

## Research Article

# A New Approach to Estimate Real-Time Traveling Speed with Accelerometer

Xiaojie Zong<sup>1</sup> and Xiangji Wen<sup>2</sup>

<sup>1</sup>*School of Computer & Information Engineering, Zhejiang Gongshang University, Hangzhou, China*

<sup>2</sup>*Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China*

Correspondence should be addressed to Xiaojie Zong; [zxj6330@163.com](mailto:zxj6330@163.com)

Received 27 July 2015; Revised 30 September 2015; Accepted 1 October 2015

Academic Editor: Javier Bajo

Copyright © 2015 X. Zong and X. Wen. 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 order to estimate the traveling speed, three parts are mainly described in this paper, including coordinate system relationship establishment between vehicle and smartphone, proper character extraction of acceleration in the vehicle coordinate system and model training process.

## 1. Introduction

In this paper we propose a real-time traffic condition estimation system with accelerometer (RTCESA), which is not sensitive to the environment and can work well in urban canyons. In the area of speed estimation with acceleration, some researchers have made some contributions [1, 2]. However, they only measured the integral of acceleration to evaluate the speed value and the experiment result showed a poor performance. Different with them, we estimate the traveling speed after analyzing the acceleration characteristic values in advancing direction and vertical direction. The main problems are how to acquire the acceleration value in the two directions and obtain the excellent characteristics and the model to estimate the traveling speed.

The principle of the system is shown in Figure 1. Firstly, the driving status and speed estimation are finished by smartphones and then sent to the server. Secondly, after data fusion and data processing, the sever could estimate the traffic condition and send the estimated result to the internet; for example, If the numbers of starting state and braking state are large, and the estimated speed is low, the traffic must have fallen into congestion state. On the contrary, if there are a small number of starting states and braking states and a high speed value, the traffic should be fluent. At last, clients can monitor the traffic condition upon computers or mobile terminal.

In this paper, we analyzed the characterises of acceleration in the two directions and have obtained a few models to estimate the speed of vehicles. Upon calculating the characteristic values of the acceleration, we could match the traveling state with different speed values by the classified model.

In order to obtain the accurate acceleration value in vehicle coordinate system, we put forward a new approach which could obtain the coordinate system relationship between vehicle and smartphone by detecting the starting action. We have designed a kind of data filtering algorithm and could calculate the parameters of their relationship when the acceleration is over  $0.5 \text{ m/s}^2$ , which can be achieved easily.

Comparing with previous work, the main contribution of our work is as follows:

- (1) We propose a new approach to estimate the velocity of vehicle only relying on the accelerometer for the first time and could achieve a satisfying performance.
- (2) A novel method, orientation calculation approach, is put forward, which has proved high accuracy in the driving state. It overcomes the weakness of the compass which is sensitive to the environment.
- (3) We analyze and discover the features of acceleration which could reflect the speed of vehicle.
- (4) Instead of GPS, our approach is more energy-efficient with the accelerometer.

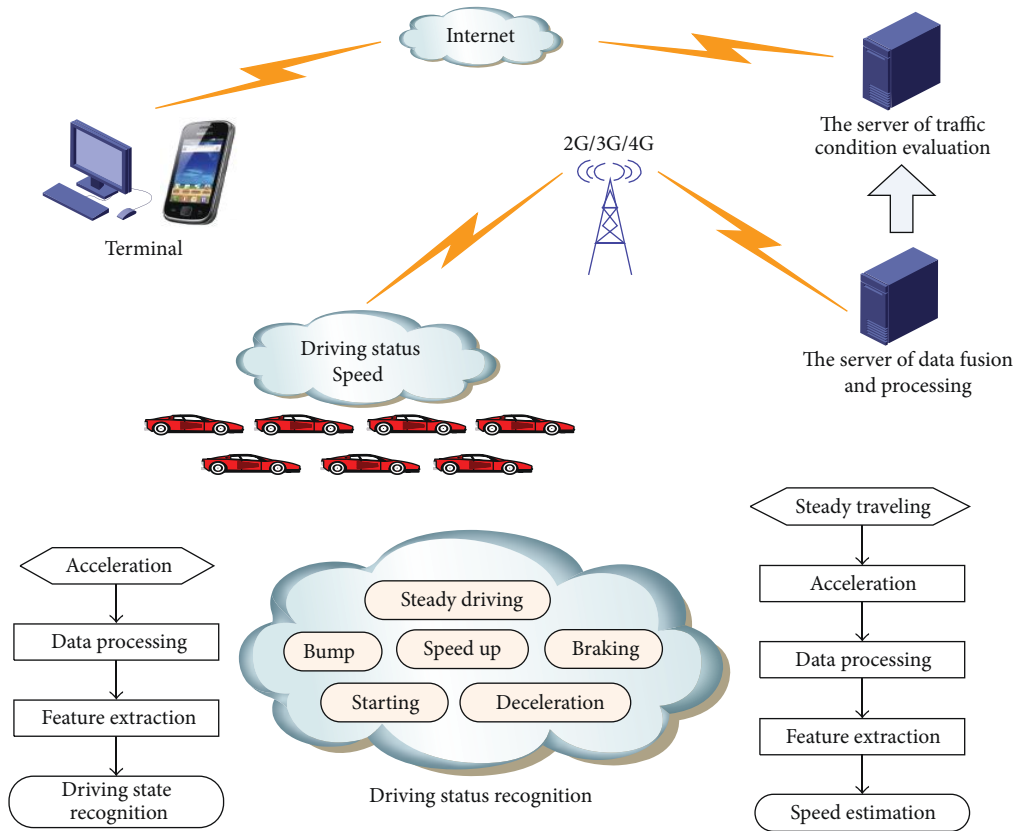


FIGURE 1: System overview.

- (5) Our approach could obtain the speed of vehicle even in the urban canyons.

The remainder of this paper consists of seven parts. Section 3 shows the architecture of our approach. Section 4 introduces the way of coordinate system relationship establishment. Section 5 describes the process of feature extraction, and Section 6 describes the model training process. Section 7 shows the experiment and analyzes the experiment result. Section 8 summarizes the conclusion and future work.

## 2. Related Work

Traffic accidents take place frequently, and the traffic condition has attached more and more attention. In the traditional ways, the traffic information is collected by the fixed facilities on the roadside, such as loop detectors, radars, and cameras which are expensive and also need long-term maintenance. With the increasing occupied percentage of the smartphone in the crowd, kinds of embedded sensors make it possible to obtain the traffic condition using the mobile phone. Some research has reversed its traditional mode, for an instance of the traffic information collection method. Instead of fixed facilities, mobile sensors are applied to collect the traffic information in recent years. Global Positioning System receiver based approaches are widely developed which are embedded in vehicles and smartphones.

**2.1. Traditional Measures.** Taking advantage of single loop detector, Guo proposed an approach to speed estimation in 2009 [3, 4]. The result shows a liner relationship between occupancy and speed. The data collected from two urban regions has proved its feasibility. Porikli and Li proposed a low-latency traffic condition estimation algorithm relying on the MPEG video data in 2004 [4]. He extracted feature information to detect traffic condition. Experiment result showed that the accuracy has reached 95%. Although the traditional measures are mature technology and could obtain a high accuracy, they need abundant fixed equipment and its installing cost and maintaining cost are extensive.

**2.2. Mobile Phone Approach.** In the last few years, the mobile devices are popular in traffic condition estimation. Burke et al. present the concept of participatory sensing in 2006 which collects the information of mobile devices to form interactive, participatory sensor networks and shares the information to the public after gathering and analyzing the collected data [5]. With the development of mobile network and the popularity of smartphones, we have been witnessing the emergence of participatory sensing [5] based on the concept of crowd-sourcing. Because smartphones (e.g., iPhone 5, Galaxy S3) are equipped with rich sensors including GPS, WiFi, cellular radio, accelerometer, and digital compass, we are able to collect massive data directly from users for further data mining

research and data mining application. Several applications have been proposed which apply sensors on smartphones to various areas, such as human activity recognition [6], cyclist experience mapping [7], transportation modes detection [8], and indoor localization [9]. These potential participatory sensing solutions take advantage of high scalability as the number of smartphone users is growing very rapidly.

**2.2.1. Highway Traffic Monitoring Using Mobile Phone.** In 2010 Herrera conducted a field experiment nicknamed Mobile Century and demonstrated that a penetration rate between 2 and 3% in the driver population is enough to get a high accurate traffic estimation [10]. Bar-Gera described a new operational system which measured the traffic speeds and travel times based on the cellular phone information and compared it with magnetic loop detectors. The result shows that both systems got a good performance, and moreover travel time measurements of floating cars offered additional information for the traffic condition [11]. Work et al. provided a Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices in 2008 [12]. However, these systems only concentrate on highway measurements.

**2.2.2. Urban Traffic Monitoring Using Mobile Phone.** Comparing with highway, urban road network is more complex and in a rapidly changing environment. In the last few years, some research have tried several approaches in urban traffic estimation [13–17]. Ferman et al. analyzed the complexity of surface road and freeway and agreed that surface road needed more sophisticated algorithm. He demonstrated that freeway needs 3% penetration rate and surface roads need 5% penetration rate to obtain the relatively accuracy traffic information [18]. Zhu et al. conducted a large-scale experiment for two years collecting speed, latitude, longitude, and timestamp provided by 4000 taxis in Shanghai. He developed a compressive sensing based algorithm which has been proved to be significantly excellent compared to other algorithms, including improved KNN and MSSA. Even though 80% of original data get lost, estimate error can still keep under 20% [13]. In this way we only need 1% penetration rate to estimate traffic condition. However, there are still some problems to be settled. Zhu et al. only considered the downtown roads which have high traffic flow. These roads have high correlation with each other.

