

## Research Article

# SenSafe: A Smartphone-Based Traffic Safety Framework by Sensing Vehicle and Pedestrian Behaviors

Zhenyu Liu, Mengfei Wu, Konglin Zhu, and Lin Zhang

*Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, China*

Correspondence should be addressed to Lin Zhang; [zhanglin@bupt.edu.cn](mailto:zhanglin@bupt.edu.cn)

Received 10 June 2016; Revised 7 September 2016; Accepted 28 September 2016

Academic Editor: Francisco Martínez

Copyright © 2016 Zhenyu Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traffic accident involving vehicles is one of the most serious problems in the transportation system nowadays. How to detect dangerous steering and then alarm drivers in real time is a problem. What is more, walking while using smartphones makes pedestrian more susceptible to various risks. Although dedicated short range communication (DSRC) provides the way for safety communications, most of vehicles have not been deployed with DSRC components. Even worse, DSRC is not supported by the smartphones for vehicle-to-pedestrian (V2P) communication. In this paper, a smartphone-based framework named SenSafe is developed to improve the traffic safety. SenSafe is a framework which only utilizes the smartphone to sense the surrounding events and provides alerts to drivers. Smartphone-based driving behaviors detection mechanism is developed inside the framework to discover various steering behaviors. Besides, the Wi-Fi association and authentication overhead is reduced to broadcast the compressed sensing data using the Wi-Fi beacon to inform the drivers of the surroundings. Furthermore, a collision estimation algorithm is designed to issue appropriate warnings. Finally, an Android-based implementation of SenSafe framework has been achieved to demonstrate the application reliability in real environments.

## 1. Introduction

Traffic safety becomes one of the serious problems in the transportation systems. As the number of the vehicles is growing, the levels of traffic accidents have increased significantly. Not only vehicles, but also pedestrians and cyclists are facing the safety threats from traffic accidents. According to [1], traffic accidents resulted in more than 500 thousand deaths and 14 million injures worldwide by the end of May in 2016. One of the main reasons for traffic accidents is that the drivers cannot notice behaviors of surrounding vehicles on time. Using smartphone while walking is increasingly apparent in our society, and when people walk with their attention on smartphones and distracted from the scenery, potential accidents are in front of them. To reduce the traffic accidents, it is necessary to assist the drivers to have a better understanding of the surrounding environments including the coming vehicles and adjacent pedestrians and cyclists.

Dedicated short range communication (DSRC) [2] has been accepted as the most promising approach to enhance

the transportation safety. It consists of a set of vehicles equipped with the on-board unit (OBU), and a set of road side units (RSUs) along the roads, and aims to provide reliable and low latency vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. However, due to additional effort for DSRC deployment, most of vehicles have not been equipped with OBU for DSRC. Besides, DSRC is not supported by the general smartphones for vehicle-to-pedestrian (V2P) communication.

To enable the safety of transportations, many strategies are proposed. There are some works using the vehicle-mounted device for safety assistance driving system. Cameras and radars are used to monitor the driving behaviors and alert the dangerous distances between vehicles to provide technical support for the auto manufacturers. Although the cost of cameras and radars based safety technology is decreasing, these safety technologies have not been deployed in economy vehicles. It still needs time before the majority of vehicles are deployed with these safety technologies. Furthermore, due to

lack of communication, vehicles need to make decisions by themselves.

With the widespread use of the smartphone, most of drivers and pedestrians have smartphones, which provides the potential to use them for the enhancement of traffic safety. Nowadays, smartphones are progressively equipped with functional sensors (e.g., accelerometer, gyroscope, GPS, camera, etc.), and these sensors can be applied to recognize and monitor various vehicle behaviors. Additionally, they contain communication resources and enable the quick deployment of new applications [3].

Sensor technology available in smartphones enables the monitoring of vehicle mobility behaviors, including lane change, turns, acceleration, and brake. If such information can be sensed and transmitted to other vehicles or pedestrians, they can be potentially used in collision prevention, early crash detection, congestion avoidance, and so forth. Smartphones have abundant communication resources, such as Bluetooth, Wi-Fi, and 3G/LTE. However, using them to construct appropriate networks for efficient information sharing in transportation systems is difficult. In particular, the transmission range of the Bluetooth cannot satisfy the requirements of the communication. Wi-Fi has the procession of authentication and association before data transmission and is more suitable for 1-to-N communication. 3G/LTE needs the assistance of the base station.

In this paper, we propose SenSafe, a smartphone-based framework for traffic safety by sensing, communication, and alerting, which overcomes the limitations of requesting additional professional equipment inherent in the existing approaches.

SenSafe is a novel framework which only utilizes the smartphone to improve the transportation safety. It detects and reports vehicle's events based on the sensing data collected from steering behaviors in urban area. The changes in the angle of vehicle heading and the corresponding displacement during a steering maneuver are calculated for driving behavior classification. Driving behaviors are classified into different types, including turn, lane change, acceleration, and brake. Beacon-based communication is provided to exchange the driving information. Surrounding environment reminding and collision forewarning are provided based on the data from the surroundings.

We highlight our main contributions as follows:

- (i) We propose a framework, SenSafe, which only uses the smartphone to sense the driving behavior, exchange the driving information, and afford the reminding and alerting.
- (ii) We provide the detection and differentiation of various driving behaviors by only utilizing the smartphone's built-in sensors in urban area. We analyze the sensing data collected from urban area and find that vehicles cannot always make turns and lane changes smoothly owing to the influence of surrounding vehicles and pedestrians. The temporary interruptions are taken into consideration to improve the accuracy of the driving behaviors detection.

- (iii) We furnish the beacon-based communication, which modifies the service set identifier (SSID) field of Wi-Fi beacon frame to carry the driving behavior, position, and mobility information. A data compression and decompression method is provided to make use of the SSID field. Event-driven communication is utilized to provide the surrounding drivers with immediate reminding. Ordinary promptings of the surrounding vehicles' behaviors and safety alerts of the coming collisions are provided to the drivers.
- (iv) An Android-based implementation of SenSafe framework has been achieved to demonstrate the evaluation results and application reliability in real environments.

The rest of this paper is organized as follows. In Section 2, a brief overview of the related works is provided. In Section 3, the framework overview of SenSafe is introduced. The details for three modules of SenSafe are explained in Sections 4–6. The implementation, performance evaluation, and experiment results are shown in Section 7. Finally, this paper is concluded in Section 8.

## 2. Related Works

There are some works to detect the vehicle behavior. Some works use the fixed vehicle-mounted devices to monitor the vehicle behavior. Mobileye uses cameras and radars to provide the technical support for the auto manufacturers [4]. Although the cost of vehicle safety technology is decreasing, these safety technologies have not been deployed in economy vehicles. It still needs time before the majority of vehicles are deployed with these safety technologies. With the wide spread of smartphones, smartphone solutions can be used in vehicles, and there are some works focusing on using smartphones to assist drivers.

Drivea [5], iOnRoad [6], and Augmented Driving [7] are apps which have capability of detecting lane departures and warn drivers when the distance to the front vehicle is too close. CarSafe [8] alerts distracted drivers using dual cameras on smartphones, one for detecting driver state and the other for tracking road conditions. Although camera-based systems have the functionality in assisting the driver, they have limitations in terms of computational overhead and requirement of image quality and work worse at night and with bad lighting conditions.

