

Research Article

Path Planning for Autonomous Vehicle Based on a Two-Layered Planning Model in Complex Environment

Jiajia Chen ¹, Rui Zhang ¹, Wei Han ¹, Wuhua Jiang ¹, Jinfang Hu ¹,
Xiaoshan Lu ¹, Xingtao Liu ¹, and Pan Zhao ²

¹School of Automotive and Transportation Engineering, Hefei University of Technology, Hefei 230009, China

²Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China

Correspondence should be addressed to Pan Zhao; pzhaohfcas.ac.cn

Received 13 October 2020; Revised 3 November 2020; Accepted 13 November 2020; Published 27 November 2020

Academic Editor: Kun Wang

Copyright © 2020 Jiajia Chen 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.

The autonomous vehicle consists of perception, decision-making, and control system. The study of path planning method has always been a core and difficult problem, especially in complex environment, due to the effect of dynamic environment, the safety, smoothness, and real-time requirement, and the nonholonomic constraints of vehicle. To address the problem of travelling in complex environments which consists of lots of obstacles, a two-layered path planning model is presented in this paper. This method includes a high-level model that produces a rough path and a low-level model that provides precise navigation. In the high-level model, the improved Bidirectional Rapidly-exploring Random Tree (Bi-RRT) based on the steering constraint is used to generate an obstacle-free path while satisfying the nonholonomic constraints of vehicle. In low-level model, a Vector Field Histogram- (VFH-) guided polynomial planning algorithm in Frenet coordinates is introduced. Based on the result of VFH, the aim point chosen from improved Bi-RRT path is moved to the most suitable location on the basis of evaluation function. By applying quintic polynomial in Frenet coordinates, a real-time local path that is safe and smooth is generated based on the improved Bi-RRT path. To verify the effectiveness of the proposed planning model, the real autonomous vehicle has been placed in several driving scenarios with different amounts of obstacles. The two-layered real-time planning model produces flexible, smooth, and safe paths that enable the vehicle to travel in complex environment.

1. Introduction

Nowadays, mobile robots represented by autonomous vehicles have been widely used in transportation, agriculture, and industry [1]. The autonomous vehicle consists of perception, decision-making, and control system, and the study of robot planning system has always been a core problem. One of the key issues is how to generate an appropriate path to reach a goal position without any collision for autonomous vehicle [2–4].

In the previous studies, two categories of searching methods can be found: geometric methods and graph search methods [5]: the calculation of geometry method is very simple, reasonable, and practical, such as spline curve, and it is adaptive to the simple environments. Although the geometric methods have some advantages, these methods

cannot cover up their disadvantages. In the situations of complex dynamic environments, the intelligence and adaptation of these path planning methods need to be enhanced [6].

When searching path based on graph, the ability of the search algorithms that are widely used has been shown in mobile robot path planning. RRT is to perform forward search in continuous coordinates [7]. This algorithm can search rapidly, but when the environment is complex, this algorithm cannot be widely used in the narrow passages. A* algorithm is effective to find a shortest obstacle-free path on basis of a certain decision criteria [8], but the generated path is always composed of straight lines that are hard to execute. D* algorithm has been proposed with the aim of navigating autonomous vehicles in the environment of 2D, and the main advantage of D* method is that it can find an optimal

path when navigating in dynamic environment [9]. This method is often constrained with vehicle kinematic.

At Defense Advanced Research Projects Agency (DARPA) urban challenge 2007, the path planning ability of many research units has been demonstrated. The design scheme and theoretical analysis set up a base for further research. MIT developed a “Talos” vehicle and a path planning method based on a closed-loop RRT [10] is used. Dolgov et al. [11] developed a well-known hybrid A* searching method. In their approach, the 3D kinematic state space of the vehicle and nonlinear optimization for local planning are all used, which lead to a local optimum. In complex dynamic environment, these path planning methods implemented in urban road do not prove their practicability and effectiveness.

In recent years, in order to avoid multiple dynamic obstacles, FSVM has been used by Suresh et al. [12] to secure a collision-free path. The results draw a conclusion that this method is effective. This method is proved in simulations by constructing the fuzzy rules using simple evaluation data. A real-time path planning algorithm has been put into application by Chu et al. [13]. After generating some paths on the basis of predefined waypoints, the optimal function deals with the selection of final safe and smooth path. Makarem and Gillet [14] have developed a method that contains a navigation function suitable for autonomous vehicle but does not consider the influence of obstacles. “Tentacle method” [15] is another way that has been proposed by Chebly et al. By considering the vehicle as the origin, the method develops a set of virtual tentacles that indicate possible paths for vehicle. Then, the best path is chosen according to the evaluation function. In complex dynamic environment, Moreau et al. [16] have investigated an improved curve planning method. All sensors required for obstacle detection are considered in an autonomous vehicle. Taking equality constraints into account, the planning problem is transferred to an optimal problem. Then, Lagrangian and gradient-based methods are used to cover it. In [17], Tazir et al. make a real-time planning based on the combination of two methods. The robot uses genetic algorithms and Dijkstra algorithm to avoid static obstacles. The wait/accelerate concept leads to travel in local dynamic environment. This method is test-efficient but lacks consideration of robot kinematic constraints. In order to let autonomous vehicles drive along a basic path and at the same time avoid obstacles and meet the vehicle kinematic constraints, researchers have proposed an original integrated local trajectory planning and tracking control (ILTPC) framework [18]. In this framework, MPC based planning method is used, which could satisfy the requirement of vehicle kinematic but cannot meet the need of real time.