Thiagarajan et al. proposed VTrack in 2009, a system for travel time estimation, which applied a hidden Markov model and viterbi algorithm to the map matching schedule and travel time estimation method [14]. He improved the traditional traffic estimation method and verified the relationship between the sampling interval of GPS and accuracy. Based on the vehicle traces obtained from GPS data, Yoon et al. proposed a traffic estimation method of surface streets which is not only effective but also easy to be comprehended. The proposed system analyzes every traffic pattern on all road segments to estimate traffic condition on each road. Experiment result shows that accuracy can be achieved to more than 90 percent after ten or more traversals [17]. Calabrese et al. proposed a new real-time urban monitoring system in 2011. He conducted a large-scale experiment covering most of

Rome and collected lots of real-time urban data. According to the collected data of participant which is anonymous, the designed application can acquire the information of the traffic status of vehicles, pedestrian, and foreigners and especially the traffic condition [16]. Koukoudidis et al. put forward a system based on mobile phones to detect driving mode and location and predict the traffic signal schedule. SignalGuru, a novel software service he designed, shows that the margin of error is 0.66 s, and even for traffic-adaptive traffic signals the margin of error is still within 2.45 s [15].

Although many approaches have been proposed and a high accuracy can also be achieved, all of them are concentrated on the GPS module which is electricity-intensive and the smartphone will run out of power quickly. Besides that, sometimes the smartphone can receive no signal from the outside world or the signal is very weak in relatively enclosed areas. In consequence, the smartphone cannot learn its location and the traffic estimation result goes to wrong.

**2.3. Accelerometer Application.** Kwapisz et al. developed an application which can distinguish six activities, including descending stairs, ascending stairs, jogging, walking, sitting, and standing [6]. He utilized the supervised learning task to realize the detection task taking advantage of the acceleration collected by the smartphone. Moreover, accelerometer can be used to detect many others.

Previous studies concentrate on the GPS data to estimate the traffic information, but we should further conduct research to save energy due to the fact that GPS is a high energy consumption equipment. When GPS device is working, the smartphone will run out of power after several hours. Fortunately, some researches provide us with a new way to obtain the traffic information.

Mohan et al. put forward a system, Nericell, to monitor and estimate traffic information taking advantage of the accelerator, GSM radio GPS sensors, or microphone [19]. Mohan et al. utilized the accelerometer to calculate the relative orientation and realized the goal of braking detection, bump detection, and even the stop-and-go traffic estimation.

In order to obtain more accurate traffic information, Liu et al. also tried to monitor the driving status. He succeeded to classify five driving status, including normal driving, braking, temporary parking, traffic lights waiting, and traffic congestion only taking advantage of the accelerometer [20]. The experiment showed a satisfying result.

If we can learn from previous studies and detect the traffic condition depending on the accelerometer instead of GPS, the smartphone will be more energy-efficient and can still obtain the information while the GPS signal is very weak or even no signal exists in urban canyons, for example, tall buildings and tunnels.

### 3. The Architecture of Traveling Speed Estimation

In this section, we summarize previous related work and analyze the problem. And then we describe the thought of our solution to the problem and show our proposal and scheme. At last, we summarize the framework of our approach.

**3.1. Problem Description.** Some contributions on speed estimation with acceleration have been done [1, 2], which mostly utilizes the integral of acceleration. The experiment results show that this method can obtain a good performance only under straight driving state and cannot persist for a long period. The key problem is that the acceleration owns frequent fluctuation and cannot reflect the variation of speed strictly. So the fact may bring a serious deviation that the traveling speed is estimated by the integral of acceleration.

If we could find the characteristics which could reflect the speed value, we could evaluate its speed value. As we all know, the acceleration in the advancing direction could reflect the variance of the speed value, and the acceleration in the vertical direction could reflect the fluctuation of road condition, which will be more frequent as the speed value increases. So the problem is how to find the characteristics on both advancing and vertical directions and own the excellent classified model.

How we can acquire the acceleration in the advancing direction and vertical direction accurately is more essential. As we all know, the smartphone may be placed casually and there are still kinds of starting modes of vehicle. If we would like to estimate the coordinate system relationship of smartphone and vehicle by starting action, another problem is how to design an effective filtering algorithm and select proper parameters to remove the invalid states.

**3.2. Coordinate System Relationship.** In this paper, we propose a new approach to establish the coordinate system relationship by starting action detection. According to the theory of relationship establishment [19], we only select the acceleration value of partial straight starting states to calculate the values of parameters, since other states may lead to huge error. Apparently, nonstraight starting state is incongruent with our theory. Reversing may lead to completely opposite result. Low starting states own no obvious characteristics and may lead to huge error.

Our approach is shown in Figure 2 (dotted portion represents the process of parameter regulation). Firstly, we should recognize whether the driving state is starting action. Secondly, we would do some work on straight driving recognition. Thirdly, the effectiveness of starting action will be considered with care. States of reversing and very slow start will be excluded. At last, we could calculate the values of parameters and learn their coordinate relationship. In order to improve its accuracy, we take advantage of GPS module and compass embedded in smartphone to help us recognize the legal starting action. Besides, more work has also been finished about parameter optimization. The detail will be discussed in Section 4.

**3.3. The Process of Speed Estimation.** In this paper, we put forward a new approach to estimate the speed value with acceleration in advancing direction and vertical direction. We try to make use of classification model to evaluate the speed value. There are two phases of speed value evaluation, including model training process and speed value evaluation with the prediction model. The detail of our approach is described as follows.

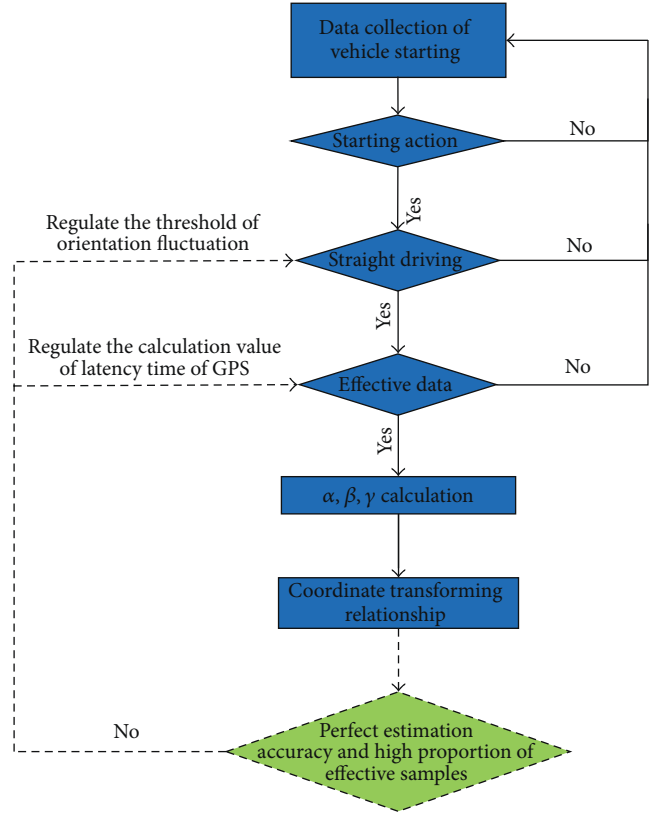


FIGURE 2: Orientation calculation algorithm.

In the model training process, firstly, we should obtain the coordinate transformation relationship between smartphone and vehicle and then transform the acceleration value from the coordinate system of smartphone to the coordinate system of vehicle. Secondly, we calculate the values of various characteristics, including time-domain characteristics, frequency-spectrum characteristics, and power-spectrum characteristics in advancing direction and vertical direction. Thirdly, we analyze these characteristics by ourselves and select the excellent characteristics which could reflect the speed value. Fourthly, the scheme of model training is formulated, and, according to the scheme, different characteristics are filtrated from the excellent characteristics and join into different components which could reflect the partial speed value better. At last, we will do some researches on the model training field and regulate the parameters, scheme of model training, and then select the proper characteristics.

In the process of speed value evaluation, we obtain the acceleration in advancing direction and vertical direction as mentioned above firstly. And then we calculate the proper characteristic values and classify them into different components in accordance with the classification scheme. At last, with the classified characteristics, we could get the evaluated speed value with the classification models.

The architecture of the approach is shown in Figure 3. And the detail of characteristic selection and the process of model training will be discussed in Sections 5 and 6.