Camera-free sensors in the smartphone have been utilized in traffic regulator detection [9], localization [10], transportation mode classification [11], vehicle speed estimation [12], and the driving behavior detection [10, 13–15]. In [13], smartphone sensing of vehicle dynamics is utilized to determine driver phone use. Inertial measurement unit (IMU) sensors including accelerometers and gyroscopes of the smartphone are used to capture differences in centripetal acceleration due to vehicle dynamics. In [14], three algorithms which detect driving events using motion sensors embedded on a smartphone are proposed. The first detection algorithm is based on data collected from GPS receiver. The second and third detection algorithms utilize accelerometer

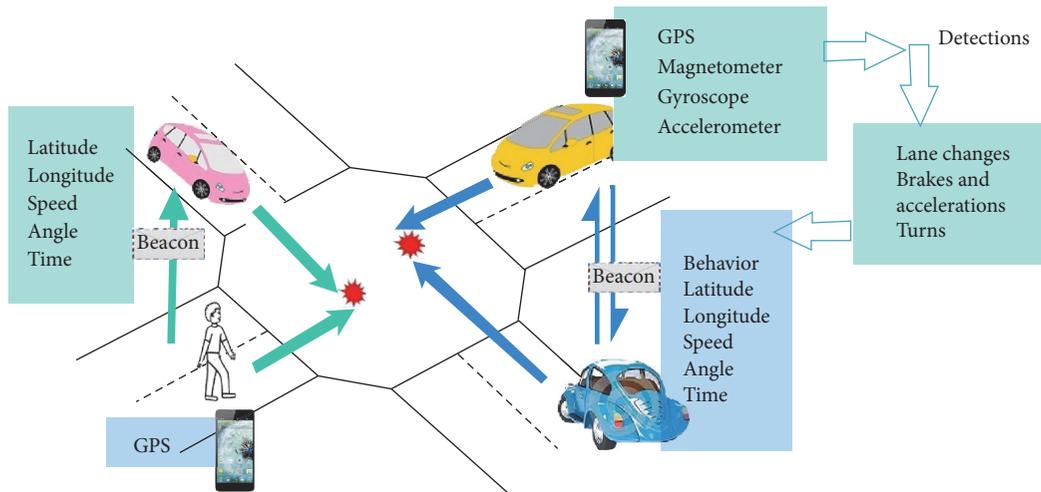


FIGURE 1: Application scenarios.

sensor and use pattern matching technique to detect driving events. In [15], a vehicle steering detection middleware called V-Sense is developed on commodity smartphones by only utilizing nonvision sensors on the smartphone. Algorithms are designed for detecting and distinguishing various vehicle maneuvers, including lane changes, turns, and driving on curvy roads. However, the paper only considers the smooth driving behaviors. Once there is a sudden brake because of the surrounding vehicles and pedestrians, it cannot work well. In [10], embedded sensors in smartphones are utilized to capture the patterns of lane change behaviors, and this paper on the driving environment of highway.

There are some works for the communication. Some companies and researchers focus on using the DSRC for V2V communication [16, 17]. However, all of these works need special devices for communications, which have not been deployed in most of vehicles. Some works use the communication resources of smartphone to avoid using extra devices. The master access point disseminates information to the rest of the units through Wi-Fi of smartphones [18], and it is suitable for the solid groups. But, for the dynamics groups, Wi-Fi has the processions of authentication and association and is usually for the 1-to-N communications. For the Wi-Fi Direct, the device discovery phase may take over ten seconds in some cases [19]. The authors modify the SSID of Wi-Fi beacons to store the location and speed of the smartphone for alert [20]. However, they do not consider the driving behaviors of the vehicles. Besides, as the access points (APs) can only broadcast beacons and the clients can only receive the beacons, how to achieve the bidirectional communications is not considered.

Different from the previous works, we propose an integrated smartphone-based framework SenSafe for traffic safety considering sensing vehicle behaviors. This framework is only based on smartphone and thus can be easily deployed due to the widespread use of the smartphone. Vehicle behavior sensing is one of the key points in this framework. To improve the robustness and the accuracy of the vehicle behavior sensing, nonvision sensors are utilized to reduce

the influence of the image quality, and unsmooth driving behaviors are considered as there may be some brakes in the procession of turns. Considering that only AP broadcasts beacons and cannot receive beacons from the other APs, an event-driven communication method is proposed to achieve the bidirectional communications. Finally, the reminder events are divided into ordinary prompting of the surrounding vehicles' behaviors and safety alert of the coming collisions for the drivers.

### 3. Framework Overview

This section describes the high level overview of the framework SenSafe that we propose for traffic safety. SenSafe is focused on the vehicles and pedestrian safety. One of the main reasons for the traffic accident is that the drivers cannot notice the behaviors of surrounding vehicles and pedestrians on time. SenSafe considers using the smartphone which is widespread to improve the traffic safety without any cost of installing new device.

As is shown in Figure 1, SenSafe uses the sensors from the smartphone to collect the data about the vehicle mobility. After that, these data can be used to obtain the driving behavior of the vehicles. Using the information transferred from the surroundings, drivers can have a better understanding of the environments. Besides, the smartphone of the pedestrian obtains the moving information from the GPS and broadcasts it to the vehicles to enhance the vulnerable pedestrian safety.

System architecture is shown in Figure 2. The framework is made of three components: driving behaviors detection module, beacon-based communication module, and collision forewarning module. The driving behaviors detection module considers the output of motion sensors and GPS. Acceleration, braking, turns, and lane changes are detected using the proposed algorithm. Beacon-based communication module compresses the sensed data into the SSID of the beacon for communication. Collision forewarning module uses the data from the surrounding to remind the driver of the driving behaviors and finds out the vehicles which have

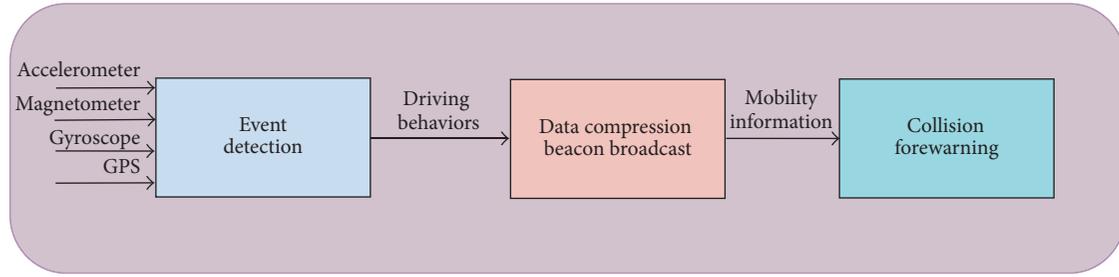


FIGURE 2: System Architecture.

threat to the current vehicle and pedestrians who might be threatened.

#### 4. Driving Behaviors Detection

During a single trip, the smartphone collects input data from motion sensors and GPS, and the driving behavior detection system decides the type of the driving behaviors. This section details the design and functionalities of driving behavior detection. First, we describe how the sensors of smartphones are utilized to determine whether the vehicle is making acceleration, brake, turn, or changing lane. Then, we show how SenSafe classifies such different vehicle driving behaviors based on the detection results. The Android-based smartphone is used as our target platform to collect raw data from four on-board sensors, 3-axis accelerometer, magnetometer, gyroscope, and GPS receiver.

**4.1. Coordinate Alignment.** Given the direction of gravity on the smartphone body coordinate system, the smartphone attitude can be fixed within a conical surface in the earth coordinate system. As a result, we align the smartphone coordinate with the earth coordinate for removing 3 degrees of freedom to 1 degree by projection, as shown in Figure 3. Magnetometer is utilized to get the angle  $\delta$  between  $Y_e$  and  $Y'$  axis in the earth coordinate system, where  $Y'$  is the projection of  $Y_s$  of the smartphone body coordinate system on the  $X_e$ - $Y_e$  plane of the earth coordinate system.  $X_e$  (pointing to the earth east),  $Y_e$  (pointing to the earth north), and  $Z_e$  (parallel with the gravity) are the three reference axes in the earth coordinate system. Combining the result with the angle derived from the magnetometer reading and the rotation matrix, the component of sensing data corresponding to the vehicle moving direction can be calculated. The detailed formulation of the rotation matrix is explained in [21].

