

## Research Article

# Stopping Accidents before They Happen: Perceiving Lane-Level Moving Vehicle Danger Regions to Warn Surrounding Drivers and Pedestrians

Chi Guo,<sup>1</sup> Guangyi Cao,<sup>2</sup> Jieru Zeng,<sup>2</sup> Jinsong Cui,<sup>2</sup> and Rong Peng<sup>3</sup>

<sup>1</sup>Global Navigation Satellite System Research Center, Wuhan University, Wuhan 430072, China

<sup>2</sup>Computer School, Wuhan University, Wuhan 430072, China

<sup>3</sup>State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Jinsong Cui; [jscui@whu.edu.cn](mailto:jscui@whu.edu.cn)

Received 8 May 2015; Revised 29 July 2015; Accepted 29 July 2015

Academic Editor: Jesus Corres

Copyright © 2016 Chi Guo 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.

Perceiving the location of dangerous moving vehicles and broadcasting this information to vehicles nearby are essential to achieve active safety in the Internet of Vehicles (IOV). To address this issue, we implement a real-time high-precision lane-level danger region service for moving vehicles. A traditional service depends on static geofencing and fails to deal with dynamic vehicles. To overcome this defect, we devised a new type of IOV service that manages to track dangerous moving vehicles in real time and recognize their danger regions quickly and accurately. Next, we designed algorithms to distinguish the vehicles in danger regions and broadcast the information to these vehicles. Our system can simultaneously manipulate a mass of danger regions for various dangerous vehicles and broadcast this information to surrounding vehicles at a large scale. This new system was tested in Shanghai, Guangzhou, Wuhan, and other cities; the data analysis is presented in this paper as well.

## 1. Introduction

For the Internet of Vehicles (IOV), vehicle active safety services have become one of the most vital research directions. Active safety focuses on the following two stages, the accident avoidance stage during vehicle crashes [1] and dynamic danger control stage under common driving conditions [2]. The former depends on active safety technologies, such as Dedicated Short Range Communications (DSRC). Traffic data indicates that the probability of vehicle accidents can be cut by more than 80% by deploying onboard navigation systems with dynamic danger control. Thus, danger control should occur before a crash happens; the most important situation therefore is the dynamic danger control stage. This stage could rely on a location service provided by the IOV to cooperating vehicles, meaning that the IOV collects driving status information about dangerous vehicles and transmits this information to surrounding vehicles. In this paper, we address the topic of dynamic danger control.

Up till now, it has only been possible to send static danger information (the location of static dangerous vehicles)

successfully [1, 3], making dynamic danger control an elusive research goal. In China, more than 60% of traffic accidents are related to high-priority vehicles (i.e., passenger vehicles) and high-danger vehicles (i.e., heavy goods vehicles) as Figure 1 shows. We term these two vehicles types as dynamic danger targets.

Thus, our motivation is to effectively control danger by developing in our service the capacity to quickly detect threats stemming from dynamic danger targets' moving statuses, share this information with surrounding vehicles or pedestrians, and guide them safely around the danger targets. The proposed system only monitors dynamic danger targets' moving statuses. Given the large number of vehicles, in order to prevent the IOV from detecting and sharing locational information for all vehicles, we only focus on a limited number of dynamic danger targets. In this way, all users can receive warnings about latent dangers around them while lessening the pressure on the IOV.

Our proposed system efficiently deals with the following two traffic safety situations. (1) When a high-danger target (e.g., dangerous chemical transportation vehicles) appears in

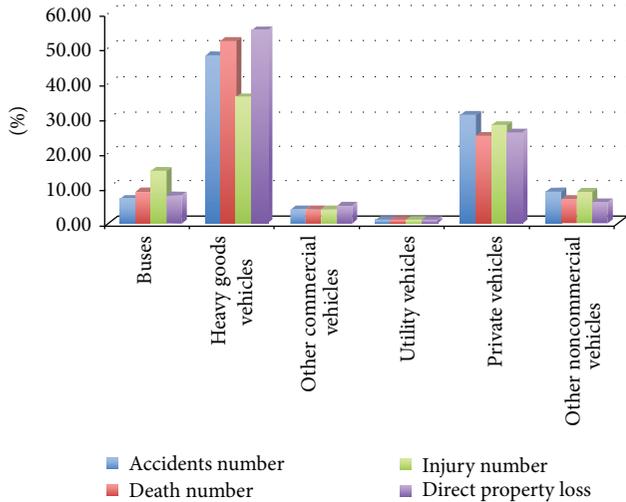


FIGURE 1: Traffic accidents' statistics from the 2010 yearbook of Chinese public security department [2].

front of passenger vehicles, a warning can be sent to nearby drivers. This kind of warning signal can be used to decrease the accident risk in certain situations, such as during long-distance/night driving with low visibility or in relation to hidden dangers. (2) When a high-priority vehicle (i.e., school bus) is in front of a driver, a warning signal will be sent to decrease the risk of traffic accidents.

Sharing safety information about dynamic danger targets with surrounding vehicles requires the precise location of those dynamic danger targets. Thanks to the rapid development of GNSS, it is possible to get precise location at present. We realize lane-level positioning based on BeiDou-Xihe system [4] and Cooperative Real-Time Precise Positioning (CRPP) technology and explain this process in Section 3. In Sections 4–6, we build a model of the region around dynamic danger targets for real-time concurrent control of the danger region by sharing information cooperatively with vehicles in the immediate region.

The contributions of this work are summarized as follows:

- (1) We realize CRPP by means of a lane-level terminal device, and with inexpensive MEMs sensors, we can monitor the locations of multiple dynamic danger targets.
- (2) We propose an IOV active safety model for dynamic danger targets; the implementation structure and active cooperation algorithms for the modeling process are detailed in this paper.
- (3) Our proposed system was tested in Shanghai, Guangzhou, Wuhan, and other cities. Test results show that the IOV model for 71,178 dynamic danger targets was verified for the Guangzhou province IOV system.

## 2. Related Work

Danger control based on the data from GPS has become one of the most vital applications of GPS. Reference [5] proposes

a road safety alerting system with radar and GPS cooperation. Reference [6] designs an improved fast simultaneous localization and mapping system based on distributed structure for autonomous robot navigation. Reference [7] provides an up-to-date overview of emerging state-of-the-art automotive sensor technologies related to danger control which can work as a guideline for the study on the danger control and sensor selection. However, these techniques achieve road-level positioning rather than lane-level positioning with standard GPS. To achieve lane-level positioning, the precision of GPS must be improved.

At present, there has been much research on lane-level positioning and perception of vehicles' moving states. Reference [8] proposes a high-precision-based vehicle network location service to achieve lane-level positioning. Reference [9] introduces the Beidou navigation network that provides preliminary positioning and navigation functions. Reference [10] uses L1 and L2 raw observations and estimates atmospheric delays to achieve high-precision positioning with atmospheric corrections derived from the reference stations. Reference [11] introduces a 3D multisensor integrated navigation system that achieves accurate estimation of inertial sensor errors (biases). This work suggests that it is technically feasible to achieve high-precision positioning and perceive vehicles' moving states through sensors. Additionally, however, we also need a technology to realize the cooperation between vehicles for a successful warning system.

Dynamic danger targets query of related objects can build on the Continuous k-Nearest Neighbor (CkNN) approach. Reference [12] puts forward Euclidean restriction to efficiently get candidate item sets and used Euclidean range query to get the final results. In this way, the performance problem resulting from the extension of road networks is avoided. References [13, 14] propose a continuous query processing framework based on multithreading to periodically calculate the query result and divides the whole process into two parts, offline precomputation and online CkNN query; thus query efficiency was improved by the parallel execution of multithreading processing framework. Considering the relative movements between moving objects and query nodes, [15, 16] come up with query pruning and refining algorithms to reduce the cost of repeated queries and communication. However, CkNN query just involves k-nearest moving objects in road networks but fails to consider the complicated traffic scenes and security semantics of active safety. For instance, moving objects that drive in the same direction with danger targets require more urgent warnings than those driving in the opposite direction.