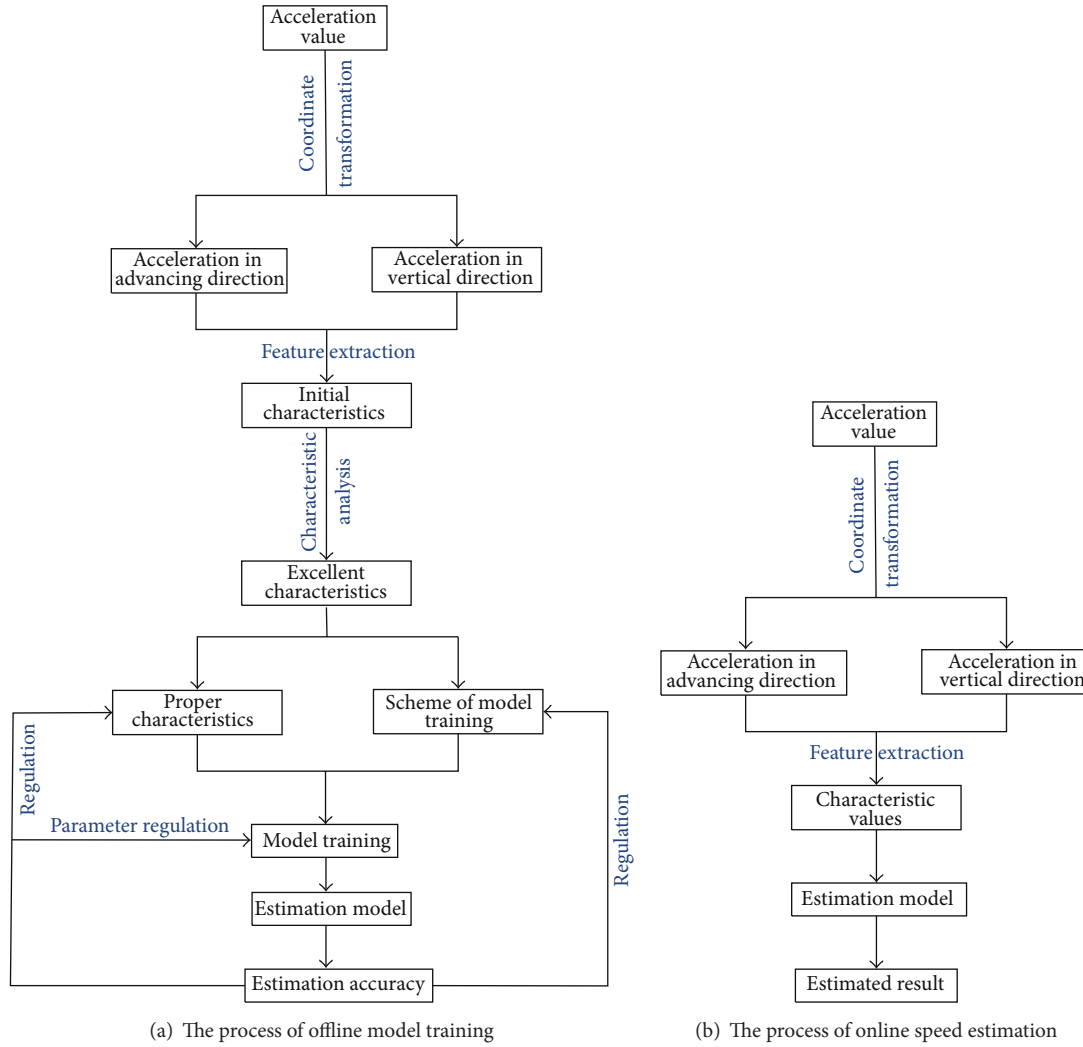


FIGURE 3: The architecture of traveling speed estimation.

#### 4. Coordinate System Relationship Establishment

In this section, we mainly describe how can we obtain coordinate relationship of our smartphone and the car we are driving.

As we all know, the orientation sensor data is susceptible to the electromagnetic environment where the TV tower, radio station, radar station, high tension line, and transformer substation are located. In addition, the electromagnetic equipment in the car can also bring an impact on the magnetic field. Therefore, the orientation sensor would suffer a serious error. Based on this consideration, we need other methods to calculate the orientation.

Some researches in the field of coordinate system relationship between smartphone and vehicle have been done. Mohan et al. utilized the accelerometer to calculate the relative orientation and realize the goal of braking detection, bump detection, and even the stop-and-go traffic estimation

[19]. Without filtering the data of braking action strictly, it may lead to relatively large calculation error.

In this section, taking advantage of the GPS, acceleration, and orientation sensor data in the starting action, we propose a high-precision approach of relative orientation calculation with vehicle. Comparing with the method Mohan proposed, our approach has a low threshold, and a starting action with slow acceleration can also realize the relative orientation calculation. Besides, our approach is more robust and accurate. Different with him, we have done lots of work in the data processing aspect. As Figure 2 shows, this section contains five procedures, including coordinate system relationship, starting action recognition, straight driving identification, filtering effective data, and parameter calculation.

**4.1. Coordinate System Relationship.** In this paper, we asked the taxi driver to help us collect the information with our smartphone and do some research on the coordinate relationship between taxi and our experiment smartphone.



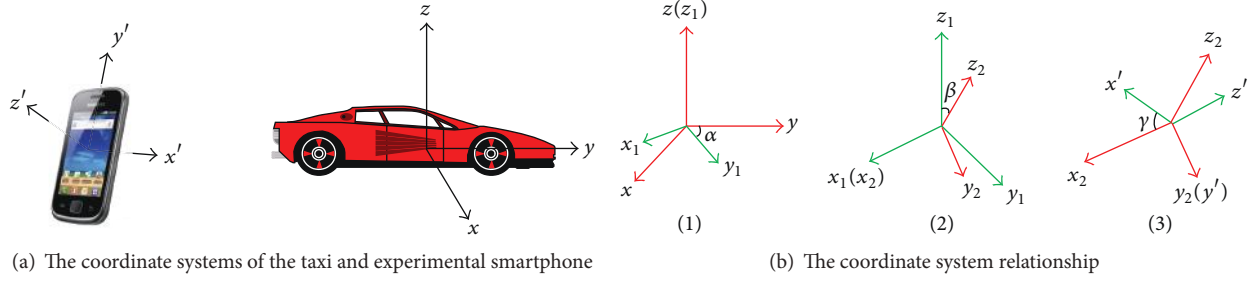


FIGURE 4: The coordinate systems and their relationship.

Figure 4 shows their coordinate systems and presents their relationship. Firstly, the taxi coordinate system  $(x, y, z)$  rotates around the  $z$ -axis by  $\alpha$  angle clockwise and turns to  $(x_1, y_1, z_1)$  coordinate system. Secondly, the  $(x_1, y_1, z_1)$  coordinate system rotates around the  $x_1$ -axis by  $\beta$  angle clockwise and turns to  $(x_2, y_2, z_2)$  coordinate system. Finally, the  $(x_2, y_2, z_2)$  coordinate system rotates around the  $y_2$ -axis by  $\gamma$  angle clockwise and turns to  $(x, y, z)$  coordinate system and coincides to the smartphone coordinate system. Therefore, we could get the coordinate transformation relationship between vehicle coordinate system  $(x, y, z)$  and smartphone coordinate system  $(x', y', z')$ . It is shown as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (1)$$

$$* \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ \sin \beta \sin \gamma & \cos \beta & \sin \beta \cos \gamma \\ \cos \beta \sin \gamma & -\sin \beta & \cos \beta \cos \gamma \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}.$$

**4.2. Starting Action Recognition.** As everyone knows, there is a sharp acceleration during vehicle starting, and many kinds of cars can reach 100 km/h within 10 seconds. Generally speaking, the average value of the acceleration is  $2.78 \text{ m/s}^2$  in this period. In our experiment, the taxi driver has a good driving habit and drives the taxi in a very smooth state. But even so, the acceleration is overwhelming above  $1 \text{ m/s}^2$  in this period. In our orientation calculation approach, we select  $0.5 \text{ m/s}^2$  as a threshold of starting action. In other words, if the speed ranges from 0 to 1.5 m/s in three seconds, we will classify this period to starting state. The selected threshold is much less than the general acceleration in this state.

**4.3. Straight Driving Recognition.** In this section, we analyze the orientation sensor data of starting state to realize the straight driving recognition. The value of orientation sensor data the experimental smartphone collected is represented by the symbol of  $(w_{x'}, w_{y'}, w_{z'})$ .  $w_{x'}$  represents the angle of the coordinate system rotating around the  $x'$ -axis from the horizontal plane anticlockwise.  $w_{y'}$  represents the angle of the coordinate system rotating around the  $y'$ -axis from the horizontal plane anticlockwise.  $w_{z'}$  represents the angle of the coordinate system rotating around the  $z'$ -axis from

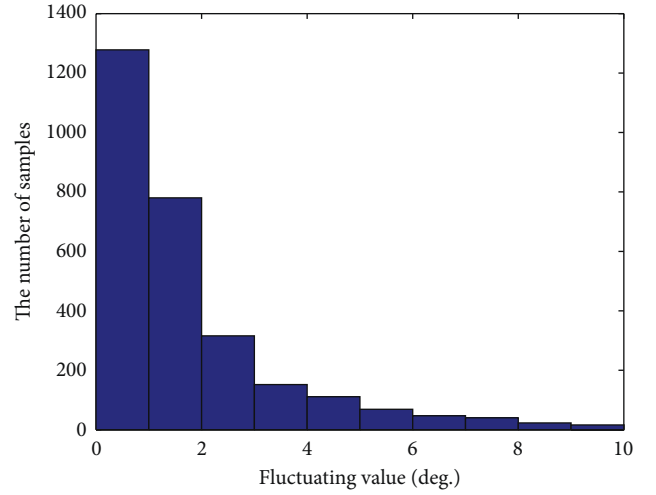


FIGURE 5: The relationship between the number of samples and fluctuating value.

north clockwise. All kinds of statuses of taxi parking exist in the experiment, including roadside parking, waiting at red lights, and parking in parking lot. Therefore, all kinds of starting states also exist as mentioned in Section 7.2. Based on this consideration, we design an algorithm to recognize the straight driving status and discard the invalid data. We perform statistical analysis on the  $w_{z'}$ , which could reflect the  $\alpha$  value. Figure 5 shows the relationship between the number of samples and the fluctuating value, which represents the difference between the max value and the min value of  $w_{z'}$  in the starting state. The picture can reflect their relationship when the fluctuating value is below 10 degrees. Obviously, the samples are opposite with straight driving state when their fluctuating value is over 10 degrees, and we obtain 2832 samples whose fluctuating value is below 10 degrees in total. According to the picture, their connection is much similar with the parabola. In order to obtain abundant samples to estimate the relative orientation with the taxi, we select 4 degrees as the threshold of the fluctuating value, which represents a low fluctuating value. In other words, if the fluctuating value is lower than 4 degrees in the starting state, we would regard the starting state as straight driving.

