IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 1, 2020, accepted June 22, 2020, date of publication June 30, 2020, date of current version July 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3006073

Low-Cost Urban Test Scenario Generation Using Microscopic Traffic Simulation

BINGJIAN YUE, SHUMING SHI, SHUO WANG, AND NAN LIN
Transportation College, Jilin University, Changchun 130052, China

Corresponding author: Nan Lin (linnan@jlu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant U1964202.

ABSTRACT Scenario-based testing is already a well-known test approach to the automotive industry for the Validation, Verification and Testing of Connected and Automated Vehicles. How to construct the scenarios of complex traffic environment is a major challenge to this method. A common approach for generating scenarios is collecting and post-processing natural traffic data with a lot of time and money costs. In order to reduce the cost of scenario collection, we propose a low-cost method based on Microscopic Traffic Simulation to obtain a large number of urban traffic scenarios. Based on public data, we establish a microscopic traffic model for a specific area within the Shenzhen urban. Through a simulation, the 24-hour traffic behavior of vehicles for this area is simulated. About 189,752 scenarios covering the entire travel process are generated, which is equivalent to the data collected by a well-equipped car traveling 254,480 kms or 8,288 h. Our scenarios include not only typical urban scenarios such as U-turn and Parallel-driving (two vehicles driving in parallel on a single lane), but also collision accident scenarios of various forms. In addition, in order to evaluate the risks faced by Connected and Automated Vehicles in different scenarios, we design a new criticality metric, Scenario Risk Index, based on the risk assessment principle. The Scenario Risk Index has nothing to do with a certain function, and can quantitatively comprehensively evaluate the criticality and loss of potential accidents.

INDEX TERMS Test scenario, microscopic traffic simulation, traffic conflict technique, risk assessment.

I. INTRODUCTION

Connected and automated vehicles (CAVs) provide new ways to prevent traffic accidents, save travel time and reduce emissions. Validation, Verification and Testing (V.V&T) is a key step in the development and deployment of CAVs. With the improvement on the level of autonomous driving, the V.V&T of CAVs is facing new challenges according for complex traffic environments and driving task diversity [1], [2]. For example, U-turns, Parallel-driving and some illegal behaviors are frequently happened in urban area. One of the main issues is ''How to overcome the dilemma of testing the entire complexity of real-world traffic'' [3].

Scenario-based testing is also known as Use-case testing. Ulbrich *et al.* [4] elaborated the basic terms of scenario, situation and scenario. It is already a well-known and established test approach to the automotive industry for V.V&T [5]. The DARPA Grand Challenge is a representative. The third DARPA Grand Challenge is also named as the DARPA Urban Challenge [6], which focuses on the autonomous driving ability of vehicles for urban environment. This method can simulate various traffic scenarios and evaluate the comprehensive performance of CAV systems. As a result of simulating real traffic, testing costs has dropped significantly. So, it is suitable for the V.V&T of CAVs under complex traffic environments. Due to the complex road network structure, diverse traffic participants and uncertain behaviors, the elements that need to be expressed in urban test scenarios are more abundant. The construction of test scenarios of urban environment is a challenge.

The real traffic is the fundamental source of all test scenarios. Parsing and reconstruction is the main way to generate scenarios. Geiger *et al.* [7], [8] used a traditional car with multiple sensors to collect real traffic scenarios and post-processed the data to build a KITTI dataset. On the basis of the above methods, Zofka *et al.* [9] and Lages *et al.* [10] performed parameterized description and reconstruction of the scenario to achieve automatic scenario generation. Thieman *et al.* [11] and Krajewski *et al.* [12] used drones to record real traffic videos and post-processed

The associate editor coordinating the review of this manuscript and approving it for publication was Chaitanya U. Kshirsagar.

their data to generate NGSIM and HighD datasets. The huge economic cost is the main problem with this method. Akamatsu *et al.* [13] estimated that the total cost of collecting N-FOT (Naturalistic Field Operational Tests) data with equipped vehicles will not be less than \$10,000,000.

In order to overcome the shortcomings of high cost, researchers have tried to generate test scenarios by simulating the behavior of traffic participants. Sippl *et al.* [14] proposed a method of extracting black box test cases from virtual simulation data. In their work, test cases were generated based on the simulation of traffic behavior at an intersection. The selection of criticality metrics based on Traffic Conflict Technique (TCT) was also discussed. Chao *et al.* [15] designed an urban mixed traffic environment simulator to simulate the decision-making processes through the "force" between traffic participants, thereby realizing the CAVs virtual test of mixed conditions. The above work has made useful attempts. There are some problems that need to be further solved. For example, only small-scale traffic environment can be simulated and the expressions of traffic behavior are rough.

Microscopic Traffic Simulation (MTS) is a special tool for obtaining and explaining the complexity of the traffic system [16]. In MTS, a single vehicle or pedestrian is used as the basic unit to simulate the microscopic motion and interaction of different traffic participants. Bloomberg and Dale [17] used CORSIM and VISSIM to implement control scheme simulations for different types of roads and different types of vehicles (cars, heavy vehicles, buses, etc.), and analyzed the performance of the control strategy. Cameron *et al.* [18] developed a parallel microscopic road traffic simulator that built a network of 1 million nodes, 4 million sections, and 32,000 areas. The simulator was used to predict the traffic flow by simulating the driving of every single vehicle and provide a scalable and dynamic 3D visualization interface. In the early stage of system development, the scenario only needs static information (lane types and parameters, non-moving targets, etc.) and dynamic information (traffic participants, motion, attributes, etc.) [19], which can be easily obtained from MTS.

Criticality metrics is indispensable scenario description information. At present, the derived technical test criteria are in general function specific. For example, only TTC (Time-To-Collision) [20] is used in the AEB functional test scenarios [21], [22]. TTC, PET (Post-Encroachment-Time), DST (Deceleration-To-Safety-Time), and their derivative indicators such as Time-Gap [23], PSD (Proportion of Stopping Distance) [24] are commonly used. These metrics are directly related to a certain function and cannot comprehensively evaluate the environment in which the CAV system is located. High-level autonomous driving system involves multiple sub-functions. It may be too simple to aggregate all sub-function metrics into a set. Therefore, it is necessary to design a general indicator to assess the risk of the scene.

In this paper, we propose a novel method for extracting test scenarios, which can reproduce complex and diverse traffic behaviors in urban. The core of the proposed method is MTS. Based on traffic environment information and behavior characteristics, it reproduces complex scenarios not only U-turn and Parallel-driving (two vehicles driving in parallel on a single lane), but also collisions. Further, this paper designs a quantitative indicator to solve the problem that the scenario critical metrics depends on the system sub-function. The likelihood of risk events is modeled as a function of the TTC in Ego (autonomous vehicle itself) perspective, and the collision energy is used to express the risk events loss. Then the Scenario Risk Index (SRI) is obtained according to the risk assessment formula. In the process of method implementation, parallel computing technology is used to solve the big data processing problem.