## 3. Perceiving Lane-Level Position of Dangerous Vehicles with CRPP

*3.1. Lane-Level High-Precision Positioning Service.* Nowadays, GNSS is widely used for positioning. However, due to errors caused by disturbances in the satellite orbits, satellite clocks, and the atmosphere, GNSS positioning precision is typically about 10~20 m, which fails to realize accurate positioning. To improve the precision, our system applies Wide-Area Real-Time Precise Positioning GNSS (WRPP-GNSS). This

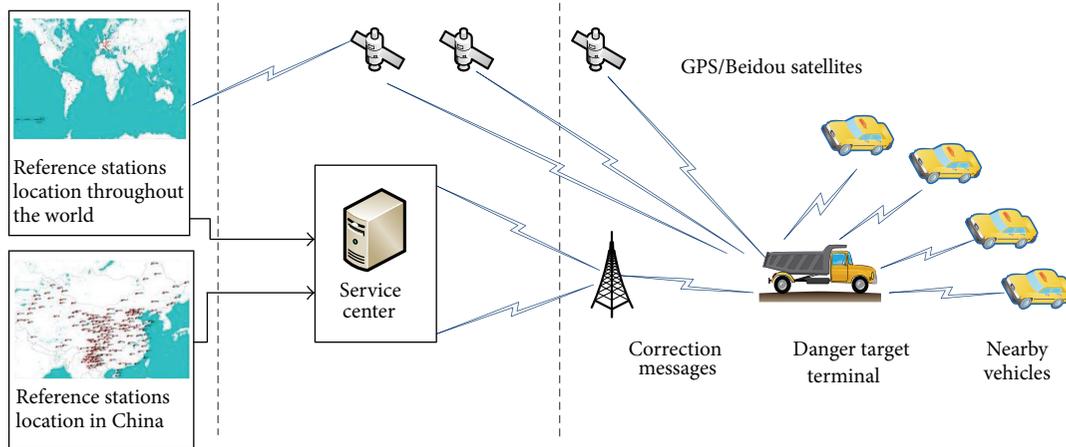


FIGURE 2: Framework of vehicle positioning system.

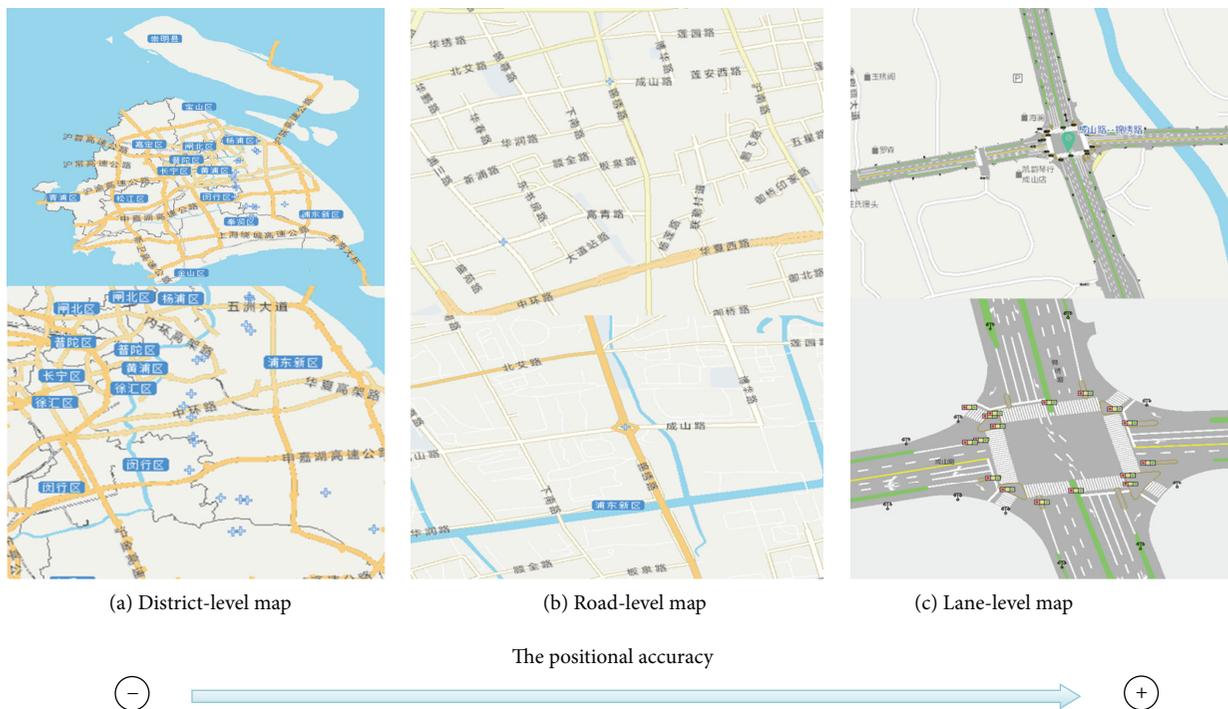


FIGURE 3: The lane-level map service.

approach takes full advantage of satellite observation data from GNSS services and satellite reference stations to model the main positioning errors, sends correction messages for each error source to terminal users, and thus helps positioning terminals enhance location precision.

Our system is capable of providing lane-level positioning of danger targets and ascertaining their danger regions by our algorithms. Then this system broadcasts warning messages to surrounding vehicles of danger targets, which belong to danger regions.

Figure 2 shows the framework of our proposed system. It involves three procedures: (1) get location information provided by GPS/Beidou satellites, (2) make service center collect correction messages which came from WRPP-GNSS (Wide-Area Real-Time Precise Positioning

GNSS) and reference stations throughout the world, especially stations located in China, and (3) combine these two kinds of information together at danger target terminals and terminals automatically broadcast the precise location information to nearby vehicles.

**3.2. Lane-Level Map Service.** Based on lane-level positioning, our system achieves a lane-level map service (Figure 3) to offer lane-level geographic information about dangerous road sections. Compared with traditional maps, our map provides accurate location for each lane on a road, including the location of the lane lines and road signs. In this way, we can perceive the exact lane where the danger target is located in real time, and this is the foundation for the rest of our work.



FIGURE 4: Vehicle movement test. We drove a test car back and forth from Chengshan Road to Jinxiu Road in Pudong new area, Shanghai, on June 23th, 2014.

**3.3. Experiment.** To test how well this positioning service might work, we conducted an experiment in several cities and chose three representative situations to show our test results. Results seen in Figure 4 present two different tracks which comprise red dots and blue dots. These dots stand for the vehicles instantaneous position. Red dots are the positioning results from ordinary GPS, and the blue dots are from WRPP-GNSS. These two tracks are the vehicle moving tracks as recorded by these two different service systems. We find that blue dots are much denser than red dots; the blue track in Figure 4 is much more precise than the red track. Thus, we conclude that our WRPP-GNSS positioning service can precisely perceive vehicle movements in different situations, such as turning, lane changing, and entering waiting zones.

We must achieve lane-level positioning of danger targets to provide precise warnings to vehicles nearby and reduce the processing demand on the IOV. As the statistics of introduction section suggests, it is much more likely for danger targets to have accidents than any other vehicles. Using the lane positioning of a danger target, for instance, a school bus, we can send warnings to vehicles nearby the bus, which in turn cuts down the risk of accidents. Considering the high cost of each precise positioning vehicle terminal as well as the massive number of vehicles in the IOV, it would be prohibitively expensive to equip every vehicle with terminals. Moreover, processing the data from all these vehicles would overwhelm the Internet. Hence, we just equip the limited number of high-priority and high-danger targets and then

share warnings with surrounding vehicles. In this way, not only do we reduce the cost, but also we lessen the pressure on the IOV.

## 4. Figuring out Danger Regions of Dangerous Vehicles

A danger region for vehicles is the region where danger targets are prone to cause accidents. Based on danger regions, we can sort the objects located in this region and push warnings to them. A simple way to define a danger region is to draw a circle around the danger target; however, without considering the road site, this way may push information to vehicles located inside but on a nonadjacent road. Therefore, it is of immense value to define danger region in a more rational way.