**4.4. Effective Data Recognition.** Apart from Sections 4.2 and 4.3, we also need other algorithms to filter the collected

information. There are still other invalid states we have not excluded. If the taxi driver backs the car at first and then drives away straightly, this state may be legal in Sections 4.2 and 4.3 algorithm; however, it is illegal in fact and we would draw an incorrect conclusion. To avoid this, we measured the latency time of the GPS module in the starting period and design another algorithm. According to the statistic, the acceleration value is below  $0.2 \text{ m/s}^2$  when the taxi is in the stationary state. As mentioned in Section 4.2, most of the acceleration in the starting state is above  $1 \text{ m/s}^2$ . In this section we select  $0.2 \text{ m/s}^2$  as the threshold of the starting state, and, in other words, if the acceleration is above  $0.2 \text{ m/s}^2$  at this moment and the taxi keeps in stationary state prior to this moment, we would look at this moment as the real time the taxi starts. In addition, it is defined as the detecting time, when the speed of the taxi changes from zero to nonzero, which is measured by the GPS module. The difference between the real time and the detecting time is called the latency time of the GPS module in the starting period.

Unfortunately, the latency time does not keep consistency in the starting experiments. In order to exclude the backing state and filter more illegal starting states, we analyzed the starting data after starting action recognition and straight driving recognition and tried to find out the latency time. In the case of selecting 4 degrees as the threshold of the fluctuating value, the relationship between the latency time and the calculation result is shown in Figure 6. In order to obtain a satisfied accuracy and effective samples as many as possible, we select 3.5 seconds as the latency time of the GPS module. This means that if the latency time we detect is larger than 3.5 seconds, we would discard the sample, and otherwise we would regard it as an effective sample.

After filtering the collected data in the 4732 starting experiments with the algorithm mentioned by Sections 4.2, 4.3, and 4.4, we obtained 2117 effective samples in total.

**4.5. Parameter Calculation.** Taking advantage of the effective data, we calculate the relationship between the vehicle coordinate system  $(x, y, z)$  and the smartphone coordinate system  $(x', y', z')$ , which is represented by the symbol  $\alpha$ ,  $\beta$  and  $\gamma$ . The value of  $\beta$  and  $\gamma$  can be calculated by  $w_{z'}$  and  $w_{x'}$  which are collected by the orientation sensor directly. We calculate the average of the  $w_{x'}$  value as the result of the  $\beta$  value and the average of the  $w_{y'}$  as the result of the  $\gamma$  value. And then we can calculate the  $\alpha$  value, making use of  $\beta$  value,  $\gamma$  value, and the acceleration during the start period. The acceleration detected by the smartphone is denoted as  $(a_{x'}, a_{y'}, a_{z'})$ , which represents the taxi acceleration in the mobile coordinate system. The  $\alpha$  value can be obtained as follows:

$$\begin{aligned} \alpha &= \arctan \frac{a_{y1}}{a_{x1}} + 90, & a_{x1} < 0 \\ \alpha &= \arctan \frac{a_{y1}}{a_{x1}} + 270, & a_{x1} > 0. \end{aligned} \quad (2)$$

$a_{x1}$  represents the component of the acceleration along the  $x_1$ -axis in the  $(x_1, y_1, z_1)$  coordinate system;  $a_{y1}$  represents the component of the acceleration along the  $y_1$ -axis in the

$(x_1, y_1, z_1)$  coordinate system. The value of  $a_{x1}$  and  $a_{y1}$  is calculated as follows:

$$\begin{bmatrix} a_{x1} \\ a_{y1} \\ a_{z1} \end{bmatrix} = \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ \sin \beta \sin \gamma & \cos \beta & \sin \beta \cos \gamma \\ \cos \beta \sin \gamma & -\sin \beta & \cos \beta \cos \gamma \end{bmatrix} \begin{bmatrix} \bar{a}_{x'} \\ \bar{a}_{y'} \\ \bar{a}_{z'} \end{bmatrix}. \quad (3)$$

$\bar{a}_{x'}$  represents the average of  $a_{x'}$  (the acceleration along the  $x'$ -axis) in the starting period;  $\bar{a}_{y'}$  represents the average of  $a_{y'}$  (the acceleration along the  $y'$ -axis) in the starting period;  $\bar{a}_{z'}$  represents the average of  $a_{z'}$  (the acceleration along the  $z'$ -axis) in the starting period.

## 5. The Process of Feature Extraction

As mentioned in Section 3.3, we will describe the process of feature extraction in this section. In practical, the feature extraction process and model training process are interrelated.

**5.1. Raw Data Acquisition.** The goal of our approach is to evaluate the traffic congestion scale. As shown in Figure 1, the traffic information can be concluded from the driving status and speed value. As analyzed above, we take measure of characteristics of acceleration in advancing direction and vertical direction of vehicle to evaluate the speed value. It is a common sense that the characteristics are always combined with samples. Different samples may have distinct characteristics. In another word, we need to classify the traffic information into different samples and estimate the speed value by extracting characteristics of these samples. In this paper, we only analyze the normal driving status and estimate its speed by the acceleration value.

We classify the driving status into five states, including starting, braking, acceleration, deceleration, and normal driving, and we select the time-window of samples as ten seconds. Several reasons account for this phenomenon. Firstly, a short time-window could improve the efficiency of traffic congestion evaluation. Secondly, a short time-window could increase the effectiveness of traffic information; for example, if the time-window is much longer than the period of starting, braking, acceleration, and deceleration, these samples including the four driving statuses will be regarded as illegal samples. At last, ten seconds is enough for these four activities.

In the model training process, we define the five states with the speed value collected by GPS module. The threshold of the acceleration value in the period of acceleration and deceleration is defined as  $0.5 \text{ m/s}^2$ , which could be achieved in these two states with ease on the basis of statistics. If the difference between the speed values of initial moment and final moment is lower than  $-5 \text{ m/s}$ , this sample will be regarded as the process of deceleration. Otherwise, it will be regarded as normal driving status.

**5.2. Feature Extraction.** As analyzed above,  $a_y$  could reflect the variation of speed value and  $a_z$  could also reflect the speed value to some extent. Once the values of  $a_y$  and  $a_z$  are known, we could extract their characteristics. If these characteristics perform well, we could evaluate the speed value accurately.

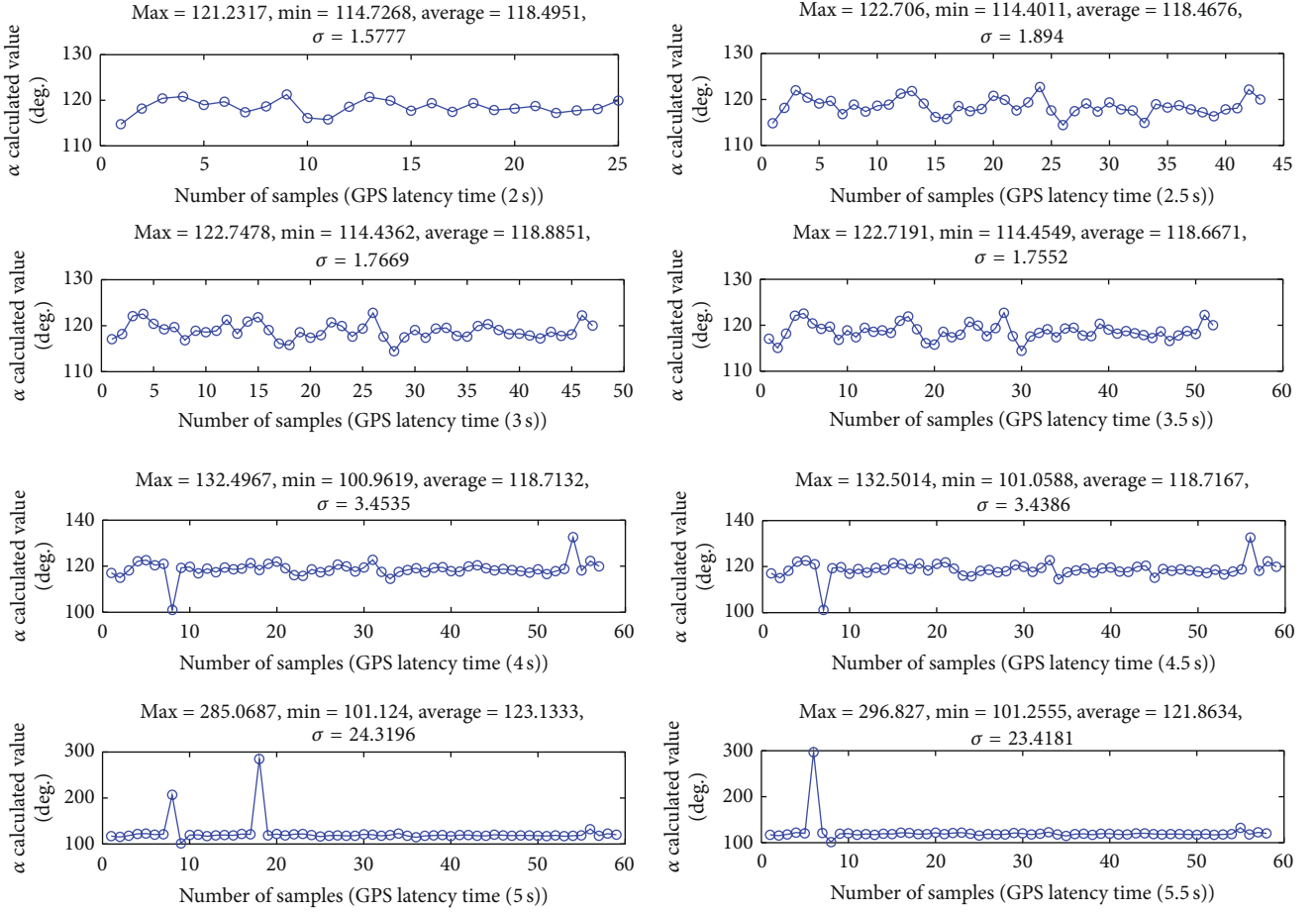


FIGURE 6: The relationship between the latency time and the calculation result.