#### 4.2. Detection Algorithm

**4.2.1. Acceleration and Brake Detection.** With the help of accelerometer, the acceleration and the brake can be distinguished (Figures 4 and 5). Moving average filter is used to remove noise from the raw acceleration. Due to the unpredictable road conditions and diverse driving preferences, the acceleration in real implementation includes noise. In order to filter out the noise, we use a state machine to detect the

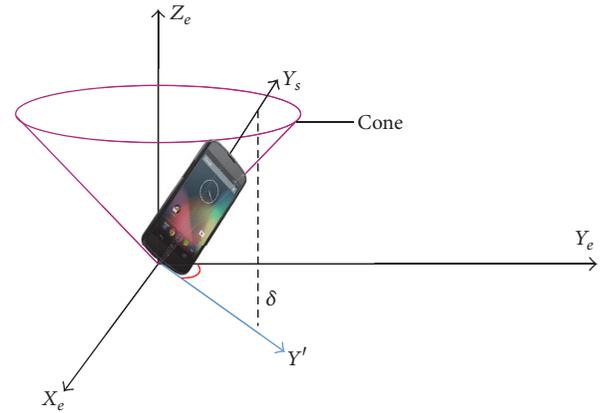


FIGURE 3: Align the smartphone coordinate system with the earth coordinate system.

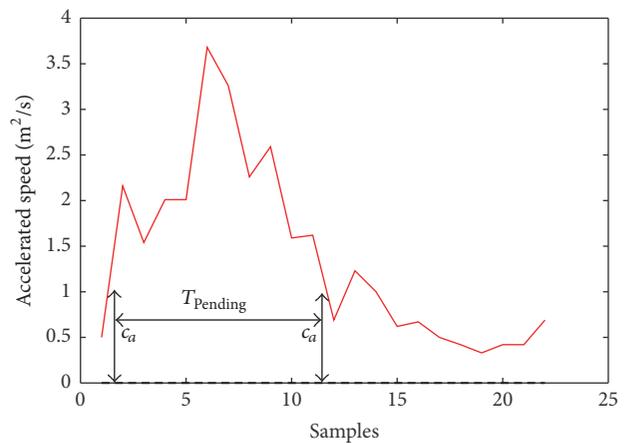


FIGURE 4: Accelerometer readings when the vehicle accelerates.

event. There are basically five states, Waiting-for-Acc, Acc-Pending, Brake-Pending, Acceleration, and Brake. The state transition procession is shown in Figure 6. The following list describes the specification of the parameters used in acceleration and brake detection algorithm:

*Acc*: acceleration from the moving average filter

*Acc.Threshold*: the threshold of the acceleration to enter the Acc-Pending state

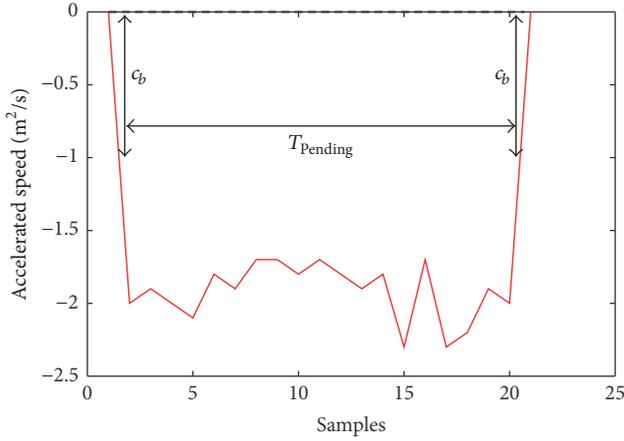


FIGURE 5: Accelerometer readings when the vehicle brakes.

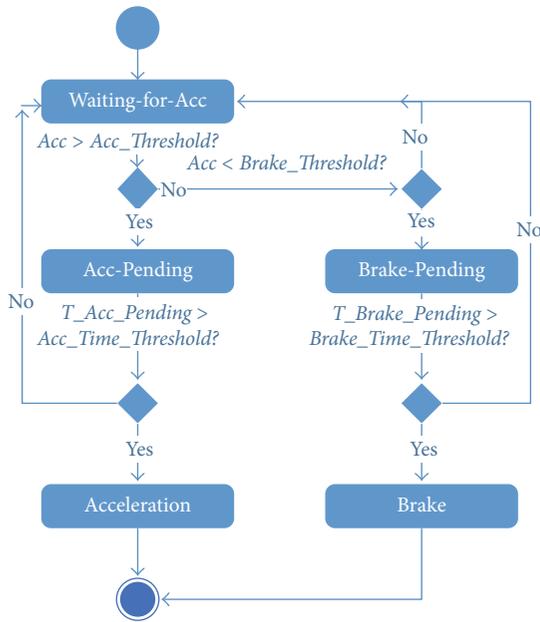


FIGURE 6: State diagram of the acceleration and brake detection.

*Brake\_Threshold*: the threshold of the acceleration to enter the Brake-Pending state

*Acc\_Time\_Threshold*: the threshold of time in Acc-Pending state to enter the Acceleration state

*Brake\_Time\_Threshold*: the threshold of time in Brake-Pending state to enter the Brake state

*T\_Acc\_Pending*: the time in Acc-Pending state up to now

*T\_Brake\_Pending*: the time in Brake-Pending state up to now

Waiting-for-Acc is the initial state for the acceleration and brake detection. As the real acceleration appears as the undulatory curve due to noises, Acc-Pending and Brake-Pending are two transitional states to reduce the influence of

the noise. For the acceleration, system enters into the Acc-Pending state after the *Acc* is greater than *Acc\_Threshold* and exits the Acc-Pending state when the *Acc* is less than or equal to *Acc\_Threshold*. If *T\_Acc\_Pending*  $T_{Pending}$  in Acc-Pending state is greater than *Acc\_Time\_Threshold*, the state becomes Acceleration, which means an acceleration is ongoing. For the brake, system enters into the Brake-Pending state after the *Acc* is less than *Brake\_Threshold* and exits the Brake-Pending state when the *Acc* is greater than or equal to *Acc\_Threshold*. If *T\_Brake\_Pending* in Brake-Pending state is greater than *Brake\_Time\_Threshold*, the state becomes Brake.

**4.2.2. Turn and Lane Change Detection.** The Z-axis readings of gyroscope on the smartphone are utilized to represent the vehicle angular speed. When the drivers do some behaviors to change the direction of vehicles (e.g., changing single or multiple lanes and making turns), the angular speeds have the obvious improving (Figure 7). For the left turn, a counterclockwise rotation around the Z-axis occurs and generates positive readings, whereas during a right turn, a clockwise rotation occurs and thus generates negative readings. For a left lane change, positive readings are followed by negative readings, whereas for a right lane change, negative readings are followed by positive readings.

By detecting bumps in the Z-axis gyroscope readings, we can determine whether the vehicle has made a left/right turn or has made single-lane change. The other steering maneuvers, that is, turn back and multiple-lane change, have a similar shape but with a different size in terms of width and height of the bumps. The parameter description of the turn and lane change detection is shown in the following list:

*Ang\_SE\_Threshold*: the threshold of angular speed to enter and leave a possible bump

*Ang\_Top\_Threshold*: the threshold of angular speed which the maximum value of the valid bump should be greater than

*T\_Bump*: duration that angular speed is greater than *Ang\_SE\_Threshold* in a bump

*T\_Bump\_Threshold*: the minimum threshold of the *T\_Bump* for a valid bump

*G\_threshold*: the threshold of GPS speed to estimate if the vehicle is motionless

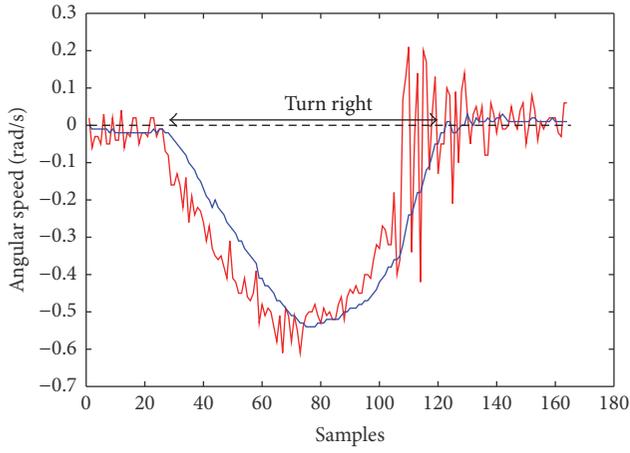
*Delay\_threshold*: the threshold to judge the consecutive bump