### 4.1. Basic Definitions

**Definition 1 (road network).** It is defined as an undirected weighted graph  $G = (N, E)$ , where  $E$  is the set of all edges and  $N$  is the set of all nodes (vertexes of edges) in the road network.

**Definition 2 (edge).** Each nonintersecting road in the road network can be defined as  $e = (ID, snode, enode, width, speedLimit)$ ,  $e \in E$ , where  $ID$  represents the number of the road in the road network,  $snode$  is the start node,  $enode$  is

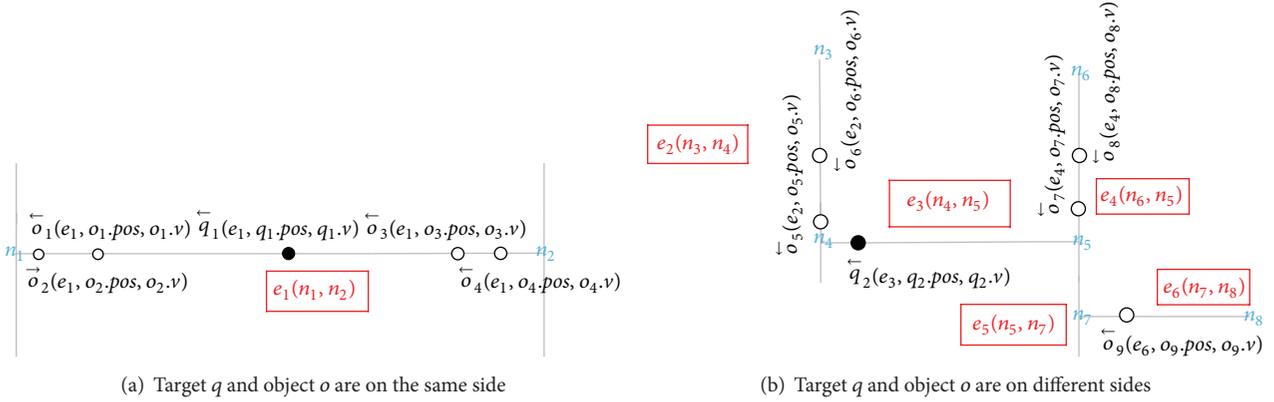


FIGURE 5: The safety distance between danger targets and related objects.

the end node of the road,  $width$  denotes the road length, and  $speedLimit$  denotes the permitted highest velocity.

**Definition 3** (related moving object). Related moving object (object)  $o = (ID, e, pos, velocity)$  refers to the receiver vehicles of traffic security information in a road network.  $o.ID$  is the number of objects in a road network,  $o.e$  is the number of the edge where it lies,  $pos \in [0, e.width]$  is the absolute distance between the object and the start node of the edge,  $velocity$  is positive when the vehicle travels from the start node to end node, and  $O = \{o_1, o_2, \dots, o_n\}$  is the set of all objects in a road network.

**Definition 4** (security event). It is a collection of various security events,  $securityEvent(ID, pos, velocity, securityLevel, Event)$ ;  $ID$  denotes the number of security events,  $pos \in [0, e.width]$  is the distance between the target and start node of the edge where it lies,  $velocity$  represents the current speed (it is positive when the target travels from the start node to end node),  $securityLevel \in \{high, medium, low\}$  means the current security state of the target, and  $Event$  is the concrete description of security event.

**Definition 5** (dynamic danger target). Dynamic danger target (target)  $q (ID, e, securityEvent)$  is a vehicle moving on a road network and sends the warning signal to related moving objects.  $q.ID$  is its number in the road network,  $q.e$  denotes the edge where it lies,  $securityEvent$  is the security event it brings about, and  $Q = \{q_1, q_2, \dots, q_n\}$  is the set of all dynamic danger targets in a road network.

#### 4.2. Security Distance Definition

**Definition 6** (security distance). Suppose that, at time  $t$ , the dynamic danger target  $q$  and related moving object  $o$  are moving at the speed  $q.velocity$  and  $o.velocity$ , respectively, in a road network. If they can meet during time interval  $(t, t + \Delta t]$ , we define the distance between  $q$  and  $o$  as their security distance; that is,  $D_{q,o} = d_{o,q}$ . If not, let the security distance  $D_{q,o} = -1$ .

**Scene 1** ( $q.e = o.e$ ).  $q$  and  $o$  are on the same edge  $e$ . Suppose that  $q_1.velocity < 0$ , which means the danger target  $q_1$  moves

from the ending node  $n_2$  to the beginning node  $n_1$ . Due to the various speeds and locations of  $q$  and  $o$ , Scene 1 can be divided into the four situations about the move relation of target  $q_1(e_1, q_1.pos, q_1.velocity)$  and objects  $o_1 \sim o_4$ , just as Figure 5(a) shows. There are three situations where  $q$  and  $o$  move towards the same direction, so it is possible for  $q$  to meet  $o$  during time interval  $(t, t + \Delta t]$ ; thus, the security distance is  $D_{q,o} = d_{o,q} = |q.pos - o.pos|$ . There is one situation where  $q$  and  $o$  move in opposite directions;  $q$  and  $o$  will not meet each other during time interval  $(t, t + \Delta t]$  so the security distance is  $D_{q,o} = -1$ .

**Scene 2** ( $\{e_q.snode, e_q.enode\} \cap \{e_o.snode, e_o.enode\} \neq \emptyset$ ). This means that there is at least one common node between the edge that danger target  $q$  located and the edge that moving object  $o$  located. Scene 2 has four different moving patterns; they are (i)  $q_2$  and  $o_5$ , (ii)  $q_2$  and  $o_6$ , (iii)  $q_2$  and  $o_7$ , and (iv)  $q_2$  and  $o_8$ ; and these objects' locations and moving directions are shown in Figure 5(b). Regarding (i), (ii), and (iii), during the time interval  $(t, t + \Delta t]$ , they may meet each other. So we get security distance  $D_{q,o} = d_{q,n_{com}} + d_{o,n_{com}}$ . In the fourth situation,  $q$  will not meet  $o$ ; thus  $D_{q,o} = -1$ .

**Scene 3** ( $\{e_q.snode, e_q.enode\} \cap \{e_o.snode, e_o.enode\} = \emptyset$ ). As shown in Figure 5(b), there are not any common nodes between edge  $e_3$  where  $q_2$  lies and edge  $e_6$  where  $o_9$  lies. So the security distance is  $D_{q,o} = -1$ .

#### 4.3. Danger Region Definition

**Definition 7** (danger region). We start with each  $q$ ; define a danger region  $R_q$  along with road networks to ensure the security distance between  $q$  and each moving object outside  $R_q$  is more than the distance threshold  $D_{safe}$  ( $D_{q,o} > D_{safe}$ ); the security distance between  $q$  and each moving object inside  $R_q$  is less than the distance threshold ( $D_{q,o} \leq D_{safe}$ ); additionally,  $D_{q,o} \neq -1$ .

**4.4. The Danger Region Computing Process.** In a real system, in order to simplify the computation and management of the IOV when computing danger regions, we grid the road networks to transform the calculation of the distance between two points into a calculation of the distance between two cells.