The Chapter II will systematically introduce the method of generating scenarios and the design concept of the scenario risk indicator. The Chapter III will detailly introduce the implementation process, i.e. the processing methods or calculation processes for each key step. Chapter IV will show the scale and diversity of the dataset. The last chapter summarizes and prospects the above work.

II. THE SCENARIO GENERATION METHOD

As shown in Fig.1, the core of this paper is to reproduce various traffic behaviors in urban through MTS and then we extract the scenarios of Ego perspective. The road network, OD (Origin–Destination) matrix [25], Car-Following and Lane-Changing model parameters of the target area are the basis of MTS. We use public data such as OSM, competition data and previous research results to obtain the above information. The open source software SUMO (Simulation of Urban Transportation) [26] is used as the core tool for performing MTS. The output of SUMO is the data onto God perspective, which needs to be converted into Ego perspective, i.e. generate Ego scenarios. In the conversion process, the boundary of the spatiotemporal range and quantitative evaluation indicators are deliberated and calculated. Matlab is used to design and implement MTS data processing algorithms.

A. DESCRIPTIONS OF URBAN TRAFFIC ENVIRONMENT

Road network, OD matrix and driving behavior parameters are necessary information for MTS. Information such as road structure (road type, number of lanes), road functions (lane direction, speed limits), traffic signs and traffic lights are road network information and can be obtained from map data. OSM is a free, open source and editable map, which can not only provide the required information, but also save the cost of information collection. There are certain difficulties in obtaining OD matrix, Car-Following and Lane-Changing model parameters, which generally need to be obtained from public data such as traffic surveys and early researches.

B. MICROSCOPIC TRAFFIC SIMULATION (MTS)

After obtaining the descriptions, it is needed to select an appropriate microscopic traffic simulator to reproduce the real scenarios and output related data. SUMO is an open

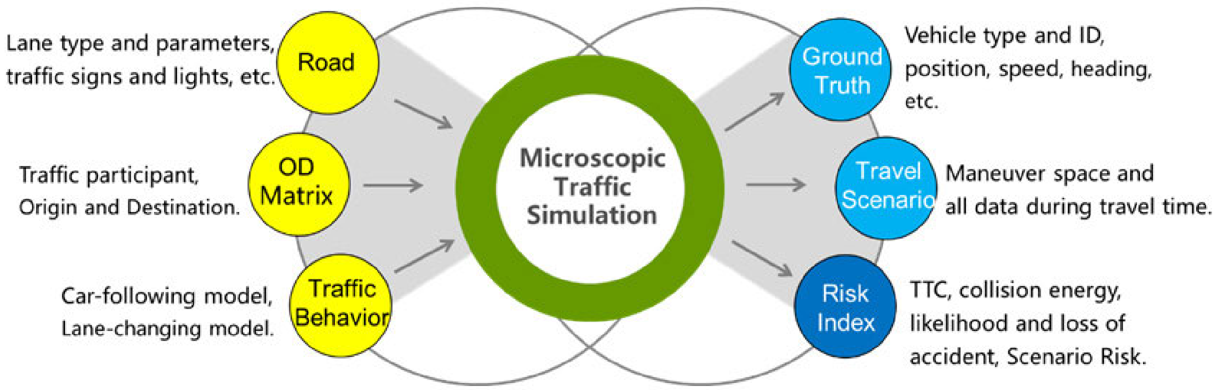


FIGURE 1. **An illustration of the proposed method for generating scenarios.**

source, highly portable, microscopic and continuous traffic simulation package designed to handle large road networks. It is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Center.

SUMO allows a variety of road network input formats, such as OSM, OpenDrive, MATsim, etc. Alternatively, it is allowed to create electronic maps and edit road networks using NETEDIT, which comes with SUMO. The sublane function of SUMO simulates the pushy behavior under dense traffic by constructing a virtual lane. According to Semrau *et al.* [27], this function is developed based on the driving behavior parameters of Chinese drivers, which can effectively simulate the traffic behavior of Chinese traffic. SUMO allows to generate a large number of different measures. Raw vehicle positions dump, trip information, vehicle routes information, and simulation state statistics are optional output. The raw vehicle positions dump is the first output capability that is implemented in SUMO. It contains detailed information for each edge, each vehicle and each simulation step. Due to the high level of detail it is very flexible but often requires additional programming to parse and filter the information that is actually desired. Lopez *et al.* [28] introduced the latest progress of SUMO in transportation solution simulation.

C. EGO SCENARIOS EXTRACTION

The output of MTS is God-Perspective, which contains the status information about traffic participants at each moment. Each vehicle in the simulation is regarded as Ego. The data in the specific time and space range of Ego is extracted to constitute the Ego scenarios.

In terms of scenario coverage space, considering that CAV only needs to perceive, analyze, and interpret the objects within a certain range, we set the spherical space with the center of Ego vehicles and the radius of R_{ego} as the Maneuver Space [29] .Only static and dynamic object within the Maneuver Space need to be extracted. Scenarios which may look harmless or initially pose no danger might also be interesting for interpretation, due to environment reasons and missing or vague traffic guidance. Therefore, all the data within the travel time needs to be extracted to form an Ego scenario.

In our Ego travel scenarios, complete static and dynamic object information are ground truth. Object annotation is easy if needed.

D. SCENARIO RISK ASSESSMENT INDICATORS

Quantitative scenario criticality metrics is the main basis for scenario classification and evaluation. The metrics should objectively reflect the degree of danger of the scenario, not the triggering conditions of a certain function. If collision is regarded as a risk event, the size of the scenario risk can reflect the degree of scenario danger. According to the basic principles of Risk Assessment (RA), the magnitude of risk is equal to the probability of a risk event times the degree of loss, as shown in (1).

$$Risk = Probability \times Loss \tag{1}$$

Therefore, we use two basic terms of TCT to express the likelihood of risk and the number of loss. Then the magnitude of risk is calculated by multiplication.

TTC is a commonly used criticality metric [30]-[32]. According to Chen [33] and Cheng and Hsiung [34], the relationship between TTC and the probability of collision can be calculated quantitatively by using (2)

$$p(TTC = x) = \begin{cases} 1 & x < a \\ 1 - 2\left(\dfrac{x-a}{b-a}\right)^2 & a \le x < \dfrac{a+b}{2} \\ 2\left(\dfrac{x-b}{b-a}\right)^2 & \dfrac{a+b}{2} \le x < b \\ 0 & x \ge b \end{cases} \tag{2}$$

where TTC is the possibility of a collision of $TTC = x$ and the parameters a and b are set as 0.5 and 2.5. A value of 100% of collision probability index means the collision has happened and a value of 0% of collision probability index means the driving is in safety scenario.