We analyze both the characteristics of  $a_y$  and  $a_z$  in time-domain and frequency-domain, including frequency spectrum and power spectrum. The characteristics in time-domain contains eight common features, including “mean,” “standard deviation,” “average deviation,” “skewness,” “kurtosis,” “RMS amplitude,” “max value,” and “min value,” and distribution of the acceleration value. According to the principle of  $3\sigma$ , the value of  $a_y$  is divided into 18 intervals and the value of  $a_z$  is divided into 22 intervals.

We conduct the experiment and collect acceleration value in a frequency of 20 Hz. Upon Fourier transform, we could obtain its frequency spectrum and power spectrum. Frequency spectrum contains eleven statistic characteristics, including mean square frequency (MSF), frequency average value (FAV), frequency median value (FMV), max value of amplitude (MVA), average value of amplitude (AVA), deviation of amplitude (DA), frequency mean error (FME), frequency deviation (FD), distribution of frequency (DF), distribution of amplitude (DIA), and ten frequencies whose amplitude is at the front (TF). Most characteristics are shown in Table 1.  $x(i)$  represents the frequency value and  $y(i)$  represents the amplitude combined with  $x(i)$  in the frequency diagram. DIA represents the number of samples falling into

the intervals ( $[0 : 0.05 : 0.5] \cup [0.5 : 0.1 : 1]$ ). TF represents ten values of  $x(i)$  where  $y(i)$  are larger than others.

The power spectrum contains nine statistic characteristics, including mean square frequency (MSF), frequency average value (FAV), frequency median value (FMV), max value of amplitude (MVA), average value of amplitude (AVA), deviation of amplitude (DA), frequency mean error (FME), frequency deviation (FD), and distribution of frequency (DF), which are the same as these characteristics with frequency spectrum.

We extract all these kinds of characteristics of  $a_y$  and  $a_z$  and obtain 178 eigenvalues in total, which are effective or invalid characteristics.

We classify ten-second window samples into 13 states by speed value. The integers from one to twelve are selected as the thresholds and the intervals divided by these thresholds are the statistical intervals of these samples. These characteristics are obtained from the 13 states.

**5.3. Effective Feature Selection.** We obtain large amount of characteristics in Section 5.2. However, some of these characteristics are invalid and could not reflect the traveling speed. In this section, we analyze the properties of these



TABLE 1: Characteristics in frequency-domain.

Feature name	Description
MSF	$f_{ms} = \frac{\sum_{i=1}^N x(i)^2 y(i)}{\sum_{i=1}^N y(i)}$
FAV	$f_a = \frac{\sum_{i=1}^N x(i) y(i)}{\sum_{i=1}^N y(i)}$
FMV	$f_{mv} = x(k) \mid \sum_{i=1}^k y(i) = \sum_{i=k}^N y(i)$
MVA	$f_m = \text{Max}(y(i)) \mid i = 1 \text{ to } N$
AVA	$A_m = \frac{1}{N} \sum_{i=1}^N y(i)$
DA	$A_a = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (y(i) - A_m)^2}$
FME	$f_{me} = \frac{1}{N} \sum_{i=1}^N  x(i) - f_a  * y(i)$
FD	$f_e = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x(i) - f_a)^2 * y(i)}$
DF	$f_d(i) = \sum_{i=(i-1)*N/10+1}^{i*N/10} y(i) \mid i = 1 : 1 : 10$

characteristics and select the effective characteristics. This section consists of two parts, initial characteristic filtering and proper characteristic selection with the scheme of model training.

In the process of initial characteristic filtering, we analyze these characteristics by manual work. The average values of these characteristics from different samples are compared and analyzed by manual work. We analyze these characteristics from two aspects, including overall trend and partial features.

The examples of statistical characteristics in overall trend are shown in Figure 7. If the average value of these characteristics has shown some regularity that the average value increases or decreases with traveling speed in tendency, the character will be regarded as legal feature; otherwise, it will be regarded as illegal feature. The characteristics of DIA[2] and DIA[3] of  $a_y$  value are legal features, and the characteristics of DIA[1] and DIA[4] of  $a_y$  value are illegal features. We analyze all these characteristics of  $a_y$  and  $a_z$  and obtain 63 characteristics, including 12 characteristics of  $a_y$  and 51 characteristics of  $a_z$ .

According to the traveling speed, we also analyze these characteristics in detail. We summarize features which perform variation apparently with traveling states mentioned in Section 5.2. We classify the traveling state into four parts. The first part consists of these samples whose average speeds range from 0 m/s to 4 m/s. The second one includes samples whose average speeds range from 4 m/s to 8 m/s. The third one contains samples whose average speeds range from 8 m/s to 12 m/s. The last one is made up with samples whose average speed is above 12 m/s.

On the one hand, the characteristics of the first four parts are concluded, which could reflect the speed variation in the partial scope. We obtain 29 characteristics, including 6 of  $a_y$  and 23 of  $a_z$ , which show an apparent variation when the average speed ranges in the scope of the first part. Based on this thought, we get 51 characteristics, including 10 of  $a_y$  and 41 of  $a_z$ , which could reflect the second part evidently. Similarly, we get 7 characteristics, including 3 of  $a_y$  and 4 of  $a_z$ , which could reflect the third part, and 9 characteristics, including 8 of  $a_y$  and 1 of  $a_z$ , which could reflect the fourth part. We could draw a conclusion with ease that the characteristics of the first two parts are far more than these characteristics in the last two parts. In another word, the first two parts can obtain a more accurate performance.

On the other hand, we analyze the characteristics combined with the relationship of different parts. We distribute the adjacent parts with characteristics selected from the 178, which could reflect the difference of the adjacent parts. There are 64 characteristics which could distribute the first two parts, including 16 of  $a_y$  and 48 of  $a_z$ . The difference between the second part and third one is also evident, and we obtain 49 characteristics in total which could reflect its difference, including 11 of  $a_y$  and 38 of  $a_z$ . Unfortunately, as shown in Figure 7, the difference between the last two parts is very week. We could only extract 10 characteristics of  $a_y$ , which could distribute the difference between the third part and the fourth one. Besides, comparing with the characteristics between the first part, second part, and the third part, these characteristics have little difference.

In summary, these characteristics could distribute the first three parts evidently, but the difference between the last two parts is very week.

## 6. The Process of Model Training

In this section, we describe the model training process. With the trained model, we could evaluate the traveling speed with the acceleration. As described in the last paragraph in Section 5.2, we divide the samples into 13 states.

This section consists of two aspects, including scheme formulation and the process of model training.

**6.1. Scheme Selection.** As analyzed in Section 5.3, the characteristics of the first three parts are distinct obviously, and the characteristics of the last two parts are similar. Therefore, the characteristics in the first two parts are easy to be distinguished, and the characteristics in the third part and the fourth part are hard to be distinguished. Based on this analysis, our scheme consists of two parts, including general classification and specific classification.

The process of general classification is shown in Figure 8. D1 and D2 reflect the global characteristics, and E1, E2, and E3 reflect the partial characteristics. According to the fact that the characteristics of the four parts have obvious distinction, especially in the first three parts, we design Decision Tree Architecture to decrease the calculated amount in the process of four parts classification. Firstly, we pick up the first two parts by the characteristics mentioned in the third paragraph of Section 5.3 as shown in subgraph

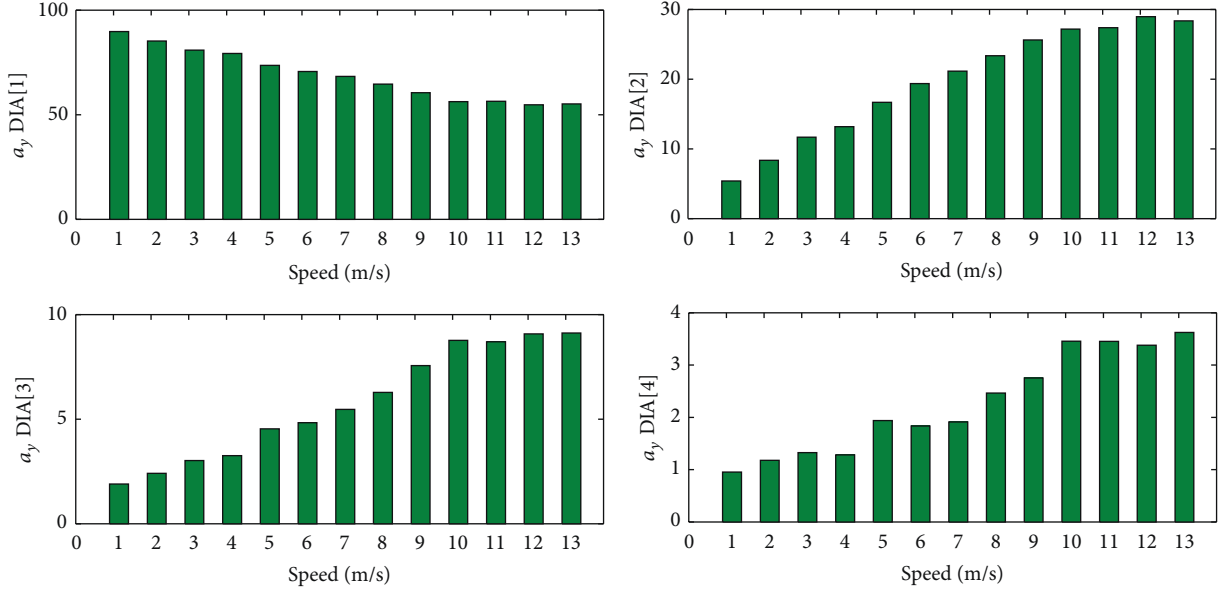


FIGURE 7: Initial characteristic filtering.