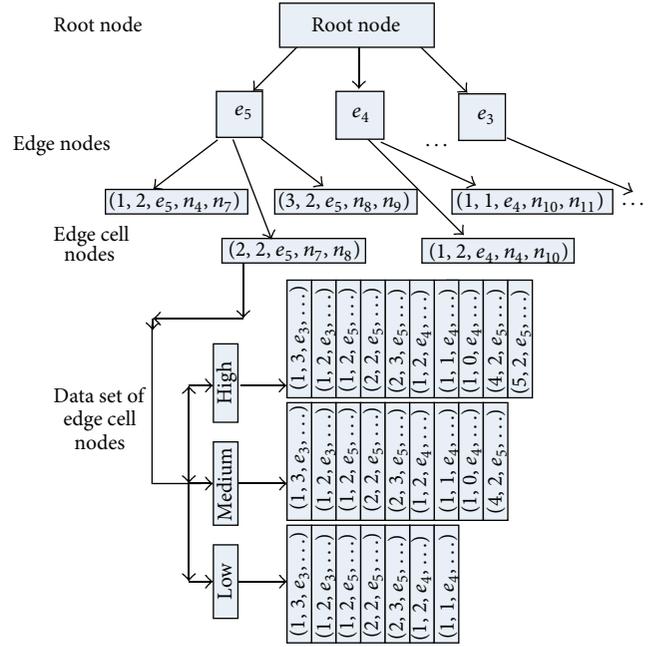
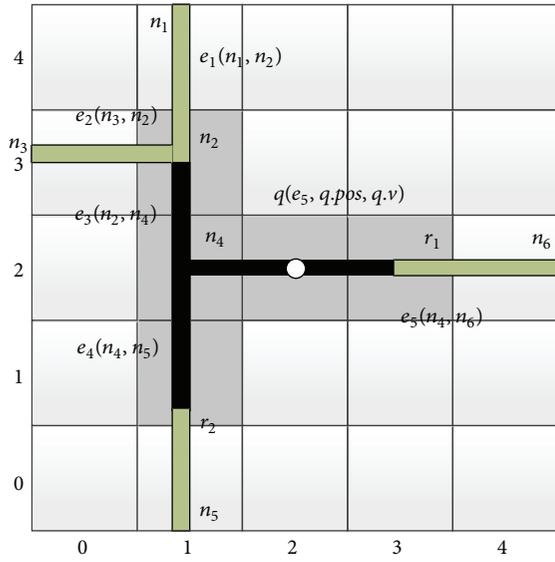


FIGURE 6: Edge cell and danger region.

**Definition 8** (edge cell). When a certain cell  $[x, y]$  only maps to one edge  $e_i$  of  $E$ , this cell is an edge cell of  $e_i$  and we define the edge cell as  $eCell(x, y, e, n_s, n_e)$ , where  $x$  and  $y$  are its coordinates,  $e$  is the edge where it is located, and  $n_s$  and  $n_e$  are the two intersections of  $e$  and the edge cell. When a cell maps to more than one edge, it is regarded as an edge cell of all these edges. For instance, a grid  $[2, 1]$  includes three edges,  $e_1$ ,  $e_2$ , and  $e_3$ , and is divided into three edge cells,  $[2, 1, e_3]$ ,  $[2, 1, e_2]$ , and  $[2, 1, e_1]$ .

After defining the edge cell, we compute the danger regions for each danger target. However, to explain this computation process clearly, we first consider the situation without gridded road networks. For example, Figure 6 shows a simple road network; the danger target  $q$  lies on the edge  $e_5$ . Suppose the distance threshold is  $D_{safe}$ , and we set  $d_{q,r_1} = D_{safe}$ ,  $d_{q,n_4} + d_{n_4,r_2} = D_{safe}$ , and  $d_{q,n_4} + d_{n_4,n_2} < D_{safe}$ . Although  $d_{q,n_4} + d_{n_4,n_2} < D_{safe}$ , we need not calculate the following two edges,  $e_1$  and  $e_2$ , because they do not have any common nodes with edge  $e_5$ , so we stop at node  $n_2$ . Thus, the danger region of  $q$  is the black area in Figure 6. According to this process, we reach the conclusion that the danger region is limited in the adjacent edges, and any locations in this region are less than  $D_{safe}$  from the danger target,  $R_q = \{p(x, y) \mid (d_{p,q} \leq D_{safe}) \wedge (e_q \cap e_p \neq \emptyset)\}$ .

Now, we consider the computation of danger region with the gridded road network. Instead of finding the adjacent edges, we need to find the adjacent edge cells; instead of computing the distance of two points, we need to compute the distance of two edge cells. As seen in Figure 6, the gray edge cells are the danger region of  $q$ .

In the model implementation, the danger regions  $R_q$  of moving danger targets only depend on road network  $G$  and distance threshold  $D_{safe}$ . So the computation of  $R_q$  can be

preprocessed in order to provide indexing for the following query. When the data is initialized, we set the edge cell where danger target  $q$  is located as  $eCell$ ; the computation process for  $R_q$  is described by Algorithm 1.

In Algorithm 1, the edge  $e$  has common nodes with the edge where  $q$  is located and transfers  $e$  into the mapping edge cells set  $eCellList$ . Then it calculates the distances between  $eCell$  and each cell  $ec$  of  $eCellList$ ; if the distance is less than  $D_{safe}$ , the cell  $ec$  will be put into  $R_q$ . In Algorithm 1, the edge length is set as  $l$ , the grid cell length is  $a$ , the number of edges is  $k$ , and the number of edges next to the edge where the current edge cell locates is  $r$ ; therefore the time complexity of Algorithm 1 is  $O(klr/a)$ .

For each danger target  $q$ , the larger the  $q.securityLevel$  is, the wider the pushing regions are. During the initializing phase, we compute corresponding  $R_q$  when  $q.securityLevel$  is *low*, *medium*, and *high* in advance; thus, in the implementation, we just need to change  $D_{safe}$ ; the higher  $q.securityLevel$  is the larger  $D_{safe}$  is. We store the mapping table of edge cells and danger areas in *indexTree* with B+ tree to build indexing as the right side of Figure 6 suggests.

## 5. Warning the Surrounding Vehicles and System Implementation

In practical applications, danger targets are always moving on a road network. As a consequence, danger regions are changing all the time. We calculate more than one danger region on a road network.

### 5.1. Dynamic Danger Region Definition

**Definition 9** (dynamic danger region). For every edge cell the danger target goes through within a certain time  $\Delta t$ ,

```

Input:  $eCell, G, D_{safe}$  //  $eCell$  is the edge cell for computation
Output:  $R_q$ 
(1) FOR  $e$  IN  $E$  DO
(2) IF  $\{e.snode, e.enode\} \cap \{eCell.e.snode, eCell.e.enode\} \neq \emptyset$ 
(3) // To get the mapping edge cell set  $eCellList$  of  $e$ :
(4)  $eCellList = getECellList(e)$ 
(5) FOR  $ec$  IN  $eCellList$  DO
(6)  $d_1 = distance(eCell.n_s, ec.n_s)$ 
// Computing the distance
(7)  $d_2 = distance(eCell.n_s, ec.n_d)$ 
(8)  $d_3 = distance(eCell.n_d, ec.n_s)$ 
(9)  $d_4 = distance(eCell.n_d, ec.n_d)$ 
(10)  $distance(ec, eCell) = \min(d_1, d_2, d_3, d_4)$ 
(11) IF  $(distance(ec, eCell) < D_{safe})$  DO
(12) Put  $ec$  into  $R_q$ 
(13) END IF
(14) END FOR
(15) END IF
(16) END FOR
(17) RETURN  $R_q$ 

```

ALGORITHM 1: The computation of the danger areas.

we calculate its danger regions and work out the union set of these regions; the dynamic danger regions depend on magnitude and direction of the velocities of danger targets as well as the time period  $\Delta t$ .

In order to manage dynamic danger regions, we need to query more than one dynamic danger target in real time and push the corresponding information to the related moving objects in road networks. The whole management process can be divided into two parts, the query phase and pushing phase, as in Figure 7. We applied the MapReduce computing framework in the system. In the map phase, we allocated a computing node for every danger target  $q$  to execute the query phase, get their dynamic danger regions based on their moving location and the preprocessed mapping table of edge cells and danger regions then filter out the moving objects that do not meet the pushing conditions from the current dynamic danger regions, and obtain the moving objects set  $OPush$  that needs to be pushed with information. Through the execution of query phase, we get several key-value pairs  $(q_i, OPush_i)$ . The reduce phase matches the pushing phase in the algorithm, where  $q_i$  are sorted to achieve prioritized pushing. The whole process is presented in Algorithm 2.