Collision Energy (CE) is the destructive kinetic energy of the conflict vehicle for the moment when the risk-avoidance

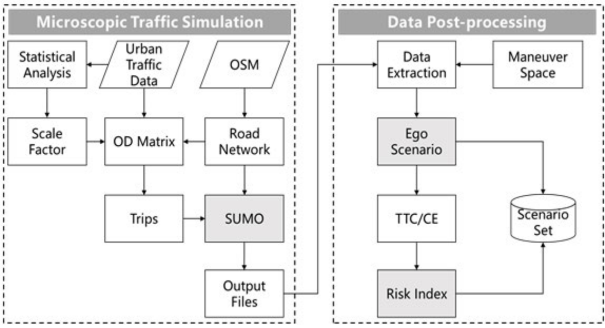


FIGURE 2. The flow diagram of our method.

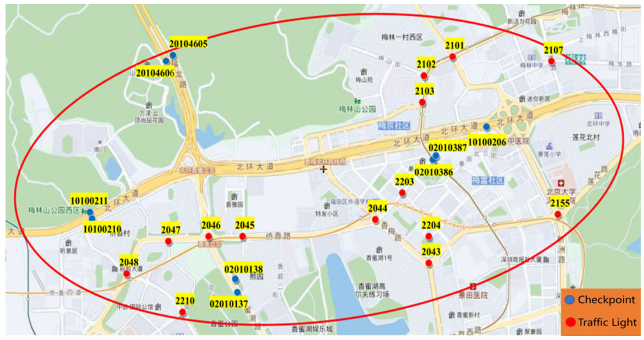


FIGURE 3. Area covered by floating cars, traffic lights, and checkpoints.

action takes effect. It is an important indicator to measure the potential loss of the conflict. Assuming that all kinetic energy before the collision is lost, CE is the amount of loss. So, the loss is obtained by using (3).

$$L(m, v) = CE = \frac{1}{2} m \cdot v^2 (t) \tag{3}$$

where m is the total mass of Ego vehicle and $v(t)$ is the speed of Ego Vehicle at time t.

The magnitude of risk Rm is equal to the risk probability p multiplied by the risk loss L, i.e.

$$Rm = p \cdot L = p(TTC = x) \cdot L(m, v) \tag{4}$$

We called Rm as Scenario Risk Index (SRI).

So far, the static information, dynamic information, and risk assessment indicators are obtained through the above steps. This information can completely describe a scenario and generate a catalog.

III. THE KEY STEPS IN SCENARIO GENERATION

The practice of the method mentioned in the previous chapter can be divided into two parts. One is Microscopic Traffic Simulation (MTS), and the other is Data Post-processing. MTS includes real traffic data processing, implementation, and simulation effect verification. The workflow is long and complicated. Since the data obtained by MTS is not a scenario that we can directly use, Data Post-processing is required. The risk indicators are calculated millions of times, which consumes a lot of computing resources. We have developed a set of parallel computing algorithm to reduce the time-consuming. The following will focus on these two aspects of work, in detail the process of generating test scenarios. Fig.2 shows the flow chart of this process and highlights the key steps.

A. MICROSCOPIC TRAFFIC SIMULATION (MTS)

1) THE SOURCE AND PREPROCESSING OF REAL TRAFFIC DATA

The real traffic data used in this paper comes from the part D of the 2018 Shenzhen Cup Mathematical Contest. The data includes the floating car data in part of Shenzhen (the area within the red line in Fig.3), 9 traffic checkpoints data (at the blue dot in Fig.3) data and the timing strategy of some traffic lights (at the red dot in Fig.3). We selected the floating car data as follows: discard the data of buses, trucks and other vehicles except for taxis; discard the data whose sampling interval is greater than 30s; according to the speed limit, discard the data whose speed is greater than 70km/h.

2) ROAD NETWORK CORRECTION

The open source OSM format map provides a basic road network. Since there are few participants in the OSM project in China, there are many deviations between OMS and actual road network. With reference to Baidu Map, the mainstream commercial software in China, we used the JOSM (Java OpenStreetMap Editor) to modify the number of lanes on some roads.

3) OD MATRIX ESTIMATION

By comparing the trajectories of floating cars with the road network, we can find the routes that are closest to the trajectories on the map. The route matching was implemented using GraphHopper/Map-Matching, which is an open source map matching program based on Hidden Markov Method on GitHub[1]. We counted the road transfer number in all routes and then obtained the road-level OD matrix. The OD matrix of floating cars is only a part of the OD matrix of all social vehicles and needs to be expanded. According to the traffic volume obtained by Traffic checkpoint, the proportion of floating cars in all vehicles represented by p was calculated. The reciprocal q=1/p was used as the amplification factor. The OD matrix of all vehicles was equal to the floating car OD matrix times the amplification factor q.

4) IMPLEMENTATION OF MTS BASED ON SUMO

The steps to run SUMO are as follows:

Step 1 Importing the original road network: We used the NETCONVERT tool to import OSM and generate road networks that can be read by other tools. We set the option − *output*.original-names. So, the IDs of nodes and edges in SUMO were the same as in OSM.

Step 2 Processing road network files: NETEDIT is a visual network editor. We used it to modify all aspects of existing networks. Some abnormal nodes and/or edges such as

[1] https://github.com/graphhopper/map-matching

TABLE 1. Microscopic traffic simulation effect.

Check point	Real Traffic Volume (Vehicles/Day)	Simulation Volume (Vehicles/Day)	Relative Error
1	67239	62492	7.06%
2	39952	36769	7.97%
3	89131	79843	10.42%
4	97875	92257	5.74%
5	62717	59461	5.19%
6	70001	64598	7.72%
7	6162	5723	7.12%
8	8018	7433	7.30%
9	88103	84793	3.76%

overpasses were deleted. Timing strategies of traffic lights were configured. The traffic assignment zones were set as 5.

Step 3 Configuring the OD matrix: As a pre-step, all vehicle properties were set to car represented by a rectangle with size parameters length×width×height=4.7m×1.8m×1.4m. These parameters are consistent with the *Actor Classification: Car* in Matlab. Then, the OD matrix of traffic zones was accumulated through the OD matrix of roads, and the *mtx* file used by SUMO was written.

Step 4 Generating trips: The number of departure vehicles was allocated by the proportion of vehicles in different time periods. We used *od2trips* command in SUMO to generate trips.

Step 5 Configuring *sumocfg* file: The *Sublane* parameter was set to 0.01m. The program was set to simulate 24-hour traffic with a time interval 0.1s. The output files were *raw vehicle positions dump* file and *fcd* file. The default car following model and lane changing model were used during the simulation.

The simulation results are verified by comparing the traffic volume between simulation and real at the 9 checkpoints. As shown in Table I, the relative errors are 5.19% to 10.42%, indicating that the simulation can approximately reflect the real traffic.

B. DATA POST-PROCESSING

1) EGO SCENARIOS EXTRACTION

As shown in Table 2, the *raw vehicle positions dump* file and *fcd* file contain several fields. These data are stored in time series. We call it the God perspective data. The following will introduce how to extract the Ego perspective scenarios from the God perspective data.