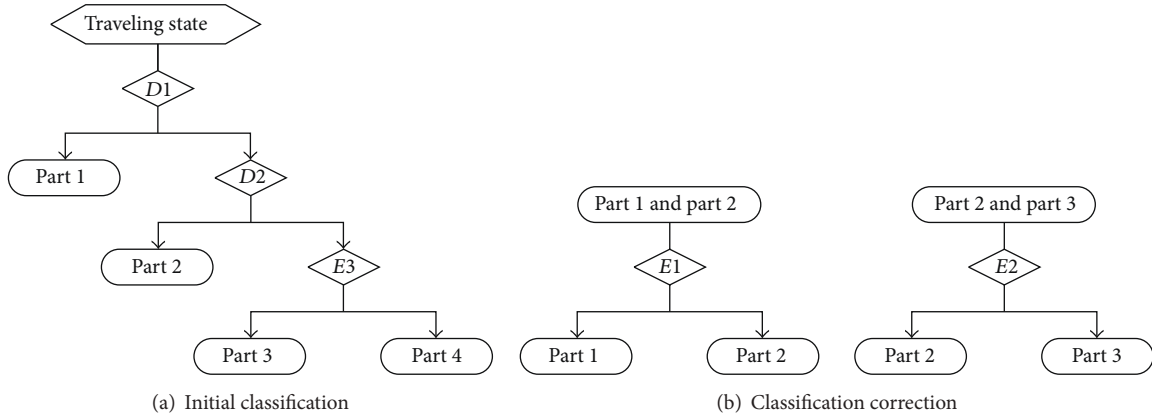


FIGURE 8: Model training scheme.

(a). Secondly, we distinguish the last two parts with these characteristics explicitly mentioned in the sixth paragraph of Section 5.3. At last, as shown in subgraph (b), we improve the evaluated results with these characteristics.

In the specific classification process, we take advantage of voting method to obtain the specific evaluated result with the characteristics mentioned in fifth paragraph of Section 5.3 and get an accurate evaluated result.

**6.2. Decision Tree for General Classification.** The general algorithms of model training contain Decision Tree, Logistic Regression, and Neural Network. As we all know, the significance of Decision Tree parameters is comprehensible. Besides, the calculated amount is relatively smaller. However, the classified result performs worse when the number of category increases. Therefore, sometime this algorithm owns the phenomenon of overfitting and has no good generalization performance. Logistic Regression is only suitable for liner problem and can lead to overfitting with ease. Most

unfortunately, it acquires poor accuracy frequently. Neural Network consists of many neurons; it has a complex structure and could achieve a good performance. It can solve kinds of problems, including liner problem and nonlinear problem.

As one of Neural Network Learning Algorithms, Support Vector Machine (SVM) algorithm has a perfect generalization performance and could achieve good accuracy. In this section, we take advantage of SVM algorithm to train the classified model.

In the process of initial classification, we have trained the model with every characteristic mentioned in Section 6.1. According to the fact that the characteristics in the last two parts are similar, the characteristics, which could reflect the difference between the second part and the third part, are regarded as the characteristics which could reflect the difference between the second part and last two parts. The evaluated results are shown in Tables 2 and 4.

It is obvious that the first part can be distinguished easily, and the recognition rate of second part achieves a moderate

TABLE 2: Initial classification.

Procedure	Feature name	Precision
D1	Comprehensive (3)	92.70%
	Comprehensive (5)	92.80%
	Voting method (5)	92.80%
D2	Comprehensive (3)	74.16%
	Comprehensive (5)	73.33%
	Voting method (5)	74.52%
E3	Comprehensive (3)	58.31%
	Comprehensive (5)	58.00%
	Voting method (5)	59.49%

TABLE 3: The performance of classification correction.

	Real part			
	Part 1	Part 2	Part 3	Part 4
Estimated part (I)				
Part 1	98	6	1	0
Part 2	51	95	32	38
Part 3	14	127	171	134
Part 4	1	18	81	133
Estimated part (II)				
Part 1	103	9	2	0
Part 2	56	128	57	43
Part 3	4	91	145	129
Part 4	1	18	81	133

performance. However, the recognition rate of last two parts make a poor performance. We evaluate the performance of several approaches; “comprehensive (3)” represents that we evaluate the speed with three characteristics which achieve best results and “comprehensive (5)” is similar to “comprehensive (3)”; “voting method (5)” means that we obtain the estimated speed with the five characteristics voting for their opinions.

Judging from Tables 2 and 4, the three approaches show similar accuracy. Because of its low calculated amount, we select the approach of “comprehensive (3).” Instead of only utilizing the first one, three characteristics could increase the robustness of the classified model.

Table 3 shows the distribution of classification correction process. “Estimated part (I)” represents the evaluated result without classification correction, and “Estimated part (II)” represents the evaluated result after classification correction. “Real part” represents the real value evaluated by GPS. There are evident improvement effects, especially when traveling with low speed.

**6.3. Voting for Secondary Classification.** In this section, we divide every part, except for the last one, into four intervals as mentioned in the last paragraph in Section 5.2 and obtain 13 states in total. In the process of specific classification, as in Section 6.2, we also sort the selected characteristics by the evaluated precision and take the approach of “comprehensive (3)” to distinguish different intervals in every part.

TABLE 4: Classification correction.

E1	Comprehensive (3)	74.16%
	Comprehensive (5)	83.66%
	Voting method (5)	83.90%
E2	Comprehensive (3)	68.17%
	Comprehensive (5)	67.61%
	voting method (5)	67.23%

As we all know, the multiclass classification methods of SVM contain three approaches, including “one-against-all,” “one-against-one,” and DAGSVM [21]. The method of “one-against-all” could lead to large offset because of unequal samples and get a poor performance. “One-against-one” and DAG methods could achieve practical effectiveness and a good performance. However, the DAG method acquires distinct variance between the first state and the last state. In this section, we utilize “one-against-one” method to estimate the traveling speed.

In order to improve the accuracy, we have trained six models in every part, which could reflect the relationship of every two states. With six models voting for the results, the evaluated result will be more accurate than other methods. The estimated performance in every part is shown in Tables 8, 6, and 7. “(F)” represents the characteristics of frequency spectrum and “(P)” represents the characteristics of power spectra. Apparently, the classification accuracy in the first part achieves a good performance. The element proportion on main diagonal also exceeds 40% in part 2. However, the classification accuracy in the third part makes a poor performance because of its similar characteristics.

## 7. Experiment

In order to verify the feasibility of our proposal, we need to conduct experiments to obtain the traveling information. In this section, we design two experiments in total for evaluating the precision and energy consumption of our scheme.

### 7.1. Experiment One

**Goal: Obtaining Rich Data for Our Future Supervised Learning Task and the Model Accuracy Evaluation.** In order to collect rich data for our future supervised learning task and the model accuracy evaluation, an abundant of traffic information needs to be collected. We cooperate with a taxi company in Hangzhou, employing the company’s taxi for the traffic information collection. The taxi driver placed the mobile phone at the fixed position before working and took the smartphone away and turned the application off after working. Figure 9 shows the located position of the experimental smartphone in the car. In order not to affect the taxi drivers when they are driving, the entire collection process would be done automatically without driver intervention. Meanwhile, we did not constrain the route and time of the driver traveling, and he could drive the taxi as usual. We collected the traffic information for more than one month and got 46164000 data sets in total. They nearly covered



FIGURE 9: The environment of data collection.

every street of Hangzhou and contain every traffic state, for example, all kinds of congestion scale data, rest state, roadside parking, waiting state.

The traffic information collection was controlled by an application we had designed that is executed on the phones (LG Nexus 4). The application permitted us to control which sensor data (e.g., gps, accelerometer) was collected and how frequently it was collected. We collected the latitude, longitude, and speed measured by the GPS module of the smartphone every 1 second and sampled the accelerometer data, the orientation sensor data, and timestamp every 50 ms.

## 7.2. Experiment Two

*Goal: Obtaining Rich Data to Optimize the Parameters of the Orientation Calculation Algorithm and Check Its Estimated Accuracy.* As mentioned in Section 4, we evaluate the relative orientation and posture of the smartphone by detecting the vehicle starting action. In order to optimize the parameters of the orientation calculation algorithm and check its estimated accuracy, we need to collect plenty of data in the starting period and all kinds of starting status. Similar to Section 7.1, we cooperate with a taxi company in Hangzhou, employing the taxi for data collection. We conduct all kinds of starting experiments, including starting with straight driving, starting with turning around, starting with turning a corner, starting with reversing, starting with overtaking, starting with slow acceleration, starting with sharp acceleration, and starting as normally. In total, we conducted 4732 starting experiments. The data we collected consists of the acceleration, speed, and orientation value in the starting action.