We focused on how to ensure the moving objects are set  $OPush$  for each danger target  $q$  in the map phase. The following two sections detail two subalgorithms of the map phase:  $getDynamicDangerRegion()$  and  $objectFilter()$ .

**5.2. Dynamic Danger Region Computation in Real Time.** The main aim of query phase is to express the dynamic danger regions during the time period  $[t, t + \Delta t]$  in the sets of edge cells. The concrete steps are seen in Algorithm 3. First, calculate moving range of the danger target  $q$  in  $[t, t + \Delta t]$  based on the magnitude and direction of velocity of  $q$ , transfer the moving range into the set of edge cells, and then, for edge

cells of the set, search out their own edge cells sets of danger regions by indexing the mapping table; next filter out the edge cells that do not have any common nodes with the edge where  $q$  locates at time  $t$ ; thus the result is dynamic danger region  $dynamicECellList$  of  $q$  during the time period  $[t, t + \Delta t]$ . In Algorithm 3, the edge length is set as  $l$ , the grid cell length is  $a$ , the number of edges is  $k$ , and the number of adjacent edges to the edge where the current edge cell is located is  $r$ ; therefore, the time complexity of rows (4)~(17) is  $O(klr/a)$  and for rows (18)~(27) it is  $O((klr/a)\log((klr/a)))$ .

**5.3. Filter out the Information Pushing Objects from the Dynamic Danger Region.** The filter algorithm has two major functions. The first one is to find out the moving objects that locate in those edge cells returned from Algorithm 3. The second function prevents pushing traffic information too frequently towards the related objects around the target  $q$ .

**5.4. System Implementation.** In this paper, active cooperation and information pushing methods for dynamic danger targets were used as plug-in software on the IOV real-time monitoring platform. An experimental deployment was conducted on the traffic management platform in Zhongshan, Guangdong. This platform supplies the latest data for road networks and was the data source for the index. This platform was accessed through the monitoring terminal or the moving objects, so we can collect the locations of vehicles at any time. Based on this, our algorithm is able to update the location of danger targets and the mapping relationship between edge cells and moving objects. After the query phase, our algorithm sends the set of pushing objects to the platform so that the platform can communicate with corresponding vehicles. The system architecture and the running interface are shown in Figure 8.

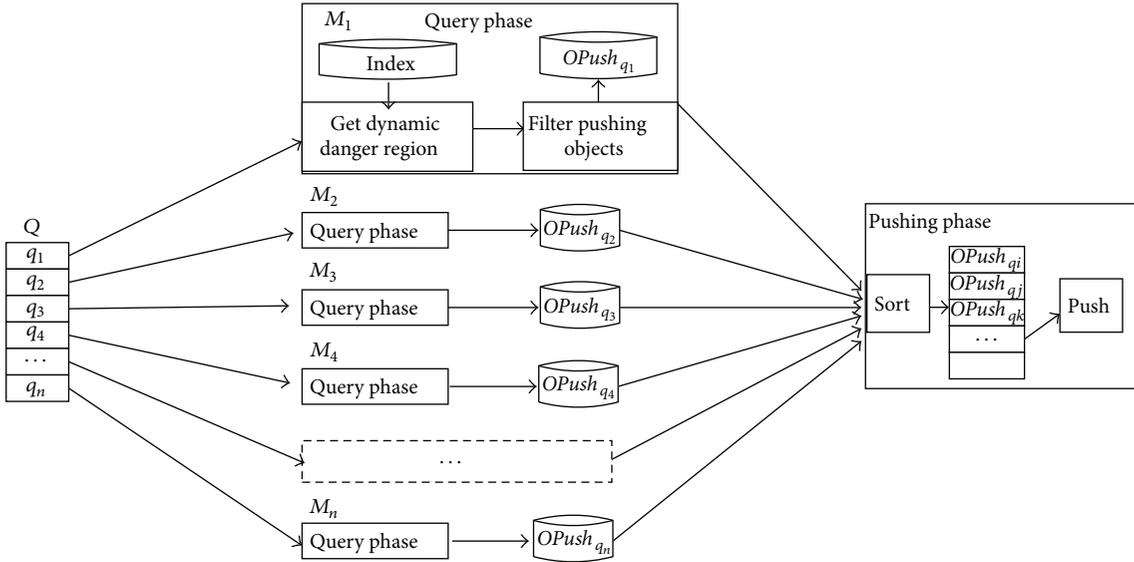
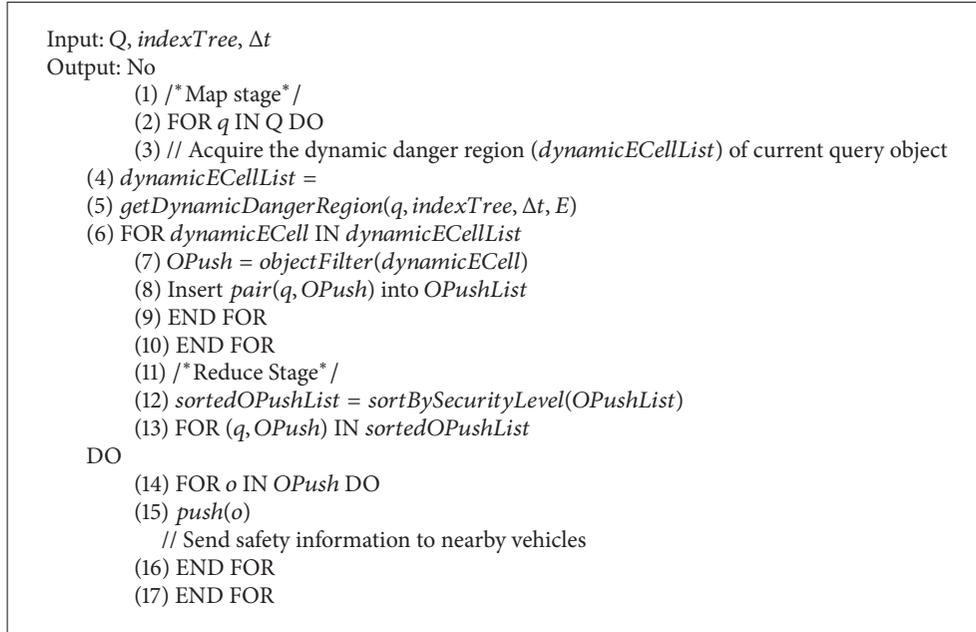


FIGURE 7: MapReduce computing framework.



ALGORITHM 2: Running algorithm.

## 6. Experiments and Simulation

6.1. *System Data, Experiments, and Analysis.* The traffic management platform in Zhongshan, Guangdong, provides us with a counterpart node. Based on this counterpart, we carried out a series of experiments to estimate the accuracy and efficiency of the whole active safety service. The targets of the experiments are 71178 vehicles that have accessed to this system; the types of vehicles are shown in Figure 9. We made simulation tests once for 300 dynamic danger targets among these vehicles, gridded the road network according to various edge lengths, and computed the index of danger region for each edge cell based on the model presented in

Section 4. When security levels were high, medium, and low, the scales of danger region were 500 m, 800 m, and 1000 m, respectively.