Before extracting the Ego scenario, the time and space ranges need to be defined. In order to verify the system reliability under long-term operating conditions, all data in the Ego vehicle's existence times were extracted. The maximum detection distance of the CAVs perception system is about 200 meters, and the information transmission distance between the network connection equipment varies greatly. In order to cover the detection range and reserve a certain distance to simulate the network connection function, the Maneuver Space radius *Rego* is set as 260 meters. The effect of object outside the radius *Rego* is ignored.

TABLE 2. Main fields included in SUMO output data.

Name	Dimension	Description	Type
time	second	The time step described by the valu es within this time step-element.	Dyna mic
vehID		The id of the vehicle.	Static
edgeID		The id of the edge.	Static
laneID		The id of the lane.	Static
vehType		The name of the vehicle type.	Static
pos	m	The position of the vehicle at the la ne within the described time step (di stance of the front bumper from the start of the lane).	Dyna mic
x	m	The absolute X coordinate of the ve hicle (center of front bumper).	Dyna mic
y	m	The absolute Y coordinate of the ve hicle (center of front bumper).	Dyna mic
angle	degree	The angle of the vehicle in navigati onal standard (0-360 degrees, going clockwise with 0 at the 12'o clock p osition).	Dyna mic
speed	km/h	The speed of the vehicle within the described time step.	Dyna mic
posLat	m	Offset from the center of the lane (o nly when using the sublane model).	Dyna mic
speedLat	m/s	Lateral movement speed (only whe n using the sublane model).	Dyna mic

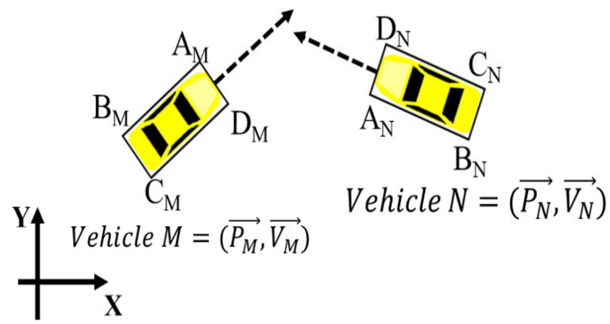


FIGURE 4. Simplified vehicle models and notes used in this paper.

The vehID is used as the Ego scenario ID. The data (see Table 2 for details) in the Maneuver Space during the vehicle's existence were extracted into an Ego scenario. Then, according to the characteristics of SUMO's edgeID and OSM wayID, road network data were extracted.

In the extraction, there are a few details should be noted. Firstly, using the output file to write all data at once is beneficial to use multiple computers to process the data, but the data will be saved in an .XML file over 40GB. The .XML file can be processed by Matlab only after it is divided and converted into a small (MB level) CSV file. Secondly, SUMO and Matlab have different definitions of vehicle position and vehicle-mounted coordinate system. We convert the above relationship while extracting. Finally, Ego scenarios are split into small .mat files about 1GB, and named *EgoSce_Order.mat* in the order of file generation. In order to reduce time, we distribute the *EgoSce_Order.mat* files to multiple computers for risk index calculation.

2) THE CALCULATION OF SRI

The TTC is a prerequisite for calculating *Rm*. A rectangle of length l and width d is used to represent a vehicle (as shown in Fig.4). According to the definition of traffic conflict, the trajectories of all objects in the Ego scenario under the condition of "maintain the speed and direction" are predicted. While predicting the trajectory, we judged whether a collision has happened. If a collision has happened at a certain time tc, then $TTC = tc$.

Before the calculation, the objects that are unlikely to collide were excluded. Objects in the Maneuver Space may collide under 2 conditions. One is that the road where they are located is the same. The other is that the roads where they are located intersect at a certain place. According to the conditions, the objects which may conflict with the Ego were selected into a set $\{Actor1, Actor2, \ldots, ActorK\}$.

Assuming that the current time is $t = t_0$, equation (5) is used to calculate the trajectory at time $t = t_0 + t^*$.

$$\begin{cases} P_{ego}(t^*) = P_{ego}(t_0) + S_{ego}(t_0) \cdot t^* \\ P^k_{act}(t^*) = P^k_{act}(t_0) + S^k_{act}(t_0) \cdot t^* \end{cases} \tag{5}$$

where $Pego(t)$ is the position of the Ego vehicle at time t, i.e. the point on the ground that is below the center of its rear axle. $Sego(t)$ is the speed of the Ego vehicle at time t. $P^k act(t)$ is the position of the kth actor at time t. $S^k act(t)$ is the speed of the kth actor at time t. Time t^* will increase by an interval of 0.1s.

Based on the vehicle size and heading, the vehicle occupied range represented by four vertices of a rectangle can be calculated. In the vehicle coordinate system with the position point as the origin and the heading angle as the positive direction, the vertex coordinates are fixed. We used the coordinate system rotation and translation formulas to calculate the coordinates of the vertices in the global coordinate system. The above method is summarized in Algorithm1.

Algorithm 1 *Calculating the Vertices in Global Coordinate*

Inpute vehicle position(x, y), heading(θ), and vehicle type(*VehType*),
Get vertices coordinates in the vehicle coordinate system according to *VehType*
switch *VehType*
  case 'car', $VertsInVehCoord = (\vec{x}_r \vec{y}_r)$;
  ...
  end
Calculate coordinate rotation matrix *rotz* $\leftarrow$ [cos(θ) -*sin*(θ)*; sin*(θ) cos(θ)]
For each vertex *i* **do**
$VertexInVeh \leftarrow VertsInVehCoordf$ (i,:);
$VertsInGlobalCoord \leftarrow rotz \cdot VertexInVeh + (x, y)$;
end

Then, we judge whether there is a collision or not. If a vertex of a rectangle is inside another rectangle, a collision happens.

In this process, the directionality of vector cross product is used for quick judgment. There is an example of judging whether the vertex A_M in Fig.4 is in the range of Vehicle N. If both (6) and (7) are satisfied, the vertex A_M is located in Vehicle N.

$$sign\left(\overrightarrow{A_N B_N} \times \overrightarrow{A_N A_M}\right) = sign\left(\overrightarrow{C_N D_N} \times \overrightarrow{C_N A_M}\right) \tag{6}$$

$$sign\left(\overrightarrow{D_N A_N} \times \overrightarrow{D_N A_M}\right) = sign\left(\overrightarrow{B_N C_N} \times \overrightarrow{B_N A_M}\right). \tag{7}$$

where *sign* is a signum function.

The pseudocode of judging whether a point is inside a rectangle is shown in Algorithm 2.