*T\_stop*: the duration whose GPS speed is less than the *G\_threshold*

*T\_wait*: duration of Waiting-for-Bump state

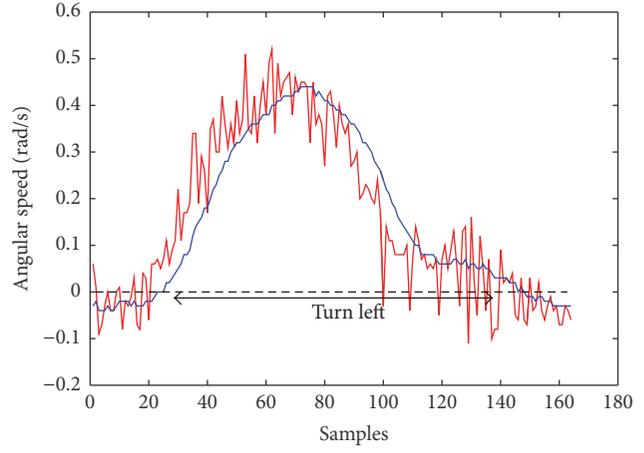
Moving average filter is adopted to remove noise from the raw gyroscope readings. The delay parameter of the filter is set to 15 samples which correspond to 0.75 seconds in the time domain. Experimental observation shows that it is short but good enough to extract the waveform of the bumps.

To reduce false positives and differentiate the bumps from jitter, a bump should satisfy the following three constraints for its validity: (1) all the readings during a bump should



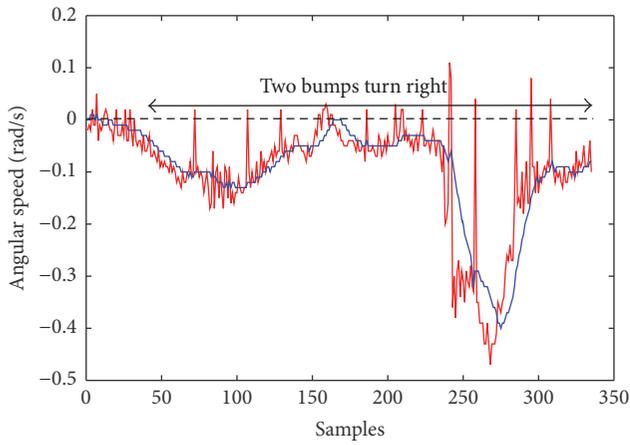
— Raw data  
— Filtered data

(a) Turn right (one bump)



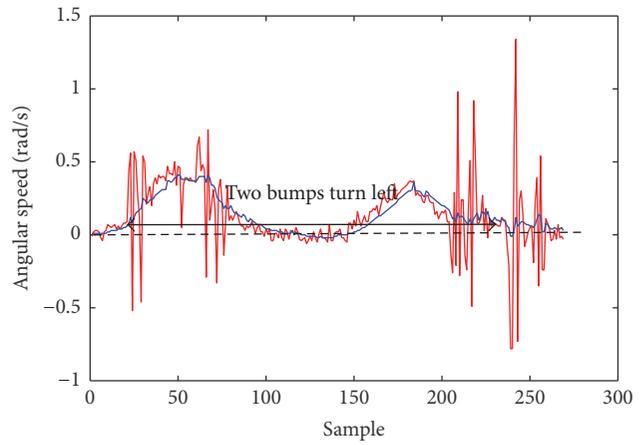
— Raw data  
— Filtered data

(b) Turn left (one bump)



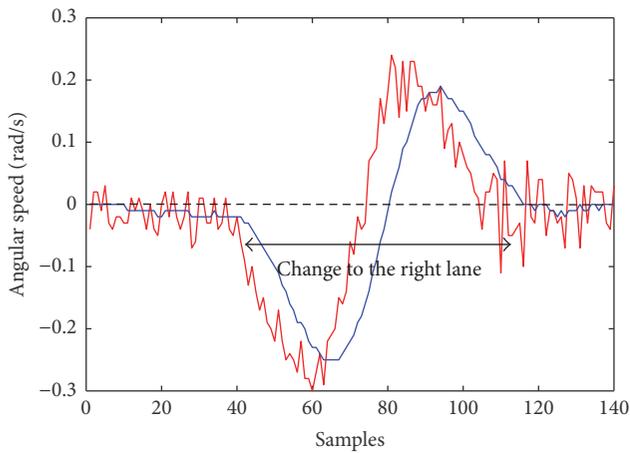
— Raw data  
— Filtered data

(c) Turn right (two bumps)



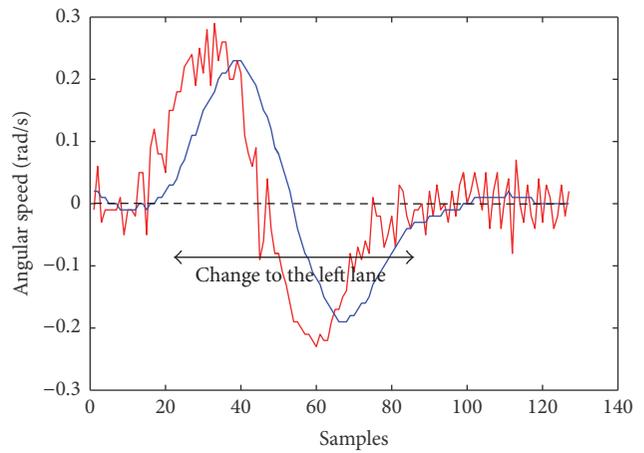
— Raw data  
— Filtered data

(d) Turn left (two bumps)



— Raw data  
— Filtered data

(e) Change to the right lane



— Raw data  
— Filtered data

(f) Change to the left lane

FIGURE 7: Continued.

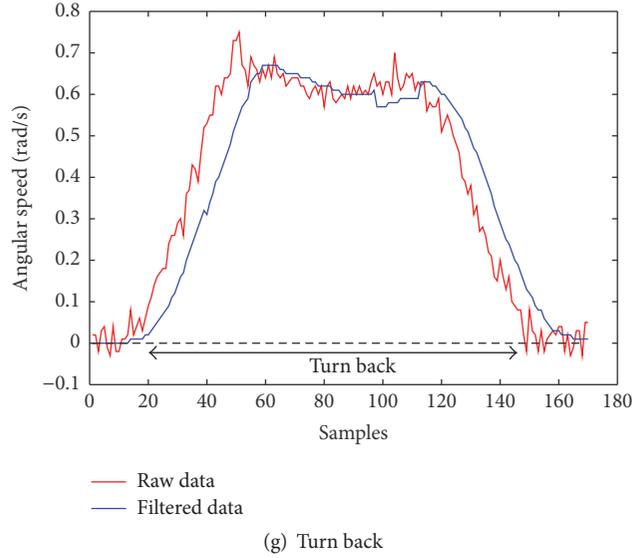


FIGURE 7: Gyroscope readings when the vehicle makes turns or lane changes.