## 7.3. Experiment Three

*Goal: Evaluating the Energy-Saving Effect of Our Approach with Other Techniques.* In order to evaluate the energy-saving effect of our approach with other techniques, we should record the energy consumption of GPS module and accelerometer. According to the latest research, in the case of traffic information estimation, GPS module collects the smartphone location at a period of 20 s at least. However, our approach relies on the accelerometer to obtain the evaluated result. In order to simulate the real experimental, we ask the driver to help us conduct the experiment. We fixed the

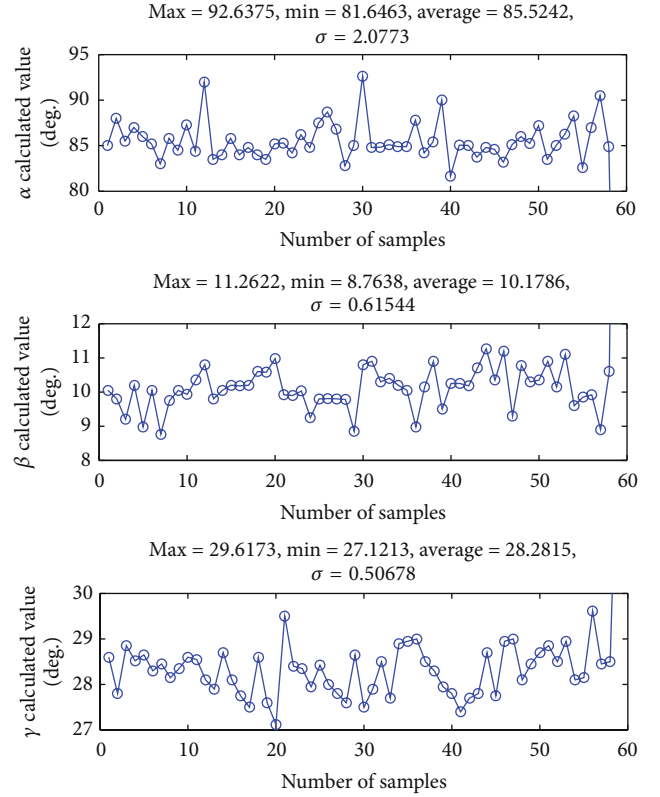


FIGURE 10: Estimated results for the coordinate parameters.

smartphone at the front window and she drove the taxi normally. We conduct three experiments with the same smartphone, including GPS module energy consumption, accelerometer energy consumption, and standby energy consumption. The GPS module collected location data at a period of 20 s, and once the location is known the GPS module will be closed in the first experiment. Accelerometer are kept on all the time in the second experiment and collected acceleration value at 20 Hz.

**7.4. Experimental Results.** In this section, we describe our experiment results, including the parameter accuracy of coordinate system, the classified accuracy, and energy consumption.

**7.4.1. Coordinate System Relationship.** As described in Figure 10, the  $x$ -coordinate represents the number of our calculated results by detecting the multiple starting states in one day, and the  $y$ -coordinate represents the calculated values of the parameters. The accuracy of the calculated value of the parameters acquired a satisfied result. The fluctuation of  $\alpha$  evaluated value is lower than 11 degrees and the standard deviation is about 2 degrees. Apparently, the  $\beta$  and  $\gamma$  values are more stable, and their standard deviations are all below 0.7 degrees.

Figure 11 shows the performance of previous research; the variance of  $\phi_{pre}$  value is more than 1 degree; however, the variance of  $\gamma$  value is only 0.5 degrees. Corresponding to  $\alpha$

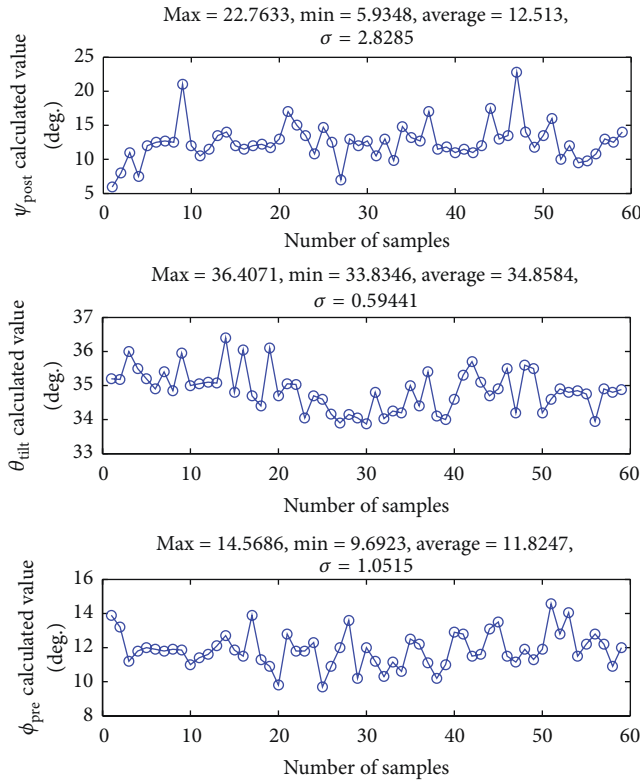


FIGURE 11: Estimated results for previous research.

value, the  $\psi_{\text{post}}$  has a larger variance, which is more than 2.8 degrees. And particularly the difference between max value and min value is 17 degrees which is 1.53 times the difference of  $\alpha$  value. Comparing with  $\alpha$  value, it has a serious error. In a word, our approach makes a good performance.

Figure 12 shows the acceleration value after coordinate transformation, the real acceleration value, and the acceleration value detected by the smartphone. The  $x$ -axis value represents time in all subgraphs. The  $y$ -axis value represents the acceleration value in the subgraph except for the first subgraph, and the  $y$ -axis value is the speed of the vehicle in the first subgraph. This picture describes the speed and acceleration in the starting period. The green line represents the acceleration value detected by the smartphone directly. The blue line represents the calculated acceleration value in the coordinate system of the vehicle, and the red line represents the real acceleration value in the coordinate system of the vehicle.

As described in the picture, the result after coordinate transformation is very similar with the real acceleration value. The  $y$ -axis value is nearly coincident with the true acceleration value in the driving direction of the vehicle.

**7.4.2. Classified Accuracy and Energy Consumption.** According to Section 5.1, after removing the data when vehicle remains stationary, we classify the traveling information into five states and obtain 11264 samples, including 7688 samples of normal driving. We select 6000 samples to train the

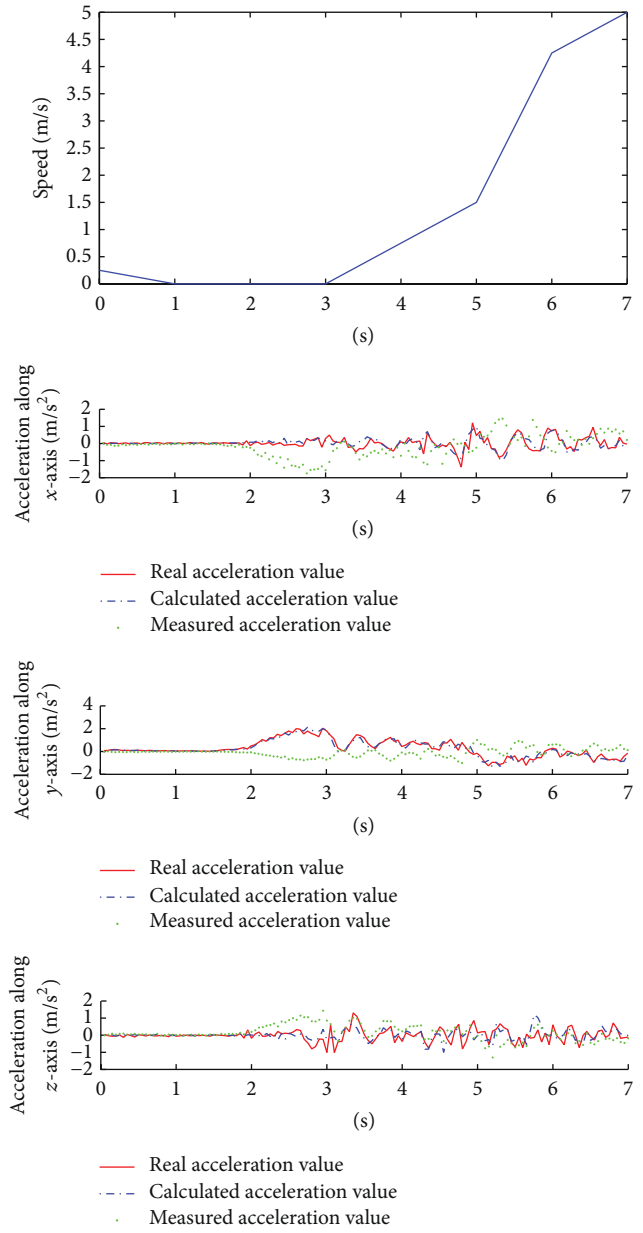


FIGURE 12: The acceleration value comparison diagram between estimation result and real value.

classified model, and the rest are applied to evaluate the estimation precision of our scheme.

Firstly, we established the coordinate relationship and calculated the acceleration on  $y$ -axis and  $z$ -axis. And then we need to extract all the available characteristics. At last, we take advantage of the model obtained in Section 6 to evaluate the traveling speed.

After making a research on the estimated result, we discover that most of the elements are distributed near the main diagonal, especially in the first seven states as shown in Table 5. The proportion of the samples whose error is less than two intervals is analyzed in Figure 13. The proportion is over 80% when traveling in the first four states, which make a satisfying performance. Unfortunately, the proportion only



TABLE 5: Evaluated result of part 1.

Part 1 DF[4](F) <sub>y</sub> , DF[3](P) <sub>z</sub> , DF[4](P) <sub>z</sub>	Estimated state			
	State 1	State 2	State 3	State 4
Real state				
State 1	29	5	9	1
State 2	14	4	11	8
State 3	3	1	19	14
State 4	2	0	12	32

TABLE 6: Evaluated result of part 2.