Algorithm 2 *Judging Point_P Is inside Rectangle_ABCD*

Inpute *Point _P* = (x_p, y_p) and coordinates of rectangle vertices$\{(X_A, Y_A), (X_B, Y_B), (X_C, Y_C), (X_D, Y_D)\}$
Set initial *flag 1* = 0 and *flag 2* =*O*
Calculate the 1st set of vectors
$AB \leftarrow (x_B, y_B) - (x_A, y_A)$, $AP \leftarrow (x_p, y_p) - (X_A, Y_A)$. $CD \leftarrow (x_D, y_D) - (x_c, y_c)$, $CP \leftarrow (x_p, y_p) - (x_C, y_C)$
Calculate outer product
0P1←*AB*× *AP,0P2*←CD × *CP*
if sign(OP 1) == sign(0P2)
  flag 1 = 1;
end
Calculate the 2nd set of vectors {D/4. *DP, BC, BP}* and flag2 with the same steps as the 1st set of vectors
$flag \leftarrow flag1 \cdot flag2$
Return *flag*

Furthermore, we used (2) to calculate the collision probability $p(TTC = x)$. It should be pointed out that since the radius *Rego* was defined, $TTC \in \left[0, Rego/Sego\right]$. In order to avoid the situation of $TTC \to \infty$ while $Sego \to \infty$, the speed lower bound was set as $Slower = 0.1$m/s. The loss $L\ (m, v)$ was calculated by using (3) synchronized with TTC.

Finally, we used (4) to calculate the SRI *Rm* to complete all calculation tasks. So far, the static information, dynamic information, and risk indicators can be aggregated to a scenario set.

3) THE ACCELERATION OF TTC ALGORITHM

Our MTS simulated the movement of multi vehicle over a long period. As shown in Table 3, the vehicle travel time is 8288.14 Veh·h with a time interval of 0.1s. So, there are 2.98×10^8 TTCs to be calculated. And under the calculation of TTC, we must predict the trajectory within a certain period in the future, calculate the rectangle vertex coordinates, and judge whether a collision happens. Tens of millions of TTC calculations will consume a lot of time. If each TTC calculation takes 0.1s, it will take 345.33 days. The time cost is unacceptable.

In order to reduce the calculation time, we used multiple computers to process the split data packets at the same time.

And, we tapped the performance of multi-core processors by using the *parfor_loop* in the Matlab parallel computing toolbox instead of *for_loop*. The pseudo code of the accelerated TTC algorithm is shown in Algorithm 3.

There are three computers participated in the calculation of TTC. A high-performance computer completed most of

Algorithm 3 *Acceleration Algorithm for TTC*

For each ego vehicle **do**
Input the time series $\{time_{ego}\}$, Ego' status $\{x_{ego}, y_{ego}, S_{ego}, \theta_{ego}, WayID_{ego}\}$
and Actors' status $\{x_{act}, y_{act}, S_{act}, \theta_{act}, WayID_{act}\}$
Set initial $TTC = 2600$
Parfor each $time_{ego} = t$ **do**
For each actor k do
If $WayID_{ego} \cap WayID_{act}^{k} = \emptyset$ **then**
delete actor k;
end
end
If $S_{ego} > 0.1$ & $\{Actors\} \neq \emptyset$ then
For each prediction time step $\tilde{t}$ **do**
(x_{ego}, y_{ego})
(x_{act1}, y_{act1})
$\vdots \leftarrow$ Update the positions with (5)
(x_{actM}, y_{actM})
Calculate Ego vertices with Algorithm 1:
$Verts_{ego}\{A, B, C, D\} \leftarrow Algorithm_1\ (x_{ego}, y_{ego}, Q_{ego}, VehType)$
For each actor **do**
calculate act_m vertices with Algorithm 1;
$Verts_{actm}\{E, F, G, H\}$
$\leftarrow Algorithm\_1(x_{actm}, y_{actm}, \theta_{actm}, VehType)$
Determine whether the vertex of one vehicle is within the rectangle enclosed
by the other vehicle's vertices with Algorithm 2
$\{flag1, flag2, flag3, flag4\}$
$\leftarrow Algorithm\_2(\{A, B, C, D\}, Verts_{actm})$
$\{flag5, flag6, flag7, flag8\}$
$\leftarrow Algorithm\_2(\{E, F, G, H\}, Verts_{ego})$
If $\sum_{i=1}^{8} flag\ i > 0$ **do**
$TTC \leftarrow \tilde{t}$
break the loop for actor;
End all loops and judgments

the work. Another desktop computer and a laptop computer played as auxiliaries. The high-performance computer is equipped with an Intel i9-7920X CPU, with 8 cores and 16 threads. Setting the computer environment variable *OMP_NUM_THREADS* to the maximum number of CPU threads can theoretically increase the calculation speed by 16 times. Due to parallel overhead, including the time required to transfer data from the client to the workers and back, the speed-up is smaller than the ideal speed-up. The *parfor_loop* is about 15 times faster than the corresponding *for_loop* calculation.

IV. THE EXHIBITION OF OUR SCENARIO SET

Through the MTS in urban, we obtained a huge test scenario set. The main attributes of our scenario set are shown in Table 3. The scenario set contains the data of 189,752 vehicles driving on 198 km roads. If a well-equipped vehicle is used for collecting the same scale data, 254,480 km or 8,288 h are needed.

TABLE 3. Main fields included in SUMO output data.

Attribute	Parameter	Attribute	Parameter
Road mileages	198 (km)	Intersections	197
Vehicles	189,752	Traffic lights	76
Driving mileages	254,478 (veh·km)	Lane changing	153,599
Travel time	8,288.14 (veh·h)	Emergency braking	6,439

There are 153,599 lane-changing and 6,439 emergency braking behaviors in the set, which can be used to construct test case for ADAS functions. In addition, 103 collision scenarios are contained. On average, a collision happens after a vehicle driving 2470 km as a small probability event. These collision scenarios provide the data before and during the collision.

There are 153,599 lane-changing and 6,439 emergency braking behaviors in the set, which can be used to construct test case for ADAS functions. In addition, 103 collision scenarios are contained. On average, a collision happens after a vehicle driving 2470 km as a small probability event. These collision scenarios provide the data before and during the collision.

A. URBAN TRAFFIC SCENARIO

Compared with the expressway or highway environment, the diversified characteristics of traffic behaviors are more obvious in the urban environment. So, the scenarios are more challenging.

1) U-TURN SCENARIO

U-turn is a common traffic behavior on urban roads. This behavior will occupy multiple lanes and affect vehicles traveling in both directions. As shown in Fig.5, the side of the U-turning vehicle faced another vehicle. This scenario poses a great challenge to the environmental perception capability of the CAVs system.

2) PARALLEL DRIVING SCENARIO

In urban, there may be a scenario in which three vehicles are driving in two lanes on a congested road. We named the above scenario as a Parallel Driving Scenario, as shown in Fig.6 (a). The sublane function in SUMO realizes the simulation of the Parallel Driving Scenario.

The ability to distinguish two adjacent targets and track them separately is required for CAV system. The Parallel Driving Scenario puts forward higher requirements on the environment perception, path planning and decision.