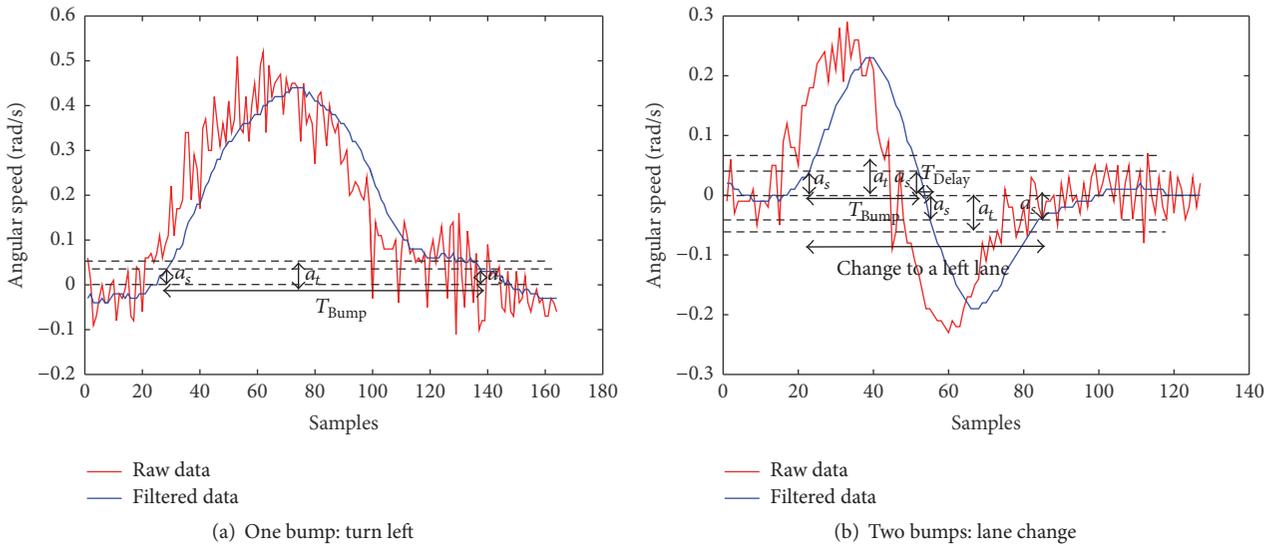


FIGURE 8: Symbol description in gyroscope reading.

be larger than  $Ang\_SE\_Threshold$   $a_s$ , (2) the largest value of a bump should be no less than  $Ang\_Top\_Threshold$   $a_t$ , and (3) the duration of a bump  $T_{Bump}$  should be no less than  $T\_Bump\_Threshold$   $t_b$  (Figure 8(a)). For the two valid continuous bumps, the time between the two bumps should be less than  $Delay\_threshold$   $T_{Delay}$  except the time when the vehicle stops (Figure 8(b)).

There are seven states in the bump detection algorithm: No-Bump, One-Bump, Waiting-for-Bump, More-Bump, Bump-End, Turn, and Lane-Change. The state transition is influenced by angular speed and GPS speed. Angular speed is the Z-axis gyroscope reading. The state transition procession is shown in Figure 9.

In No-Bump state, angular speed is continuously monitored. When the absolute value of the measured angular speed

reaches  $Ang\_SE\_Threshold$ , it is treated as the start of a possible bump, and the algorithm enters One-Bump state.

The One-Bump state terminates when the absolute value of the measured angular speed is below  $Ang\_SE\_Threshold$ . If the time of duration in One-Bump state is larger than  $T\_Bump$  and the largest angular speed is larger than  $Ang\_Top\_Threshold$ , hence satisfying the three constraints, the first detected bump is assigned to be valid. In such a case, the algorithm enters Waiting-for-Bump state. Otherwise, it returns to No-Bump.

In Waiting-for-Bump state, it monitors the angular speed until its absolute value reaches  $Ang\_SE\_Threshold$ , and the duration is defined as  $T\_wait$ . The GPS speed is also monitored to see if the turn is interrupted halfway.  $T\_stop$  is used to define the duration whose GPS speed is less than

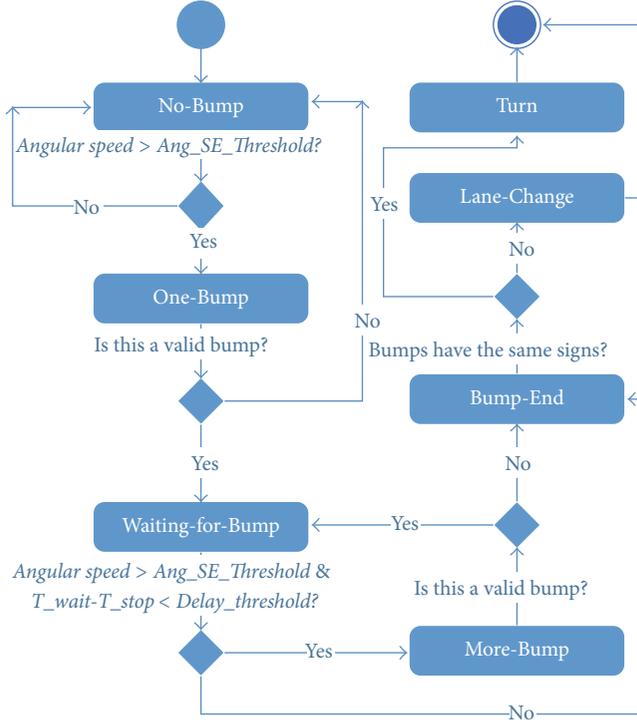


FIGURE 9: State diagram of the turn and lane change detection.

the  $G\_threshold$ , which means the turn is interrupted. If the difference between the  $T\_wait$  and  $T\_stop$  is less than  $Delay\_threshold$ , the system enters the More-Bump state, which means the oncoming bump is considered as consecutive bumps. If the new bump is valid, the system enters Waiting-for-Bump state for new bumps. If not, the system enters Bump-End state to determine the driving behavior.

In Bump-End state, the signs of these consecutive bumps can be used for driving behavior judgment. If these bumps have the same signs, the driving behavior is determined to be turn. If they have the opposite signs, the driving behavior is determined to be lane change. Besides, if there is only one valid bump (i.e., the second bump is invalid), the driving behavior is determined to be turn.

Based on bump detection, driving behaviors are classified into turns and lane changes. To get the detailed classification of the behaviors (e.g., left turn, right turn, and U-turn), the difference in the heading angle between the start and the end of a driving behavior is used.

We calculate the change in vehicle's heading angle from the readings of the gyroscope.  $\Delta t$  is the time interval of sampling.  $i$  is the bump index of the valid bumps.  $A_{n_i}$  is the angular speed of the  $n_{i\text{th}}$  sampling and is used to approximately represent the average angular speed during the sampling period. Thus, we get the change in vehicle's heading angle from the integration of the heading angle change of each sampling period:

$$\theta = \sum_{i=1}^I \sum_{n_i=1}^{N_i} A_{n_i} \Delta t. \quad (1)$$

TABLE 1: The bounds of the angle change of heading for driving behavior.

	Lower bound	Upper bound
Turn left	$70^\circ$	$110^\circ$
Turn right	$-70^\circ$	$-110^\circ$
Turn back	$-200^\circ/160^\circ$	$-160^\circ/200^\circ$
Left lane change	$-20^\circ$	$20^\circ$
Right lane change	$-20^\circ$	$20^\circ$

For the turning left,  $\theta$  is around  $90^\circ$ . For turning right, it is around  $+90^\circ$ . For turning back,  $\theta$  is around  $\pm 180^\circ$ . For the lane change, it is around  $0^\circ$ .

As the drivers cannot make a perfect driving behavior to get the perfect coincident degree, the ranges of the driving behaviors are extended as shown in Table 1.

Based on the horizontal displacement, we furthermore extract the multiple-lane change from the lane change as the horizontal displacement of multiple-lane change is greater than the single one. If the horizontal displacement is less than  $HD\_Lower\_Threshold$  (e.g., the width of the lane), it means that these consecutive bumps are caused by some interference instead of lane change. If the horizontal displacement is greater than  $HD\_Upper\_Threshold$ , the behavior is treated as the multiple-lane change. Otherwise, the behavior is treated as single-lane change. The horizontal displacement  $H$  can be calculated as follows. Thereinto,  $\Delta t$  is the time interval of

sampling,  $A_i$  is the angular speed of the  $i_{th}$  sampling, and  $G_n$  is the GPS speed of the  $n_{th}$  sampling:

$$\begin{aligned} \theta_n &= \sum_{i=1}^n A_i \Delta t, \\ H &= \sum_{n=1}^N G_n \Delta t \sin(\theta_n). \end{aligned} \quad (2)$$

## 5. Beacon-Based Communication

We use the Wi-Fi of smartphones for communication. As the normal Wi-Fi has the association and authentication procession which can cause the latency, we choose the beacon frame of the Wi-Fi AP to carry the sensing information. The beacon frame is used to declare the existing AP and can be transferred by the AP without association and authentication procession. The Beacon Stuffing embeds the intended messages within the SSID field of the Wi-Fi beacon header and is available in the Wi-Fi AP mode [22].