6.1.1. *Accuracy Analysis.* Active safety warning depends on the accuracy of the selection of related objects highly associated with the layout of road networks and the relationship among dynamic targets. We define the accuracy of active safety warning as follows:

$$\frac{\sum X_i}{\sum K_i}, \quad (1)$$

```

Input:  $q$ ,  $indexTree$ ,  $\Delta t$ ,  $E$ 
Output:  $dynamicECellList$ 
(1) // Initialize  $R_m$ ,  $R_m$  is the edge cell set of these edges that  $q$  goes by during  $\Delta t$ 
(2)  $R_m = \emptyset$ 
(3) // Compute  $q$  moving region  $d_m$  during  $\Delta t$ 
(4)  $d_m = q.velocity * \Delta t$ 
(5) // Compute  $R_m$ 
(6) FOR  $e$  IN  $E$  DO
(7) IF  $\{e.snode, e.enode\} \cap \{eCell.e.snode, eCell.e.enode\} \neq \emptyset$ 
(8) // Get  $e$ 's  $eCellList$ 
(9)  $eCellList = getECellList(e)$ 
(10)   FOR  $ec$  IN  $eCellList$  DO
(11)     IF  $distance(ec, eCell) < d_m$ 
(12)       Insert  $ec$  into  $R_m$ 
(13)     END IF
(14)   END FOR
(15) END IF
(16) END FOR
(17)  $dynamicECellList = \emptyset$ 
(18) FOR  $e_m$  IN  $R_m$  DO
(19)    $tempDynamicECellList = searchIndex(e_m, indexTree)$ 
(20)   FOR  $e_d$  IN  $tempDynamicECellList$  DO
(21)     IF  $\{e_d.snode, e_d.enode\} \cap$ 
(22)        $\{eCell.e.snode, eCell.e.enode\} \neq \emptyset$ 
(23)       Insert  $e_d$  into  $dynamicECellList$ 
(24)     END IF
(25)   END FOR
(26) END FOR
(27) RETURN  $dynamicECellList$ 

```

ALGORITHM 3:  $getDynamicDangerRegion()$ .

where  $X_i$  is the number of related objects that are returned from system query of danger targets before the road network gridded and  $K_i$  is the number of related objects returned from system query of danger targets after the road network gridded. In order to examine the impact of the accuracy of system query result with road network gridded, we chose the peak hour in Zhongshan, from 8:00 am to 9:00 am, taking 600 times queries for the danger targets at the rate of 1/6 times per second.

Figure 10 shows us that when the warning time is 30 s, 150 s, and 210 s, the system accuracy varies with different lengths for edge cells. We can learn from the result that no matter how long the length of an edge cell is the system accuracy can be kept above 80%. When the length is 50 meters, we get the most accurate output. As the length of an edge cell increases, there will be more mixing of objects and vehicles in an edge cell leading to a decline of system accuracy. Another thing we can learn from Figure 10 is the fact that, with the increase in the warning time, accuracy decreases. Just as Figure 10 shows, when the warning time is 210 seconds, there is no significant difference in system accuracy with various lengths of edge cells. In addition, when warning time is short, the accuracy is influenced so much that the error limit becomes quite obvious. When the warning time is long, the accuracy remains stable. The reason for this phenomenon is the fact that the shorter the warning time is the smaller the dynamic danger regions are, so the related

objects are much fewer and the change in accuracy is more evident.

**6.1.2. Efficiency Analysis.** Extending the range of edge cell helps to enhance the query efficiency of related vehicle objects surrounding dynamic danger targets so as to meet the demand of response time of active safety location-based services for vehicles' normal driving phases. We made a simulation experiment in Zhongshan during the peak period, 8:00-9:00, and test the danger targets 1/6 times per second.

Figure 11 shows that when the length of edge cell changes, different warning times impact the time for system queries and the number of related objects returned. We find that as warning time increases more time is spent by the system to complete the analysis and computation of 300 dynamic danger regions, while more related objects are returned from the computation. Comparing 50 m, 125 m, and 250 m edge cells, the query time is the shortest when the length of edge cell is 50 m. This is because a shorter edge cell length means the road network is divided into more small cells. Though the system uses MapReduce framework in the parallel calculation of each dynamic danger targets, since there are more edge cells, more data is required to pass to each single computing unit, thus contributing to a longer system query time. When the length of edge cell is longer, the system query time does not increase with the warning time. At the same time, it is easily observed that as for different lengths of edge cells

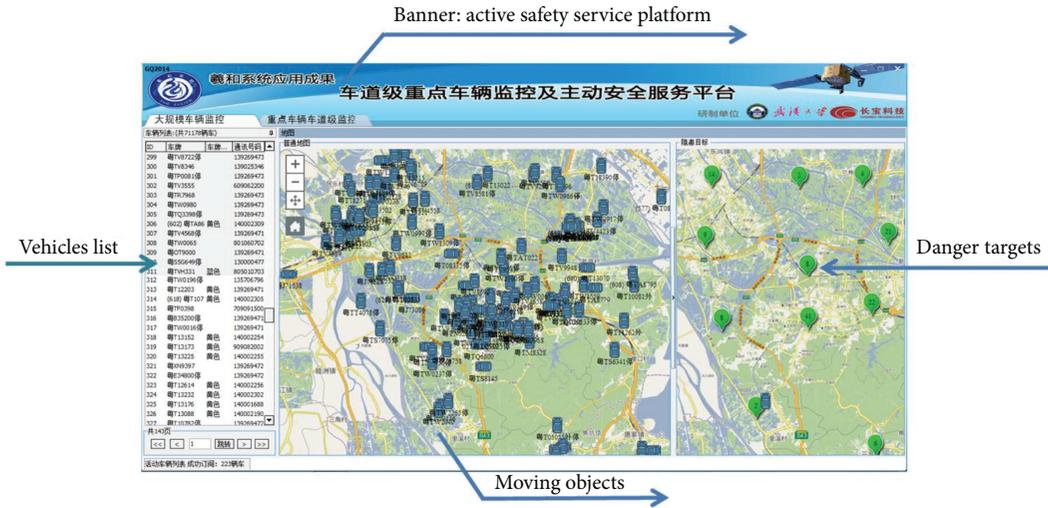


FIGURE 8: System architecture and the running interface.

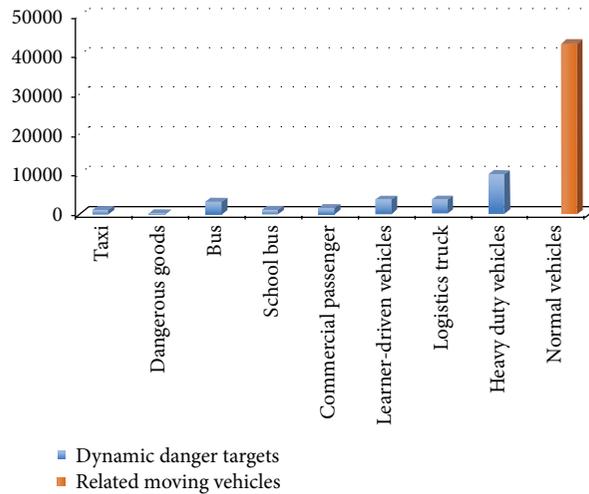


FIGURE 9: Experimental data.

the number of related objects returned from the system query makes no significant difference. We can properly select the relevant parameter in practice when these findings are combined with the analysis of warning accuracy described in Section 5.1.

6.1.3. *Gridding Effect Analysis.* Gridding is an important implement for our experiment; therefore it is necessary to analyze the effect of gridding. Figure 12 shows the system query time and the number of returned objects without gridding. The query time is little changed with different warning times. In comparison with Figure 11, the query time shown in Figure 12 is notably longer than the query time with gridding. Thus, it can be concluded that after gridding the network system query efficiency is significantly enhanced with little accuracy loss.

6.2. *Simulation.* In order to further extend our system to a larger IOV platform so that the active safety warning services can be provided for more vehicles, we carried out a numerical simulation on the extended capabilities of the system. The simulation experiment was based on the single server and single node computation. The experimental environment included an Intel Xeon E3-1230 v2 CPU, 16 GB of memory, Linux Debian 3.13-1-amd64 system operation, and an ext4 file system as well as Java programming language. The NoSQL database used for storage was MongoDB. In this experiment, we constructed a road network consisting of 5521 nodes and 8980 edges and created a moving objects generator to produce dynamic targets and related objects. The velocity of both the dynamic targets and related objects was set at 40–140 km/h. As for the security level of dynamic targets, 10% of them were set to high, 20% were set to medium, and the rest were set to low.