B. COLLISION SCENARIO

During the test of the automated driving system, collision data play an important role in the evaluation. However, collecting the data before and during a collision is by chance. First of all, every collision represents a casualty or loss of property, which we do not want to meet. Secondly, a collision is a

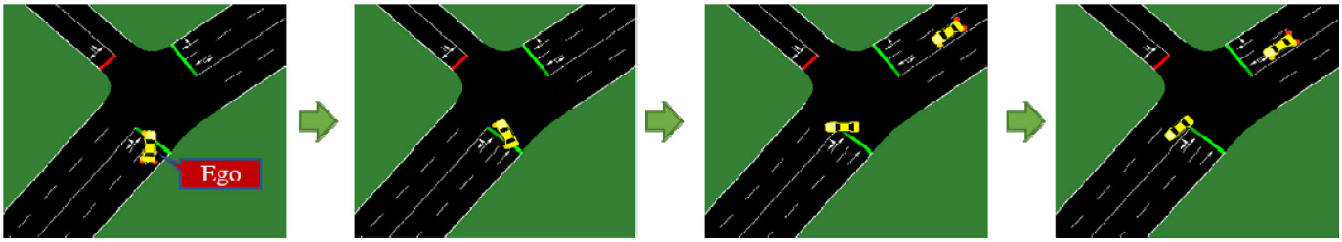


FIGURE 5. An example of U-turn Scenario.

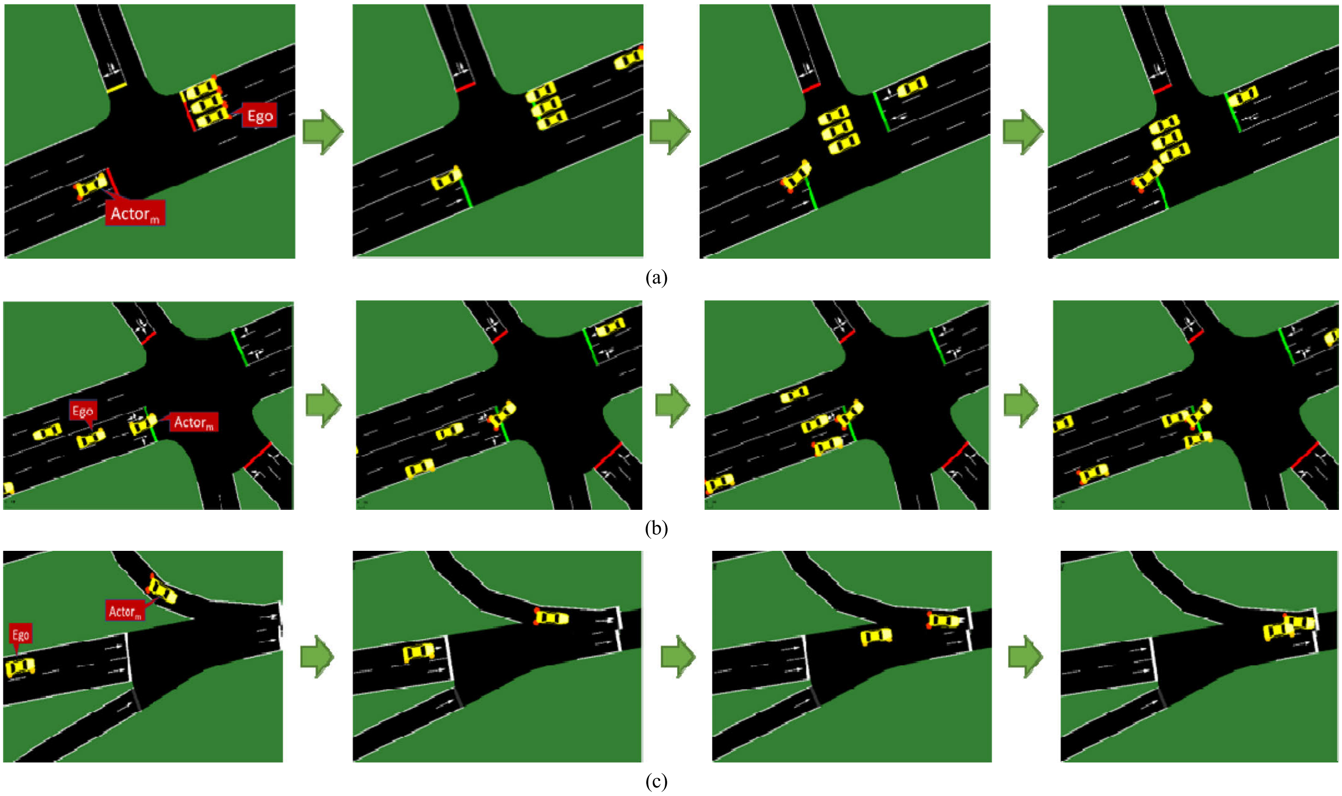


FIGURE 6. Three typical collision scenarios, (a) a Parallel Driving Scenario with a front collision, (b) a crossing collision scenario, (c) a rear-end collision scenario.

small probability event. According to Zhao *et al.* [35], in the US, an average of 530,000 miles traveled before a collision was reported to the police. Through our method, the collision scenario can be reproduced with diversity. A front collision scenario, a crossing collision scenario, and a rear-end collision scenario are shown in Fig.6 (a) to (c).

1) FRONT COLLISION SCENARIO

A frontal collision is shown in Fig.6 (a) at a T-intersection. When the simulation time was 1349.9s, Ego and the other two vehicles stopped at a two-lane road intersection. When the simulation time was 1350s, all vehicles started and drove straightly. At the same time, Actorm turned left and took a portion of the space in the lane where Ego was. Then a collision happened. According to the TTC and SRI which are shown in Fig.7, TTC dropped rapidly after 1351s from 10 to 0 with the SRI rose from 0 to 3.2 KJ. Since the collision energy is at a low level, it is a slight accident.

2) CROSSING COLLISION SCENARIO

Fig.6 (b) shows a crossing collision scenario at an irregular intersection. We can see that a series of operation such as lane changing and parking are taken by Actor$_m$ in the right lane. The operations of Actor$_m$ are illegal and cause this collision. As shown in Fig.8, the decline of TTC and the rise of SRI indicate a dramatical increase in the risk of the scenario. Since the collision energy is at a low level of 1.5kJ, it is a slight accident too.