The problem of using beacon to transfer sensing information is that the beacon can only be transferred from the AP to the client, which causes the one-way transmission. However, it is not enough that only a part of the devices transfers the sensing information, as every device needs to broadcast its mobility information to the others. To solve this problem, we propose the event-driven communication algorithm; once the vehicle detects an event (e.g., acceleration, brake, turn, and lane change), the smartphone inside will switch to the AP mode to broadcast the beacons for a while; after that the smartphone will switch to the mode to scan the beacons.

To provide the reference for collision forewarning module, these elements are selected: the latitude and longitude from the GPS reader, the speed from the GPS reader (m/s), the travel direction from the magnetometer sensor (degree 0~360), the time from the GPS reader, and the driving behaviors. These elements are combined together to replace SSID field of beacon. A special string "Sen" is used in front of these element for differentiating SenSafe's SSID from the others.

To satisfy the requirements of accuracy, the 0.1 meters approximately correspond to the 0.000001 degrees in the latitude and longitude, which means the latitude and the longitude need 19 characters in the SSID field (e.g., 116.364815 and 39.967001 in BUPT). Besides, the time from the GPS reader also needs too many characters. If we want to make the precision of the time to be millisecond (e.g., 20:20:20 234), it will take 9 characters even without considering separators. However, the length of beacon messages' SSID field is 32 characters, which is not enough for all elements. As a result, we compress these elements in the AP side and decompress these elements in the client side.

For the latitude, one degree is about 111 kilometers. As the maximum transmission range of the Wi-Fi beacon is less than 1 kilometer, most of the significant digits of latitude and longitude are same for the sender and receiver. There is no need to transfer all the significant digits of latitude and longitude. Thus, the significant digits of latitude and

	Time	Latitude	Longitude
Remote data	20:20:20 234	116.364815,	39.967001
Compressed data	20234	64815	67001
Local data	20:20:21 136	116.365173,	39.966872
Decompressed data	20:20:20 234	116.364815,	39.967001

FIGURE 10: The example of compression and decompression.

	Time	
Remote data	20:20:59 121	
Compressed data	59121	
Local data	20:21:00 136	
Boundary situation	20:21:59 121? 20:20:59 121?	Difference is too large
Decompressed data	20:20:59 121	

FIGURE 11: The example of time boundary situation.

longitude are chosen to cover the range around the 1000 meters (i.e., last 5 significant digits). It is similar for the time, as hour and minute are always same for the senders and receivers. Therefore, we use 4 significant digits to represent the time from 0.01 seconds to 10 seconds. An example is shown in Figure 10. Further, due to the boundary situation of the time, when the local device decompresses the time from the remote devices, it will compare the difference between the decompressed time and the local time. If the absolute value of difference is greater than the threshold (30 seconds), the minute of the decompressed data will be modified to be the adjacent number (+1/-1) to get the minimum reasonable difference. An example is shown in Figure 11. The remote time is 20:20:59 121, and the local time is 20:21:00 136, which will make the normal decompressed time 20:21:59 121. As it is unreasonable, the adjacent minute (20:20) is chosen to be the prefix of the remote time to get the minimum absolute value of time difference. For the latitude and longitude, the purpose of the decompression is obtaining the minimum difference between the remote location and the local location. Similar to the time, if the absolute value of difference is greater than the threshold (200 meters), the first decimal places of the latitude and longitude are modified to be the adjacent number (+1/-1) to get the minimum reasonable difference.

The format of the SSID field is demonstrated in Figure 12. We use the number to represent the special driving behaviors in the driving behavior segment (0: pedestrian, 1: acceleration, 2: brake, 3: turn left, 4: turn right, 5: turn back, 6: left lane change, and 7: right lane change). For the pedestrians, their smartphones sense their moving information from the GPS readers and broadcast the same elements to remind the drivers of their existence. The driving behaviors element is set to "0", which means this message is from the pedestrian.

“Sen”	Driving behavior	Latitude	Longitude	Speed	Direction	Time	Reservation
0	3	5	10	15	19	22	26
							32

FIGURE 12: The format of the SSID field.

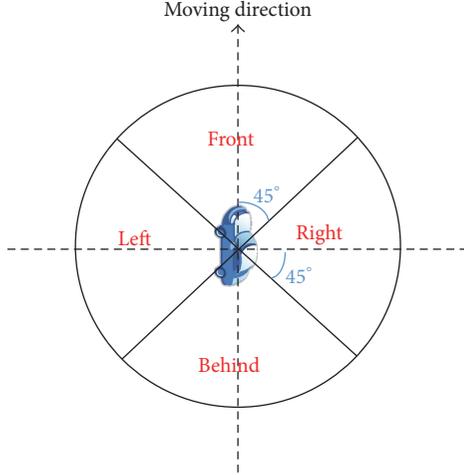


FIGURE 13: The area division for safety reminding.

## 6. Collision Forewarning

The vehicle will receive two kinds of messages: one kind from the pedestrians and the other from the vehicles.

After the vehicle gets the driving information of the surrounding vehicles, it will estimate if these vehicles will crash into it to ensure the safety of the vehicle and the driver. Besides, it will also remind the drivers of the driving behaviors from the surrounding vehicles to make the drivers have a better understanding of the driving environment.

As is shown in Figure 13, considering the vehicle moving direction, the surrounding area of the vehicle is divided into four quarters: front, behind, left, and right. After receiving a new message, the vehicle will calculate which area it belongs to. Furthermore, the driver can get the information where the threat comes from.

When the vehicle receives the messages from the surrounding vehicles, it translates latitudes and longitudes to Gauss-Krueger plane rectangular coordinates system. Then it calculates the driving vector of each vehicle and finds vehicles that have intersection with it and gets the location where they may have the intersection, as shown in Figure 14. If the vectors have the intersection in the near future, it will calculate the distance of the current vehicle and the threat vehicle to intersection and calculate the time to the intersection (safety reaction time) furthermore. If one of these two times is less than the *safety\_reaction\_time\_threshold*, and the absolute difference of them (safety intersection time) is less than *safety\_intersection\_time\_threshold*, these two vehicles will be estimated to have the threat of crashing into each other. When the vehicle receives the messages from the surrounding pedestrian, it will calculate the distance of the current vehicle and pedestrian to intersection and the time

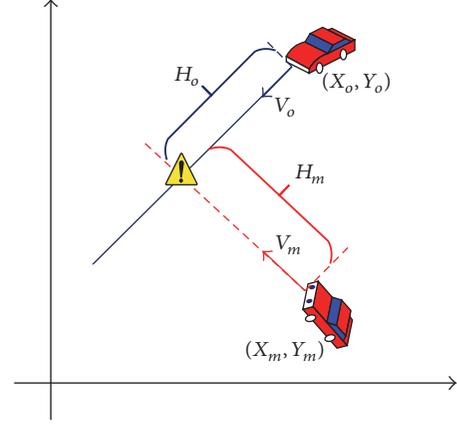


FIGURE 14: Intersection collision estimation.

of vehicle to the intersection. If the distance of pedestrian is less than *safety\_distance\_threshold*, and the time of vehicle to the intersection is less than *safety\_reaction\_time\_threshold*, the vehicle will be estimated to have the threat to the pedestrian.

We reference the safety reaction time threshold (3 seconds) recommended in [23] and safety intersection time threshold (2 seconds) recommended in [24] to avoid a collision when GPS is accurate. Assuming the inaccuracies of GPS location are up to 10 meters and the speeds of the vehicles are around 10 m/s, as a result, the error of the safety reaction time is up to 1 second and the error of the safety distance is up to 10 meters. Thus, we set safety reaction time threshold (4 seconds) to 1 second higher than the value which assumes the GPS is accurate and safety intersection time threshold (4 seconds) to 2 seconds higher than the value which assumes the GPS is accurate.

## 7. Performance Evaluation

To evaluate the performance of SenSafe, we implemented SenSafe on a Samsung Galaxy Note II and a Huawei Changwan 4X.