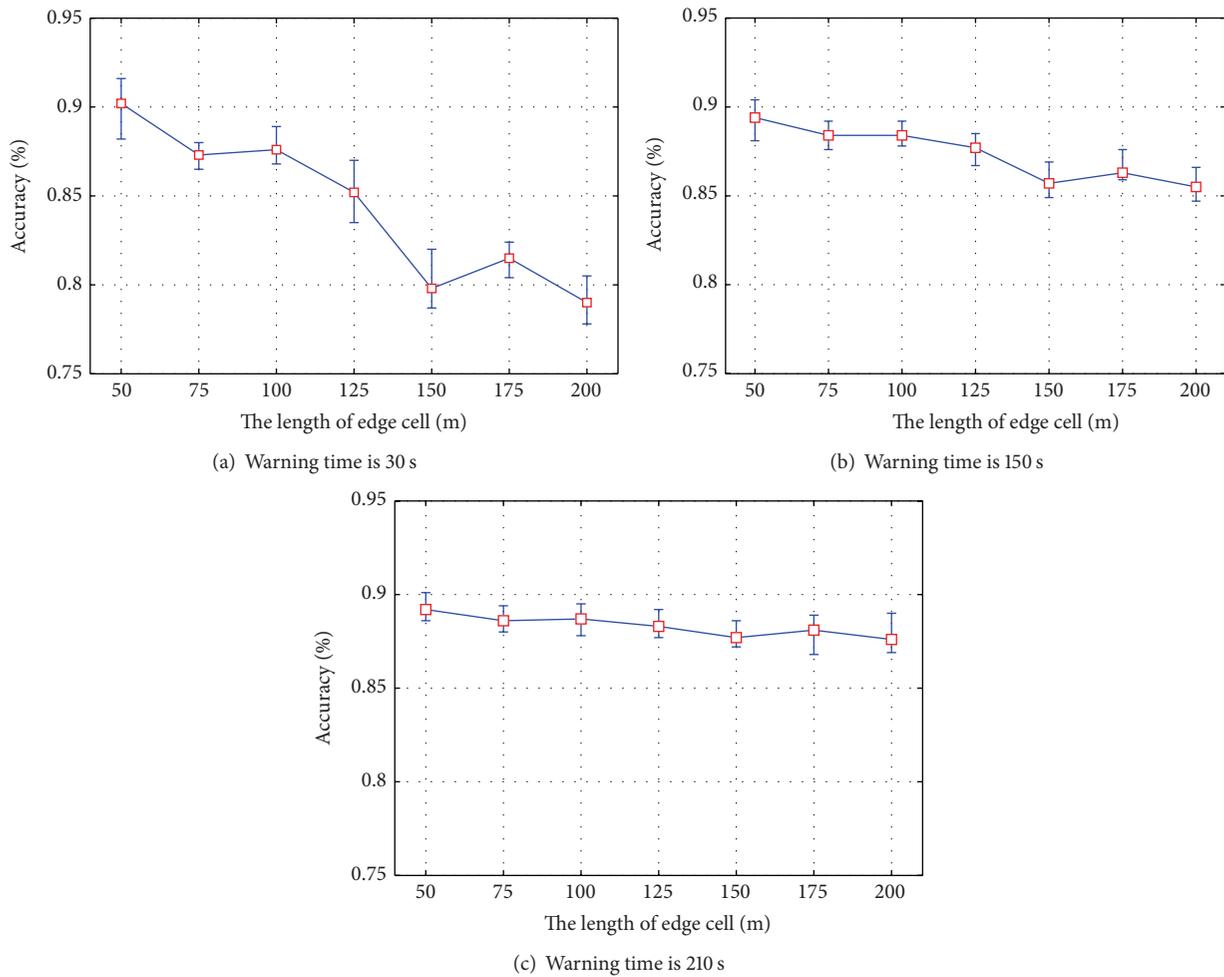


FIGURE 10: The system accuracy varies with different length of edge cell.

The selection of parameters is essential under various road and traffic situations. Consider the following situation. A vehicle drives at 100 km/h on the highway; according to the “Specification for Layout of Highway Traffic Signs and Markings” [17], the advance exit sign and lane guide marking should be set three kilometers away from an interchange on the highway. It determined the warning time for the vehicle as 120 s. From Figures 10 and 11, we can see the 120 s warning time promises query time and system accuracy a sufficient level. Moreover, in Figure 11, the query time with a 100 m edge cell length is significantly less than that with 50 m cell length. In Figure 11, the system accuracy with 100 m cell length is higher than those with a 200 cell length. Based on above conclusions, we selected 100 m as the cell length in our experiment.

In the experiment, the length of edge cell was limited to 100 m and the warning time was set at 120 s. Figure 13 shows the simulated effects of different numbers of dynamic targets and related objects on the query time. We find that more query time is spent as the number of dynamic targets and related objects increases. However, with 800 dynamic targets and 200,000 related objects, the time for the system to update remains low, 2.5 s. Therefore, we conclude that the system is equipped with an appropriate extended capability.

## 7. Conclusions

This paper proposes an active safety model for the IOV, focusing on dynamic danger targets consisting of high-danger and high-priority-of-way vehicles. We elaborate a relevant modeling process, implementation structure, and active cooperation algorithm. We use these proposed solutions to monitor all kinds of dynamic danger targets in traffic and facilitate cooperation among surrounding related objects.

According to the results of verification and simulation experiments, our method can track and query dynamic danger targets on a large scale, providing active safety services for vehicles in dynamic danger regions. Relevant achievements were brought into the national BeiDou-Xihe research project.

## Disclosure

Chi Guo is the first author.

## Conflict of Interests

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

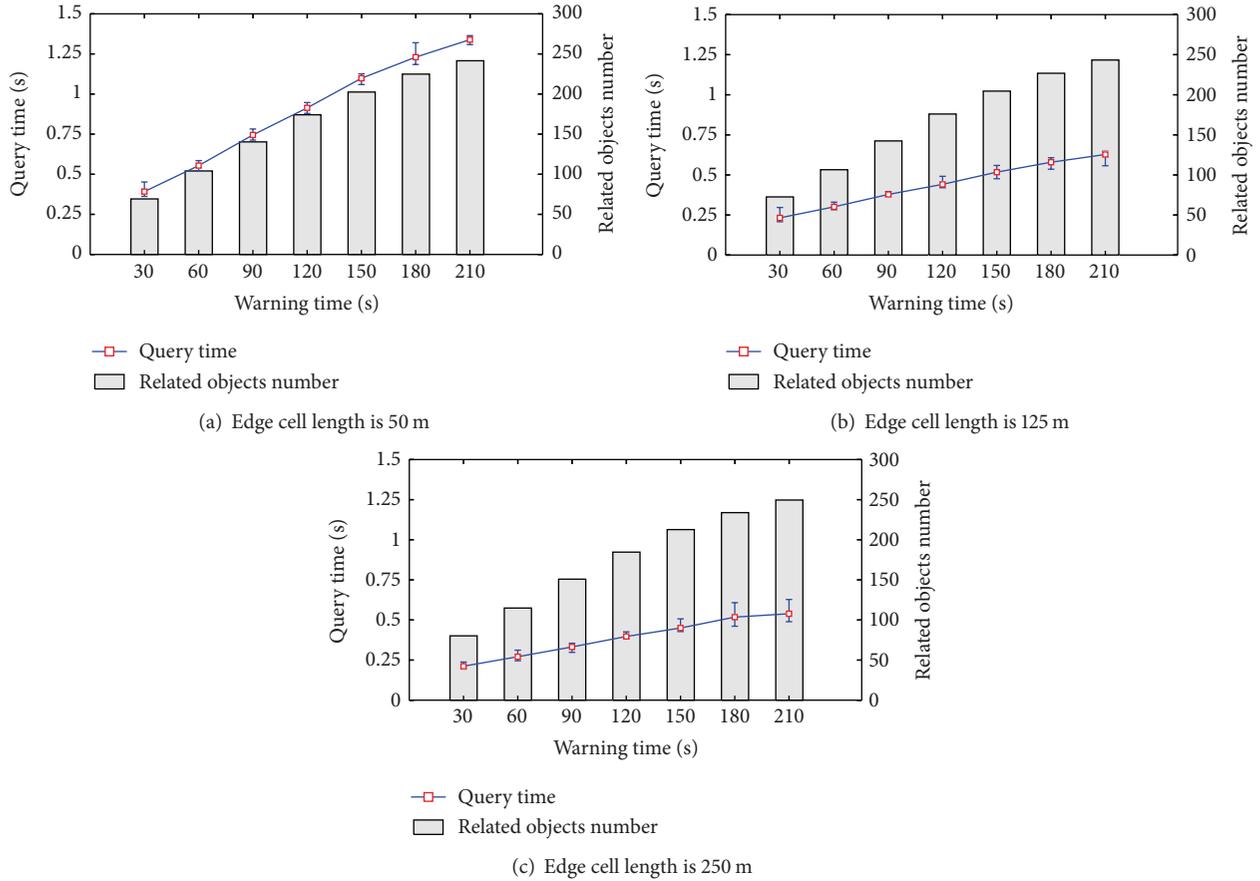


FIGURE 11: The analysis efficiency under 300 dynamic danger regions with different warning time.