3) REAR-END COLLISION SCENARIO

Fig.6 (c) shows a rear-end collision scenario at another irregular intersection. The Ego drove along a straight road. Actorm changed lanes after turning out of a curve and appeared in front of Ego. During the lane changing, Actorm was decelerating and Ego vehicle was accelerating. Then a rear-end collision happened. TTC and SRI are shown in Fig.9. As

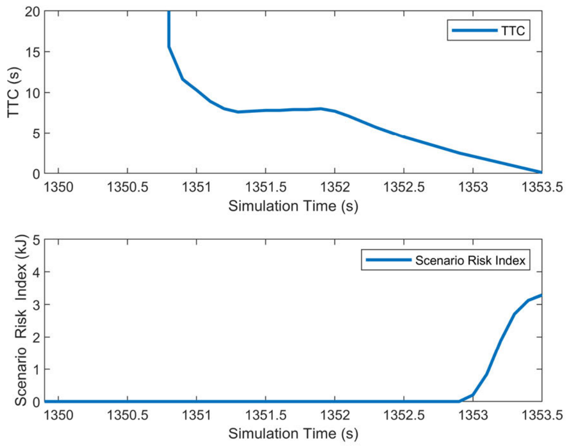


FIGURE 7. The TTC and SRI during the front collision scenario.

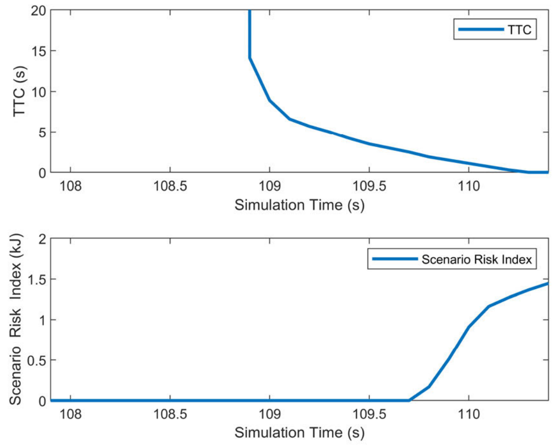


FIGURE 8. The TTC and SRI during the crossing collision scenario.

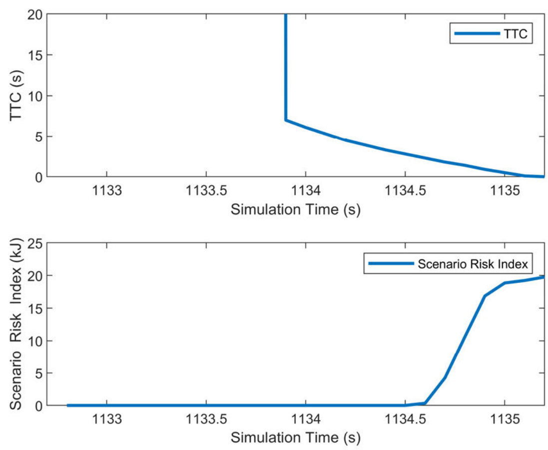


FIGURE 9. The TTC and SRI during the rear-end collision scenario.

can be seen, the SRI rose rapidly before the collision. The maximum risk index of the scenario is close to 20kJ, and this collision will cause a certain loss.

V. CONCLUSION AND FUTURE WORK

This paper proposes a completed method for extracting scenarios from MTS data. Our method does not require expensive equipment and labor, thereby reducing the cost of scenario collection. According to the simulation results in the part urban area of Shenzhen, a set of 189,752 scenarios was obtained at one time. The set contains typical urban traffic scenes such as U-turn, Parallel Driving and different types of collision. At the same time, this paper breaks the convention of designing criticality metrics for scenarios based on specific functions. Risk assessment method is used to design SRI which is not related to any CAV function. An optimized calculation process through parallel computing technology was designed and realized a 15-times acceleration.

In future work, it is first necessary to increase the types of traffic participants and construct a mixed scenario of urban environment that includes motor vehicles, non-motor vehicles and pedestrians. Secondly, how to evaluate traffic scenarios under mixed traffic is an open question. The existing evaluation indicators cannot yet reflect the scenario complexity.

REFERENCES

[1] M. Ardelt, C. Coester, and N. Kaempchen, "Highly automated driving on freeways in real traffic using a probabilistic framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1576–1585, Dec. 2012, doi: 10.1109/TITS.2012.2196273.

[2] F. Saust, J. M. Wille, B. Lichte, and M. Maurer, "Autonomous vehicle guidance on Braunschweig's inner ring road within the stadtpilot project," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Baden-Baden, Germany, Jun. 2011, pp. 169–174.

[3] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zollner, "Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Las Palmas, Spain, Sep. 2015, pp. 1455–1462.

[4] S. Ulbrich, T. Nothdurft, M. Maurer, and P. Hecker, "Graph-based context representation, environment modeling and information aggregation for automated driving," in *Proc. IEEE Intell. Vehicles Symp.*, Dearborn, MI, USA, Jun. 2014, pp. 541–547.

[5] P. Junietz, W. Wachenfeld, K. Klonecki, and H. Winner, "Evaluation of different approaches to address safety validation of automated driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, Nov. 2018, pp. 491–496.

[6] A. S. Huang, M. Antone, E. Olson, L. Fletcher, D. Moore, S. Teller, and J. Leonard, "A high-rate, heterogeneous data set from the DARPA urban challenge," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1595–1601, Nov. 2010, doi: 10.1177/0278364910384295.

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3354–3361.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res*, vol. 32, no. 11, pp. 1231–1237, Nov. 2013, doi: 10.1177/0278364913491297.

[9] M. R. Zofka, F. Kuhnt, R. Kohlhaas, C. Rist, T. Schamm, and J. M. Zöllner, "Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems," in *Proc. 18th Int. Conf. Inf. Fusion*, Washington, DC, USA, Jul. 2015, pp. 1422–1428.

[10] U. Lages, M. Spencer, and R. Katz, "Automatic scenario generation based on laserscanner reference data and advanced offline processing," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Gold Coast, QLD, Australia, Jun. 2013, pp. 146–148.

[11] C. Thiemann, M. Treiber, and A. Kesting, "Estimating acceleration and lane-changing dynamics based on NGSIM trajectory data," *Transp. Res. Rec.*, vol. 2088, pp. 90–101, Jan. 2008, doi: 10.3141/2088-10.

[12] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, Nov. 2018, pp. 2118–2125.

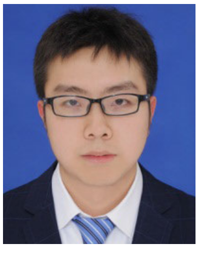
[13] M. Akamatsu, P. Green, and K. Bengler, "Automotive technology and human factors research: Past, present, and future," *Int. J. Veh. Technol.*, vol. 2013, no. 27, pp. 1687–5702, Mar. 2013, doi: 10.1155/2013/526180.