7.1. *Parameter Setting.* The parameters used in SenSafe are set as the values in the following list:

- Acc\_Threshold*: 0.8 m/s<sup>2</sup>
- Brake\_Threshold*: -1 m/s<sup>2</sup>
- Acc\_Time\_Threshold*: 0.6 seconds
- Brake\_Time\_Threshold*: 0.6 seconds
- Ang\_SE\_Threshold*: 0.03 rad/s
- Ang\_Top\_Threshold*: 0.05 rad/s
- T\_Bump\_Threshold*: 1.5 seconds

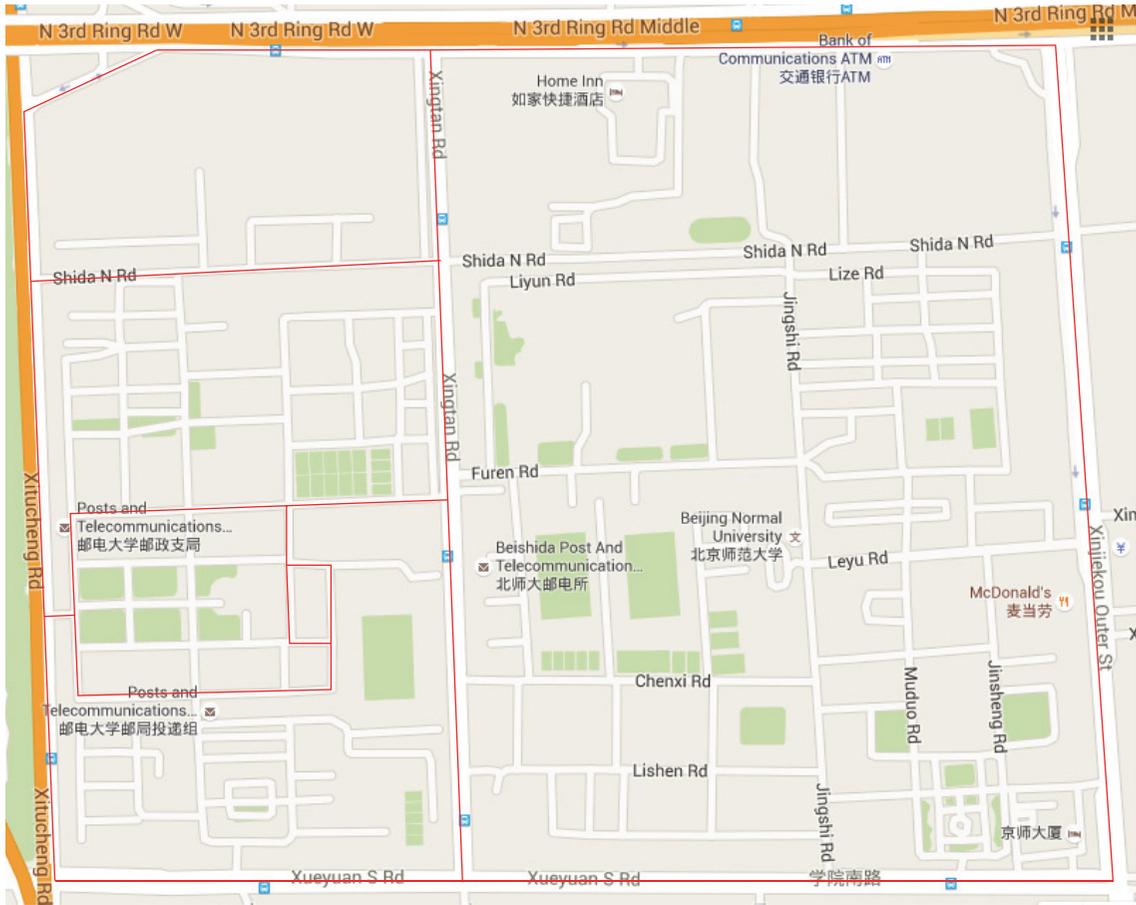


FIGURE 15: Real road testing routes.

- Delay\_threshold*: 2 seconds
- G\_threshold*: 0.3 m/s
- safety\_reaction\_time\_threshold*: 4 seconds
- safety\_intersection\_time\_threshold*: 4 seconds
- safety\_distance\_threshold*: 12 meters
- HD\_Lower\_Threshold*: 1.5 meters
- HD\_Upper\_Threshold*: 4 meters

7.2. Accuracy of the Driving Behavior Detection. First, we evaluated the accuracy of SenSafe in determining driving behaviors around the Beijing University of Posts and Telecommunications. The roads we used for testing are shown in Figure 15. The car we used for the test was Dongfeng Peugeot 307. During these experiments, the smartphones were mounted on the windshield.

We tested the driving behaviors including turn left, turn right, acceleration, brake, single-lane change, multiple-lane change, and turn back. The difference between the single-lane change and multiple-lane change is that multiple-lane change needs more horizontal displacement. The test results are shown in Figure 16. The real number of times for each driving behavior is manually recorded. From the figure we can see that almost all of the turn left, turn right, acceleration,

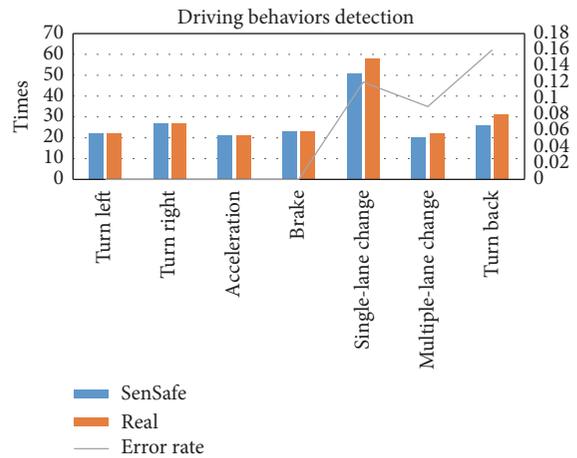


FIGURE 16: Driving behaviors detection results.

and brake have been detected. 88% single-lane change, 91% multiple-lane change, and 84% turn back can be successfully detected.

We furthermore summarized the horizontal displacement and angle change of heading for single-lane change, multiple-lane change, and turning back in Tables 2, 3, and 4.

TABLE 2: Horizontal displacement and angle change of heading for single-lane change.

Displacement	1	2	3	4	5	Average
SenSafe (m)	1.91	1.61	3.06	2.1	2.45	2.226
Real (m)	1.92	1.59	3	1.9	2.27	2.136
Heading angle change	0.6	0.86	1.63	0.66	4.24	1.598

TABLE 3: Horizontal displacement and angle change of heading for multiple-lane change.

Displacement	1	2	3	4	5	Average
SenSafe (m)	4.87	4.83	5.53	9.3	8.67	6.656
Real (m)	4.81	4.84	5.76	9.1	9.77	6.854
Heading angle change	1.98	0.66	1.55	7.76	0.52	2.494

TABLE 4: Horizontal displacement and angle change of heading for turning back.

Displacement	1	2	3	4	5	Average
SenSafe (m)	7.88	7.47	8.09	7.01	8.91	7.87
Real (m)	9.07	7.69	9.51	10.21	7.79	8.854
Heading angle change	179.45	172.4	179.48	183.98	183.86	179.834

The real horizontal displacement is calculated using the speed from the OBD of the vehicles. From Tables 2, 3, and 4 we can see that SenSafe can provide the accurate horizontal displacement and angle change of heading.

**7.3. Performance of Communication.** We test the performance of the communication considering the relationship between the distance and the probability of successfully receiving at least one packet per second.

We used one smartphone to broadcast the beacons and used the other to scan the beacons. The results are shown in Table 5. From the table we can see that, when the distance is less than 30 meters, the client can almost receive at least one packet per second. After the distance is beyond 50 m, the probability has an obvious drop.

**7.4. Performance of Collision Forewarning.** We tested the performance of collision forewarning considering the alert for the brake in front, acceleration in behind, and intersection collision forewarning.

For the brake in front, we used the front vehicle to make the brake to see if the behind vehicle can get the alert. The distance between the two vehicles was around 30 meters. We tested this scenario ten times, and the behind vehicles had gotten the alert ten times.