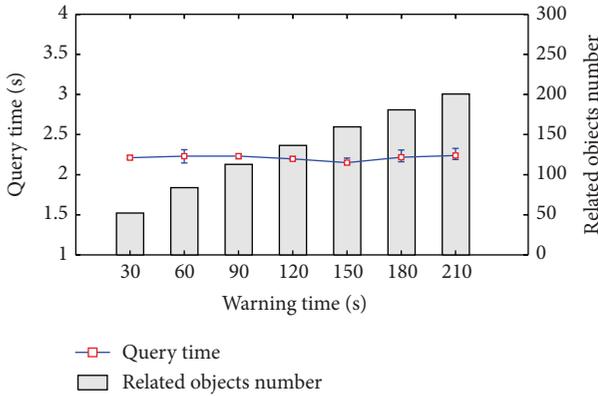


FIGURE 12: The analysis efficiency with different warning time without gridding.

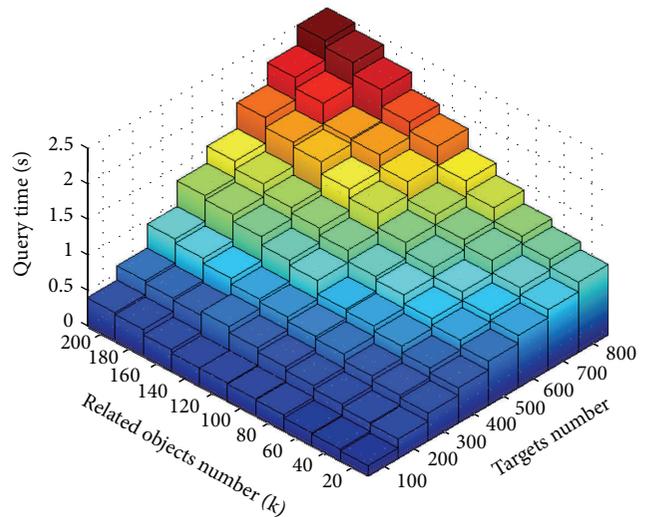


FIGURE 13: The simulation results of different number of dynamic targets and related objects.

### Acknowledgment

The authors thank Professor Lou Yidong and postgraduate ChenXi for their help. This paper is supported by (1) The National High Technology Research and Development Program (863 Program) major project (2013AA12A206, 2015AA124100), (2) The program of Introducing Talents of Discipline to Universities (B07037), and (3) National Natural

Science Foundation of China research project: social network awareness and security cooperation technologies in location-based services (41104010).

## References

- [1] C.-L. Huang, Y. P. Fallah, R. Sengupta, and H. Krishnan, "Adaptive intervehicle communication control for cooperative safety systems," *IEEE Network*, vol. 24, no. 1, pp. 6–13, 2010.
- [2] Traffic Management Bureau of Ministry of Public Security, *Statistics of Road Traffic Accidents of PRC*, Traffic Management Bureau of Ministry of Public Security, Beijing, China, 2013, (Chinese).
- [3] A. Nait-Sidi-Moh, W. Ait-Cheik-Bihi, M. Bakhouya, J. Gaber, and M. Wack, "On the use of location-based services and geofencing concepts for safety and road transport efficiency," in *Trends in Mobile Web Information Systems*, vol. 183 of *Communications in Computer and Information Science*, pp. 135–144, Springer, 2013.
- [4] Ministry of Science and Technology of PRC, *Technical Specification White Paper in Xihe System*, 2014, (Chinese).
- [5] A. F. Santamaria, C. Sottile, F. De Rango, and M. Voznak, "Road safety alerting system with radar and GPS cooperation in a VANET environment," in *Wireless Sensing, Localization, and Processing IX*, vol. 9103 of *Proceedings of SPIE*, May 2014.
- [6] F.-J. Pei, H.-Y. Li, and Y.-H. Cheng, "An improved FastSLAM system based on distributed structure for autonomous robot navigation," *Journal of Sensors*, vol. 2014, Article ID 456289, 9 pages, 2014.
- [7] Md. S. Amin, S. S. Nasir, M. B. I. Reaz, M. A. M. Ali, and T.-G. Chang, "Preference and placement of vehicle crash sensors," *Tehnicki Vjesnik-Technical Gazette*, vol. 21, no. 4, pp. 889–896, 2014.
- [8] C. Guo, W. Guo, G. Cao, and H. Dong, "A lane-level LBS system for vehicle network with high-precision BDS/GPS positioning," *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 531321, 13 pages, 2015.
- [9] C. Shi, Q. Zhao, Z. Hu, and J. Liu, "Precise relative positioning using real tracking data from COMPASS GEO and IGSO satellites," *GPS Solutions*, vol. 17, no. 1, pp. 103–119, 2013.
- [10] M. S. Amin, S. S. Nasir, M. B. I. Reaz, M. A. Mohd. Ali, and T.-G. Chang, "Preference and placement of vehicle crash sensors," *Tehnički Vjesnik*, vol. 21, no. 4, pp. 889–896, 2014.
- [11] X. Li, M. Ge, J. Douša, and J. Wickert, "Real-time precise point positioning regional augmentation for large GPS reference networks," *GPS Solutions*, vol. 18, no. 1, pp. 61–71, 2014.
- [12] M. M. Atia, T. Karamat, and A. Noureldin, "An enhanced 3D multi-sensor integrated navigation system for land-vehicles," *Journal of Navigation*, vol. 67, no. 4, pp. 651–671, 2014.
- [13] U. Demiryurek, F. Banaei-Kashani, and C. Shahabi, "Efficient continuous nearest neighbor query in spatial networks using Euclidean restriction," in *Advances in Spatial and Temporal Databases*, vol. 5644 of *Lecture Notes in Computer Science*, pp. 25–43, Springer, Berlin, Germany, 2009.
- [14] L. Zhao, L. Chen, N. Jing, and W. Liao, "Continuous k nearest neighbor queries of moving objects in road networks," *Chinese Journal of Computers*, vol. 33, no. 8, pp. 1396–1404, 2010 (Chinese).
- [15] L. Zhao, N. Jing, L. Chen, W. Liao, and Z.-N. Zhong, "Continuous K nearest neighbor queries over moving objects based on multi-core and multi-threading," *Journal of Software*, vol. 22, no. 8, pp. 1805–1815, 2011 (Chinese).
- [16] P. Fan, G. Li, and L. Yuan, "Continuous K-Nearest Neighbor processing based on speed and direction of moving objects in a road network," *Telecommunication Systems*, vol. 55, no. 3, pp. 403–419, 2014.
- [17] G. Li, P. Fan, Y. Li, and J. Du, "An efficient technique for continuous K-nearest neighbor query processing on moving objects in a road network," in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology (CIT '10)*, pp. 627–634, July 2010.



**Hindawi**

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

