Rate

In order to evaluate the sampling rate adaptation method of SenSpeed, we first analyze its impact on vehicle speed estimation accuracy. Fig. 24 shows the CDF of the speed estimation errors using and without using the sampling rate adaptation. It can be seen that no matter under online or offline environment, SenSpeed could precisely sense reference points and estimate vehicle speed accurately with the sampling rate adaptation. Compared with the highest sampling frequency (200 Hz), the average sampling frequency of SenSpeed is reduced to 47.7 Hz. However, the speed estimation accuracy remains nearly unchanged, which changes from 1.05 to 1.08 km/h with offline algorithm and from 2.1 to 2.17 km/h with online algorithm. Thus, the sampling rate adaptation method could significantly reduce sampling rate without the degradation of speed estimation accuracy.

Next, we evaluate the impact of the sampling rate adaptation on power consumption by using Nexus4 with the maximum sampling rate is 200 Hz. We collect around 500 kilometers driving traces for over eight hours on local roads and elevated roads in Shanghai. It can be seen from Fig. 25a that GPS has a relative large power consumption, which is 213 mW on average. Besides, SenSpeed with full sampling rate consumes 56 and 58 mW with offline and online algorithm, respectively. Compared with that, SenSpeed with frequency adaptation only need power consumption of 13 mW, which is very power efficient.

In order to consider both the speed estimation accuracy and power consumption, a novel evaluation criterion, *Estimation Efficiency*, is proposed, i.e., $EstimationEfficiency = 1/(EstimationError \times PowerConsumption)$. It can be seen from Fig. 25b that SenSpeed with frequency adaptation achieves the highest Estimation Efficiency, and the Estimation Efficiency of GPS is the lowest.

## 7 Conclusion

In this paper, we address the problem of performing accurate vehicle speed estimation in urban environments to support pervasive vehicular applications. We employ smartphone sensors to sense natural driving conditions to achieve high estimation accuracy. In particular, we propose a vehicle speed estimation system called SenSpeed to identify three useful reference points, including making turns, vehicle stopping, and passing through uneven road surfaces, to measure and eliminate the errors caused by directly using phone's accelerometer readings for speed estimation. The key insight is that natural driving conditions present unique features and can be exploited to enable accurate real-time vehicle speed estimation. Our extensive experiments driving in two different cities over one month time period show that SenSpeed can estimate the vehicle speed in real-time with a low average error of 2.12 km/h, while achieving 1.21 km/h during the offline estimation.

## Acknowledgments

This work is partially supported by National Science Foundation grants CNS1217387, CNS1318748, SES1450091, the 973 Program (2014CB340303), NSFC (Nos. 61373157, 61472254, 61170238, 61420106010), Singapore NRF (CREATE E2S2), Program for Changjiang Scholars and Innovative Research Team in University (IRT1158, PCSIRT). J. Yu is the corresponding author.

## References

[1] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 4372–4378.

[2] F. Chausse, J. Laneurit, and R. Chapuis, "Vehicle localization on a digital map using particles filtering," in *Proc. Symp. Intell. Veh.*, 2005, pp. 243–248.

[3] 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 11th Annu. Int. Conf. Mobile Syst., Appl. Serv.*, 2013, pp. 41–54.

[4] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt, "Wreckwatch: Automatic traffic accident detection and notification with smartphones," *Mob. Netw. Appl.*, vol. 16, no. 3, pp. 285–303, Jun. 2011.

[5] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles, "Driving behavior analysis with smartphones: Insights from a controlled field study," in *Proc. ACM 11th Int. Conf. Mobile Ubiquitous Multimedia*, 2012, pp. 36:1–36:8.

[6] D. Johnson and M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proc. IEEE Int. Conf. Intell. Transportation Syst.*, 2011, pp. 1609–1615.

[7] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proc.6th ACM Conf. Embedded Netw. Sens. Syst.*, 2008, pp. 323–336.

[8] T. N. Schoepflin and D. J. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," *IEEE Trans. Intell. Transportation Syst.*, vol. 4, no. 2, pp. 90–98, Jun. 2003.

[9] A. Gorski. understanding gps performance in urban environments. [Online]. Available: http://blogs.agi.com/agi/2011/01/04/understanding-gps-performance-in-urban-environments/, 2011.

[10] G. Chandrasekaran, T. Vu, A. Varshavsky, M. Gruteser, R. Martin, J. Yang, and Y. Chen, "Tracking vehicular speed variations by warping mobile phone signal strengths," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2011, pp. 213–221.

[11] G. Chandrasekaran, T. Vu, A. Varshavsky, M. Gruteser, R. P. Martin, J. Yang, and Y. Chen, "Vehicular speed estimation using received signal strength from mobile phones," in *Proc. 12th ACM Int. Conf. Ubiquitous Comput.*, 2010, pp. 237–240.

[12] B. Coifman, "Improved velocity estimation using single loop detectors," *Transportation Res. Part A: Policy Pract.*, vol. 35, no. 10, pp. 863–880, 2001.

[13] B. Coifman, "Using dual loop speed traps to identify detector errors," *Transportation Res. Rec.: J. Transportation Res. Board*, vol. 1683, pp. 47–58, 1999.

[14] V. Cevher, R. Chellappa, and J. McClellan, "Vehicle speed estimation using acoustic wave patterns," *IEEE Trans. Sonignal Process.*, vol. 57, no. 1, pp. 30–47, Jan. 2009.