For acceleration in behind, we used the behind vehicle to make the acceleration to see if the front vehicle can get the alert. The distance between the two vehicles was around 30 meters. We tested this scenario ten times, and the front vehicles had gotten the alert ten times.

TABLE 5: Relationship between the distance and the probability of successfully receiving at least one packet per second.

Distance (m)	Total time (second)	Seconds no packet	Probability
10	1200	0	100%
20	1200	39	97%
30	1200	72	94%
40	1200	234	81%
50	1200	354	71%
60	1200	557	54%

For intersection collision forewarning, we tested it in the campus of Beijing University of Posts and Telecommunications. Considering that the probability of successfully receiving at least one packet per second is less than 80% when the distance is greater than the 40 meters, we add 1 more second in safety reaction time threshold and safety intersection time threshold when the distances between vehicles are greater than the 40 meters. We used two vehicles to meet at the intersection to see if the driver will be reminded of the coming vehicles. The speed of the vehicle was around 6 m/s. One of the vehicles accelerated for 1 second with acceleration around  $1 \text{ m/s}^2$  to trigger the beacon broadcasting. We tested this scenario ten times, and the listening vehicle had gotten the alert nine times. We used one vehicle and a pedestrian to meet at the intersection to see if the driver will be alerted of the coming pedestrian. The speed of the pedestrian was around 2 m/s, and the speed of the vehicle was around 6 m/s. We tested this scenario ten times, and the vehicle had gotten the alert nine times.

## 8. Conclusion

In this paper, we develop a smartphone-based traffic safety framework named SenSafe to sense the surrounding events and provide alerts to drivers. Firstly, a driving behaviors detection mechanism is provided which can run on commodity smartphones. Secondly, the Wi-Fi association and authentication overhead is reduced to broadcast the compressed sensing data using the Wi-Fi beacon to inform the drivers of the surroundings. Thirdly, a collision estimation algorithm is proposed to provide the appropriate warnings. Finally, an Android-based implementation of SenSafe has been achieved to evaluate the performance of SenSafe in real environments, and experimental results show that SenSafe can work well in real environments.

## Competing Interests

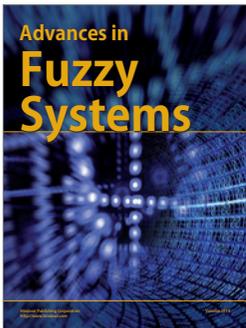
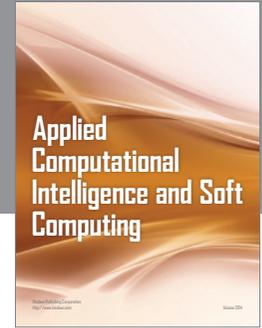
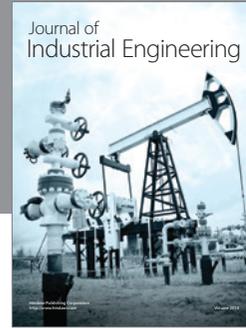
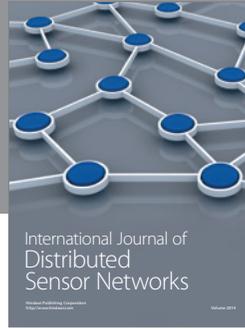
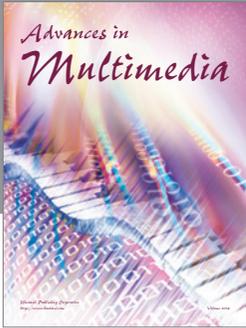
The authors declare that they have no competing interests.

## Acknowledgments

This work was supported by the National Key R&D Program of China (2016YFB0100902).

## References

- [1] Real time traffic accidents statistics, <http://www.icebike.org/real-time-traffic-accident-statistics/>.
- [2] Intelligent Transportation Systems-Dedicated Short Range Communications, <http://www.its.dot.gov/DSRC/>.
- [3] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.
- [4] Applications, <http://www.mobileye.com/technology/applications/>.
- [5] Drivea-driving assistant app, <https://play.google.com/store/apps/details?id=com.driveassist.experimental&hl=en>.
- [6] Turn Your Smartphone Into a Personal Driving Assistant, <http://www.ionroad.com/>.
- [7] Augmented Driving, <https://itunes.apple.com/us/app/augmented-driving/id366841514?mt=8>.
- [8] C.-W. You, N. D. Lane, F. Chen et al., "CarSafe app: alerting drowsy and distracted drivers using dual cameras on smartphones," in *Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*, pp. 13–26, ACM, Taipei, Taiwan, June 2013.
- [9] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "Smartroad: smartphone-based crowd sensing for traffic regulator detection and identification," *ACM Transactions on Sensor Networks*, vol. 11, no. 4, article 55, 2015.
- [10] Z. Wu, J. Li, J. Yu, Y. Zhu, G. Xue, and M. Li, "L3: sensing driving conditions for vehicle lane-level localization on highways," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM '16)*, pp. 1–9, IEEE, San Francisco, Calif, USA, April 2016.
- [11] D. Shin, D. Aliaga, B. Tunçer et al., "Urban sensing: using smartphones for transportation mode classification," *Computers, Environment and Urban Systems*, vol. 53, pp. 76–86, 2015.
- [12] J. Yu, H. Zhu, H. Han et al., "SenSpeed: sensing driving conditions to estimate vehicle speed in urban environments," *IEEE Transactions on Mobile Computing*, vol. 15, no. 1, pp. 202–216, 2016.
- [13] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin, "Sensing vehicle dynamics for determining driver phone use," in *Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*, pp. 41–54, ACM, Taipei, Taiwan, June 2013.
- [14] C. Saiprasert, T. Pholprasit, and S. Thajchayapong, "Detection of driving events using sensory data on smartphone," *International Journal of Intelligent Transportation Systems Research*, 2015.
- [15] D. Chen, K.-T. Cho, S. Han, Z. Jin, and K. G. Shin, "Invisible sensing of vehicle steering with smartphones," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '15)*, pp. 1–13, Florence, Italy, May 2015.
- [16] CohdaWireless, <http://www.cohdawireless.com/>.
- [17] L. Zhenyu, P. Lin, Z. Konglin, and Z. Lin, "Design and evaluation of V2X communication system for vehicle and pedestrian safety," *The Journal of China Universities of Posts and Telecommunications*, vol. 22, no. 6, pp. 18–26, 2015.
- [18] U. Hernandez-Jayo, I. De-La-Iglesia, and J. Perez, "V-alert: description and validation of a vulnerable road user alert system in the framework of a smart city," *Sensors*, vol. 15, no. 8, pp. 18480–18505, 2015.
- [19] W. Sun, C. Yang, S. Jin, and S. Choi, "Listen channel randomization for faster Wi-Fi direct device discovery," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM '16)*, IEEE, San Francisco, Calif, USA, April 2016.
- [20] K. Dhondge, S. Song, B.-Y. Choi, and H. Park, "WiFiHonk: smartphone-based beacon stuffed WiFi Car2X-communication system for vulnerable road user safety," in *Proceedings of the 79th IEEE Vehicular Technology Conference (VTC '14)*, pp. 1–5, IEEE, Seoul, South Korea, May 2014.
- [21] P. Zhou, M. Li, and G. Shen, "Use it free: instantly knowing your phone attitude," in *Proceedings of the 20th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '14)*, pp. 605–616, ACM, Maui, Hawaii, USA, September 2014.
- [22] R. Chandra, J. Padhye, L. Ravindranath, and A. Wolman, "Beacon-stuffing: Wi-Fi without associations," in *Proceedings of the 8th IEEE Workshop on Mobile Computing Systems and Applications (HotMobile '07)*, pp. 53–57, Tucson, AZ, USA, February 2007.
- [23] M. M. Minderhoud and P. H. L. Bovy, "Extended time-to-collision measures for road traffic safety assessment," *Accident Analysis & Prevention*, vol. 33, no. 1, pp. 89–97, 2001.
- [24] K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident Analysis & Prevention*, vol. 35, no. 3, pp. 427–433, 2003.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