Part 2 DF[7](F) <sub>y</sub> , DF[5](F) <sub>z</sub> , DIA[3](F) <sub>z</sub>	Estimated state			
	State 5	State 6	State 7	State 8
Real state				
State 5	17	9	6	11
State 6	17	23	12	16
State 7	11	11	25	16
State 8	5	13	20	34

TABLE 7: Evaluated result of part 3.

Part 3 DIA[2](F) <sub>y</sub> , TF[8](F) <sub>y</sub> , DF[2](P) <sub>z</sub>	Estimated state			
	State 9	State 10	State 11	State 12
Real state				
State 9	29	10	12	15
State 10	36	19	16	16
State 11	27	14	13	18
State 12	12	11	16	21

succeeds by 60% when the traveling speed is between 4 m/s and 7 m/s. However, the distribution of evaluated speed in other areas is very sparse. and with this proportion we could also evaluate the traffic congestion scale by means of estimated result fusion.

Referring to [22], the congestion scale is divided into five grades, including expedite scale, smooth scale, slight congestion scale, moderate congestion scale, and severe congestion scale. They define 7 m/s as the threshold to distinguish the congestion scale and smooth scale. In a word, our scheme could achieve a satisfying performance in congestion scale and have some practicability.

We measured the energy consumption of GPS sampled at a period of 20 s (the longest period of calculated traffic condition using GPS), the energy consumption of accelerometer which was sampled at 20 Hz, and the energy consumption in standby mode as shown in Figure 14. The GPS is turned off after sampling in a period and turned on at the beginning of the next sampling period [14]. The experiment of energy consumption measuring is implemented on LG Nexus 4 mobile phone. The experiment result shows that in case of identical energy GPS works for 12 hours, and accelerometer works for more than 20 hours and the phone keeps in standby mode for 42 hours. In other words, the mobile phone

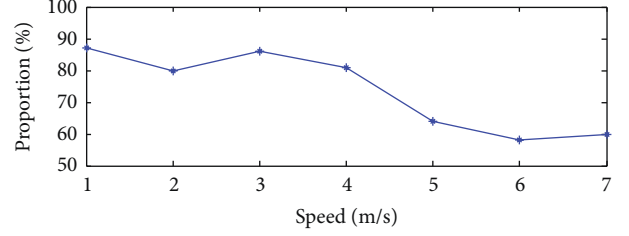


FIGURE 13: The proportion of samples near domain diagonal.

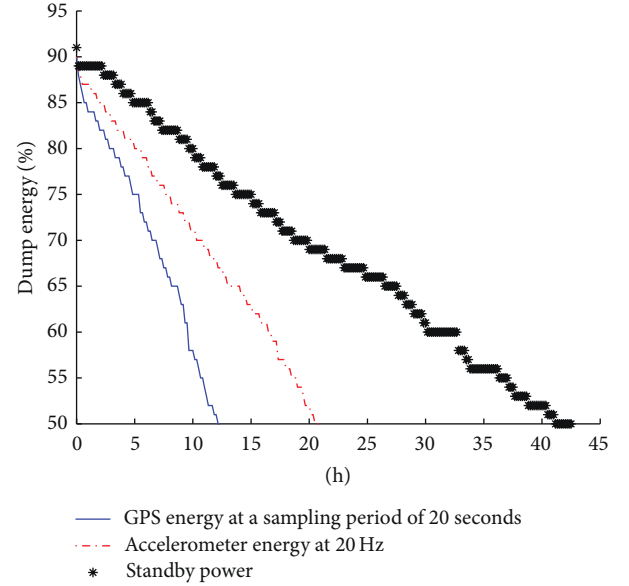


FIGURE 14: The energy consumption of GPS, accelerometer, and standby mode.

consumes 2.38% specified energy in standby mode in one hour, and the accelerometer module consumes 2.6% specified energy in one hour, and the GPS module consumes 5.95% in one hour. Therefore the energy consumption of GPS module is 2.3 times the energy consumption of accelerometer module in this experimental environment.

## 8. Conclusion and Future Work

On the one hand, the approach of coordinate system relationship establishment achieves a good performance including high accuracy and availability by detecting the starting action. Instead of compass, the accelerometer is not sensitive to the environment. Most importantly, our approach only requires an acceleration value of  $0.5 \text{ m/s}^2$ , which can be realized in most situations. On the other hand, we only take advantage of accelerometer to evaluate the traveling speed for the first time and could achieve a satisfying performance in congestion scale. Comparing with GPS, it is more energy-efficient. Besides, we could also detect the congestion scale of urban canyons where GPS is disabled.

In the near future, we would study the tracking algorithm and combine it with the approach of speed estimation. Furthermore, more work of the data fusion should be followed

TABLE 8: Speed evaluated result.

	Evaluated speed value												
	1 m/s	2 m/s	3 m/s	4 m/s	5 m/s	6 m/s	7 m/s	8 m/s	9 m/s	10 m/s	11 m/s	12 m/s	>12 m/s
Real speed value													
1 m/s	34	4	2	4	2	0	0	0	0	0	0	0	0
2 m/s	17	8	9	11	3	5	2	1	0	0	0	0	0
3 m/s	7	7	21	19	18	6	3	2	0	0	0	0	1
4 m/s	4	1	18	20	39	9	5	4	3	0	0	0	5
5 m/s	1	2	7	6	29	12	6	6	7	2	4	1	11
6 m/s	0	1	10	2	22	13	10	12	5	3	10	2	12
7 m/s	0	1	8	4	28	9	12	12	9	5	1	2	26

up to obtain the real traffic information of road network and achieve better precision.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgment

This work was supported by the Science and Technology Planning Project of Zhejiang Province (no. 2013C33062), Zhejiang Provincial Natural Science Foundation of China (Y13G020041).

### References

- [1] M. Tanelli, S. M. Savaresi, and C. Cantoni, "Longitudinal vehicle speed estimation for traction and braking control systems," in *Proceedings of the IEEE Conference on Control Applications (CCA '06), Symposium on Computer-Aided Control Systems Design (CACSD '06) and International Symposium on Intelligent Control (ISIC '06)*, pp. 2790–2795, IEEE, Munich, Germany, October 2006.
- [2] L.-J. Wu, "Experimental study on vehicle speed estimation using accelerometer and wheel speed measurements," in *Proceedings of the 2nd International Conference on Mechanic Automation and Control Engineering (MACE '11)*, pp. 294–297, Hohhot, China, July 2011.
- [3] J. Guo, J. Xia, and B. L. Smith, "Kalman filter approach to speed estimation using single loop detector measurements under congested conditions," *Journal of Transportation Engineering*, vol. 135, no. 12, Article ID 002912QTE, pp. 927–934, 2009.
- [4] F. Porikli and X. Li, "Traffic congestion estimation using HMM models without vehicle tracking," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 188–193, IEEE, Parma, Italy, June 2004.
- [5] J. A. Burke, D. Estrin, M. Hansen et al., *Participatory Sensing*, Center for Embedded Network Sensing, 2006.
- [6] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
- [7] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "BikeNet: a mobile sensing system for cyclist experience mapping," *ACM Transactions on Sensor Networks*, vol. 6, no. 1, article 6, 2009.
- [8] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks*, vol. 6, no. 2, article 13, 2010.
- [9] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "SoundSense: scalable sound sensing for people-centric applications on mobile phones," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services*, pp. 165–178, ACM, Kraków, Poland, June 2009.
- [10] J. C. Herrera, D. B. Work, R. Herring, X. J. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via gps-enabled mobile phones: the *Mobile Century* field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 568–583, 2010.
- [11] H. Bar-Gera, "Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: a case study from Israel," *Transportation Research Part C: Emerging Technologies*, vol. 15, no. 6, pp. 380–391, 2007.
- [12] D. B. Work, O.-P. Tossavainen, S. Blandin, A. M. Bayen, T. Iwuchukwu, and K. Tracton, "An ensemble kalman filtering approach to highway traffic estimation using gps enabled mobile devices," in *Proceedings of the 47th IEEE Conference on Decision and Control (CDC '08)*, pp. 5062–5068, IEEE, Cancun, Mexico, December 2008.
- [13] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2289–2302, 2013.
- [14] 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 (SenSys '09)*, pp. 85–98, ACM, Berkeley, Calif, USA, November 2009.
- [15] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, "SignalGuru: leveraging mobile phones for collaborative traffic signal schedule advisory," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, pp. 127–140, ACM, July 2011.
- [16] F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, and C. Ratti, "Real-time urban monitoring using cell phones: a case study in Rome," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 141–151, 2011.
- [17] J. Yoon, B. Noble, and M. Liu, "Surface street traffic estimation," in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys '07)*, pp. 220–232, ACM, San Juan, Puerto Rico, June 2007.

- [18] M. A. Ferman, D. E. Blumenfeld, and X. Dai, "An analytical evaluation of a real-time traffic information system using probe vehicles," *Journal of Intelligent Transportation Systems*, vol. 9, no. 1, pp. 23–34, 2005.
- [19] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*, pp. 323–336, ACM, Raleigh, NC, USA, November 2008.
- [20] X. Liu, Z. Wang, Z. Wang, S. Lv, and T. Guan, "A novel real-time traffic information collection system based on smartphone," in *Advances in Wireless Sensor Networks*, vol. 334 of *Communications in Computer and Information Science*, pp. 291–303, Springer, Berlin, Germany, 2013.
- [21] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [22] H. M. Wen and J. P. Sun, "Urban road traffic performance index," Tech. Rep. Db11/t 785-2011, Beijing Municipal Administration of Quality and Technology Supervision, 2011.