[15] V. Tyagi, S. Kalyanaraman, and R. Krishnapuram, "Vehicular traffic density state estimation based on cumulative road acoustics," *IEEE Trans. Intell. Transportation Syst.*, vol. 13, no. 3, pp. 1156–1166, Sep. 2012.

[16] H. Li, H. Dong, L. Jia, D. Xu, and Y. Qin, "Some practical vehicle speed estimation methods by a single traffic magnetic sensor," in *Proc. IEEE Int. Conf. Intell. Transportation Syst.*, 2011, pp. 1566–1573.

[17] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson, "Virtual trip lines for distributed privacy-preserving traffic monitoring," in *Proc. ACM 6th Int. Conf. Mobile Syst., Appl. Serv.*, 2008, pp. 15–28.

[18] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proc. ACM 7th ACM Conf. Embedded Netw. Sens. Syst.*, 2009, pp. 85–98.

[19] D. GundlegÃĕrd, "Handover location accuracy for travel time estimation in GSM and umts," *IET Intelligent Transport Systems*, vol. 3, no. 7, pp. 87–94, 2009.

[20] S. Thajchayapong, W. Pattara-Atikom, N. Chadil, and C. Mitrpant, "Enhanced detection of road traffic congestion areas using cell dwell times," in *Proc. IEEE Intell. Transportation Syst. Conf.*, 2006, pp. 1084–1089.

[21] Z. He, J. Cao, X. Liu, and S. Tang, "Who sits where? infrastructure-free in-vehicle cooperative positioning via smartphones," *Sensors*, vol. 14, pp. 11 605–11 628, 2014.

**Jiadi Yu** received the PhD degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2007. He is currently an assistant professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. Prior to joining Shanghai Jiao Tong University, he was a postdoctoral fellow in the Data Analysis and Information Security (DAISY) Laboratory at the Stevens Institute of Technology from 2009 to 2011. His research interests include cyber security and privacy, mobile and pervasive computing, cloud computing, and wireless sensor networks. He is a member of the IEEE and the IEEE Communication Society.

**Hongzi Zhu** received the PhD degree from the Department of Computer Science and Engineering at the Shanghai Jiao Tong University in 2009. He is an associate professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University, China. His research interests include vehicular ad hoc networks, wireless networks, mobile computing, and network security. He is a member of the IEEE Computer and the IEEE Communication Society.

**Haofu Han** received the master's degree in the Department of Computer Science and Engineering, Shanghai Jiao Tong University in 2014. He is currently a software engineer at Tencent. His research interests include mobile computing.

**Yingying (Jennifer) Chen** received the PhD degree in Computer Science from Rutgers University. She is an associate professor in the Department of Electrical and Computer Engineering at the Stevens Institute of Technology. Her research interests include cyber security and privacy, mobile and pervasive computing, and mobile healthcare. She has published more than 80 journals and referred conference papers in these areas. Prior to joining Stevens, she was with Alcatel-Lucent. She received the US National Science Foundation CAREER Award and Google Faculty Research Award. She also received the NJ Inventors Hall of Fame Innovator Award. She received the Best Paper Award from ACM International Conference on Mobile Computing and Networking (MobiCom) 2011. She also received the IEEE Outstanding Contribution Award from IEEE New Jersey Coast Section each year from 2005 to 2009. Her research has been reported in numerous media outlets including MIT Technology Review, *Wall Street Journa*l, and National Public Radio. She is on the editorial boards of the *IEEE Transactions on Mobile Computing* (IEEE TMC), *IEEE Transactions on Wireless Communications* (IEEE TWireless), and *IEEE Network Magazine*. She is a senior member of the IEEE.

**Jie Yang** received the PhD degree in computer engineering from the Stevens Institute of Technology in 2011. He is currently an assistant professor in the Department of Computer Science at Florida State University. His research interests include cyber security and privacy, and mobile and pervasive computing, with an emphasis on network security, smartphone security and applications, security in cognitive radio and smart grid, location systems, and vehicular applications. His research is supported by US National Science Foundation (NSF) and Army Research Office (ARO). He recieved the Best Paper Runner-up Award from IEEE Conference on Communications and Network Security (CNS) 2013 and the Best Paper Award from ACM MobiCom 2011.

**Yanmin Zhu** received the PhD degree from the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology in 2007. He is currently an associate professor with the Department of Computer Science and Engineering at Shanghai Jiao Tong University. His research interests include wireless sensor networks and mobile computing. Before that, he was a research associate with the Department of Computing at Imperial College London. He is a member of the IEEE and the IEEE Communication Society.

**Zhongyang Chen** received the bachelor's degree in computer science from the Beijing University of Posts and Telecommunications, Beijing, China, in 2013. She is a master's student in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Her research interests include mobile computing and wireless sensor networks.

**Guangtao Xue** received the PhD degree from the Department of Computer Science and Engineering at the Shanghai Jiao Tong University in 2004. He is an associate professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University, China. His research interests include vehicular ad hoc networks, wireless networks, mobile computing, and distributed computing. He is a member of the IEEE Computer Society and the IEEE Communication Society.

**Minglu Li** graduated from the School of Electronic Technology, University of Information Engineering, in 1985 and received the PhD degree in computer software from Shanghai Jiao Tong University (SJTU) in 1996. He is a full professor and the vice chair of the Department of Computer Science and Engineering and the director of Grid Computing Center of SJTU. Currently, his research interests include grid computing, services computing, and sensor networks.