[14] C. Sippl, F. Bock, D. Wittmann, H. Altinger, and R. German, "From simulation data to test cases for fully automated driving and ADAS," in *Proc. 28th IFIP WG 6.1 ICTSS*. Graz, Austria: Technical Univ. Graz, 2016, pp. 191–206.
[15] Q. Chao, X. Jin, H.-W. Huang, S. Foong, L.-F. Yu, and S.-K. Yeung, "Force-based heterogeneous traffic simulation for autonomous vehicle testing," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 8298–8304.
[16] P. M. Ejercito, K. G. E. Nebrija, R. P. Feria, and L. L. Lara-Figueroa, "Traffic simulation software review," in *Proc. 8th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Larnaca, Cyprus, Aug. 2017, pp. 1–4.
[17] L. Bloomberg and J. Dale, "Comparison of VISSIM and CORSIM traffic simulation models on a congested network," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1727, no. 1, pp. 52–60, Jan. 2000, doi: 10.3141/1727-07.
[18] G. D. B. Cameron and G. I. D. Duncan, "PARAMICS-Parallel microscopic simulation of road traffic," *J. Supercomput.*, vol. 10, no. 1, pp. 25–53, 1996, doi: 10.1007/BF00128098.
[19] E. de Gelder and J.-P. Paardekooper, "Assessment of automated driving systems using real-life scenarios," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Los Angeles, CA, USA, Jun. 2017, pp. 589–594.
[20] A. Sadek, B. Abdullah, and W. R. Anis, "Safety improvement in vehiclar communication systems," in *Proc. 12th Int. Conf. Comput. Eng. Syst. (ICCES)*, Cairo, Egypt, Dec. 2017, pp. 325–328.
[21] M. Zhao, H. Wang, J. Chen, X. Xu, and Y. He, "Method to optimize key parameters and effectiveness evaluation of the AEB system based on rear-end collision accidents," *SAE Int. J. Passenger Cars-Electron. Electr. Syst.*, vol. 10, no. 2, pp. 310–317, Mar. 2017, doi: 10.4271/2017-01-0112.
[22] M. Lin, J. Y. Jaewoo, and K. Byeongwoo, "Proposal and validation of AEB system algorithm for various slope environments," in *Proc. 10th Future Tech.*, Hanoi, Vietnam, 2015, pp. 241–246.
[23] Z. Gao, W. Yan, and H. Li, "Design of the time-gap-dependent robust headway control algorithm for ACC vehicles," *Int. J. Veh. Des.*, vol. 70, no. 4, pp. 325–340, 2016, doi: 10.1504/IJVD.2016.076737.
[24] Y. Kuang, X. Qu, S. Wang, Z. Liu, S. Wang, K. Huang, A. Charly, T. V. Mathew, K. N. Porfyri, A. Kotsi, and E. Mitsakis, "A tree-structured crash surrogate measure for freeways," *Accident Anal. Prevention*, vol. 77, pp. 137–148, Apr. 2015, doi: 10.1016/j.aap.2015.02.007.
[25] P. Krishnakumari, H. V. Lint, T. Djukic, and O. Cats, "A data driven method for OD matrix estimation," in *Proc. 23rd ISTTT, EPFL Campus*, Lausanne, Switzerland, 2019, pp. 38–56.
[26] D. Krajzewicz, "Traffic simulation with SUMO–simulation of urban mobility," in *Fundamentals of Traffic Simulation*, vol. 145. New York, NY, USA: Springer, 2011, pp. 269–293, doi: 10.1007/978-1-4419-6142-6_7.
[27] M. Semrau, J. Erdmann, and B. Friedrich, "Multi-level-validation of Chinese traffic in the ChAoS framework," in *Proc. SUMO User Conf.*, Berlin, Germany, 2018, pp. 194–205.
[28] P. A. Lopez, E. Wiessner, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flotterod, R. Hilbrich, L. Lucken, J. Rummel, and P. Wagner, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, Nov. 2018, pp. 2575–2582.
[29] C. Rodemerk, S. Habenicht, A. Weitzel, H. Winner, and T. Schmitt, "Development of a general criticality criterion for the risk estimation of driving situations and its application to a maneuver-based lane change assistance system," in *Proc. IEEE Intell. Vehicles Symp.*, Alcala de Henares, Spain, Jun. 2012, pp. 264–269.
[30] *Intelligent Transport Systems (ITS); V2x Applications; Part 1: Road Hazard Signaling (RHS) Application Requirements Specification*, Standard ETSI Ts 101 539-1, 2013.
[31] *Intelligent Transport Systems (ITS); V2x Applications; Part 3: Longitudinal Collision Risk Warning (LCRW) Application Requirements Specification*, Standard ETSI Ts 101 539-3, 2013.
[32] *Intelligent Transport Systems (ITS); V2x Applications; Part 2: Intersection Collision Risk Warning (ICRW) Application Requirements Specification*, Standard ETSI Ts 101 539-2, 2018.
[33] Y. L. Chen, "Study on a novel forward collision probability index," *Int. J. Vehicle Saf.*, vol. 8, no. 3, pp. 193–204, 2015, doi: 10.1504/IJVS.2015.070762.
[34] K.-P. Chen and P.-A. Hsiung, "Vehicle collision prediction under reduced visibility conditions," *Sensors*, vol. 18, no. 9, p. 3026, Sep. 2018, doi: 10.3390/s18093026.
[35] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 733–744, Mar. 2018, doi: 10.1109/TITS.2017.2701846.

BINGJIAN YUE received the B.Sc. degree in information and computing science from Beihang University, Beijing, China, in 2010, and the M.Sc. degree in vehicle operation engineering from Jilin University, Changchun, China, in 2015, where he is currently pursuing the Ph.D. degree with the Department of Vehicle Operation Engineering, Transportation College. His research interests include vehicle driving cycle generation, test scenarios construct, and environmental perception reliability of intelligent vehicles.

SHUMING SHI received the B.Sc. degree in automotive engineering and the Ph.D. degree in vehicle operation engineering from Jilin University, Changchun, China, in 1987 and 1998, respectively.

From 2001 to 2002, he was a Senior Visiting Scholar with TNO Automotive, Helmond, The Netherlands. Since 2004, he has been a Professor with Jilin University, where he is the Academic Leader of the Vehicle Operation Simulation Group, Transportation College. He has presided over three projects of the National Natural Science Foundation, four provincial and ministerial projects, and more than ten enterprise projects. He is the author of one textbook of the National 11th Five-Year Plan, more than 40 academic articles, and more than ten inventions. His research interests include driving cycle generation, nonlinear vehicle dynamics, vehicle control systems, and state estimation.



SHUO WANG received the B.Sc. degree in vehicle operation engineering from Jilin University, Changchun, China, in 2017, where he is currently pursuing the master's degree with the Department of Vehicle Operation Engineering, Transportation College. His research interests include microscopic traffic simulation and simulation scene modeling.

NAN LIN received the B.Sc. and Ph.D. degrees in vehicle operation engineering from Jilin University, Changchun, China, in 2011 and 2016, respectively. Since 2016, she has been a Postdoctoral Fellow with the State Key Laboratory of Automotive Simulation and Control, Department of Automotive Engineering, Automotive College, Jilin University. She is currently a Lecturer with the Transportation College, Jilin University. Her research interests include the development of driving cycle, vehicle state estimation, and speed profile optimization.