In simulations or simple environments, the path planning methods developed by scholars are easy to handle the navigation problem. Despite the fact that doable paths can be found by using these traditional methods, it is difficult to meet the requirement of the involved vehicle dynamic and fluxional constraints. In complex dynamic environment, when carrying out the path planning method, the following

must be considered: (1) The path is often planned once. The path is not suitable when change occurs in dynamic environment, and the changed environment cannot influence the performance of vehicle. (2) To ensure safety, a secure clearance barrier must be generated and verified. (3) There is a real-time performance of planning method. (4) The maximum curvature of generated smooth path must be less than the steering curvature of vehicle. (5) The execution error of control leads to the failure of planning, and the path must be trackable [19].

To address these problems, this paper proposed a two-layered path planning model. This method includes a high-level model that produces a rough path and a low-level model that provides precise navigation according to the rough route computed previously. In the high-level model, the improved Bi-RRT [20, 21] based on steering constraints is used to generate an obstacle-free path while satisfying the nonholonomic constraints of vehicle; the result path guides the planning of low level. When executing improved Bi-RRT, the searching speed and vehicle constraints can be easily considered. The vehicle steering constraints are introduced when the nodes of Bi-RRT are extended. The Bi-RRT with two trees searching in the same time can speed up the planning obviously. In the low-level model, the planning is refining, and the obstacle avoidance, dynamic scene, smoothness, and real-time performance all need to be considered. With these constraints, a Vector Field Histogram (VFH) guided polynomial planning method based on Frenet coordinates is introduced. Borenstein and Koren have developed the VFH path planning algorithm in order to guide the robot CARMEL [22]. The candidate directions around the vehicle have been taken into consideration by VFH. After calculating the effect of surround obstacles, the appropriate direction is chosen. This method can handle the cluttered obstacle scenarios well [23].

In this paper, the Frenet coordinates system is applied to generate local paths [24]. At any point along the curve, the Frenet frame coordinates can be computed analytically. In the low-level model, the vehicle position is transformed to the Frenet coordinates based on the improved Bi-RRT path. According to the result of VFH, the aim point chosen from the improved Bi-RRT path is moved to the most suitable location. Then, a local path is generated from the vehicle to the moved aim point using the polynomial planning method in Frenet coordinates. The method presented in this paper is insensitive when environmental change occurs; also the safety and smoothness of path are guaranteed.

The rest of this paper is organized as follows: Section 2 briefly introduces the system architecture, especially the two-layered planning model framework for autonomous vehicle. Section 3 proposes the improved Bi-RRT planning method in high-level model. Section 4 describes the planning method in low-level model; in this section, a VFH guided polynomial planning method based on Frenet coordinates is detailed. The simulation of two-layered path planning model is shown in Section 5. The description of experiments, results, and future work are provided in Section 6. Finally, the conclusions are given in Section 7.

2. Two-Layered Path Planning Model Structure

Figure 1(a) shows the experimental autonomous vehicle with a suite of sensors. There are one camera, one VLP-16 lidar, and a GPS positioning receiver. These sensors consist of the perception system.

The perception system generates a grid map and passes the map to the decision-making system. According to the cooperation of global path, maneuver, and path planning subsystem in decision-making system, the vehicle is able to deal with different scenarios. Finally, the decision-making system produces a path and transfers the path to the control system of vehicle.

Figure 2 illustrates the model structure of the proposed method. The input of high-level model is a global grid map and GPS positioning information. The effective lidar scanning range is about 50 meters; all detected objects are filled to the global grid map, although there may be some inaccuracy caused by the sensor noise. The high-level model based on steering constraint quickly generates a collision-free path and transmits the path to the low-level model.

In low-level model, besides the improved Bi-RRT path, the local grid map which is $20\text{ m} \times 20\text{ m}$ and GPS information are all treated as the input. The local grid map is changing in real time and this close-range map is accurate. According to the preview follower theory [25, 26], the aim point that is treated as the planning end point is chosen dynamically from the improved Bi-RRT path with different speed of vehicle. Due to the noise of global grid map or the drawback of Bi-RRT method, the aim point may not be suitable, and perhaps it is too close to the barrier. Then, AFH method is introduced to find out the most suitable direction. After obtaining the best aim point, Frenet coordinates are set up on the grounds of base line generated by improved Bi-RRT. In Frenet space, the final local path is obtained through the polynomial planning method.

3. The High-Level Path Planning Method Based on Improved Bi-RRT

3.1. Basic Bi-RRT Algorithm. Construct the search tree simultaneously in the initial state and the target state is a principle of Bi-RRT. The initiative procedure of Bi-RRT is not different from the basic RRT algorithm, as shown in Figures 3 and 4, and the tree is expanded after generating random points. The search process of Bi-RRT is shown in Figure 3. Figure 5 shows the searching schematic diagram of basic Bi-RRT algorithm. The basic Bi-RRT algorithm that is more effective is shown in Figure 4.

3.2. Vehicle Kinematical Model. The nonholonomic constraint of autonomous vehicle must be considered in complex environment, owing to the fact that there are always some tortuous lines in final path. Once the curvature of path exceeds the maximum curvature of vehicle, the control system is hard to deal with the path tracking problem. Observing Figure 6, it is a simplified model of vehicle. The

kinematics equation can be described as follows and Table 1 shows the parameters of vehicle.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ k \end{bmatrix} v. \quad (1)$$

According to the kinematic equation mentioned above, the radius and the radius change rate can be explained as

$$|k| \leq k_{\max} = \frac{1}{L} \tan \phi_{\max}, \quad (2)$$

$$|\sigma| \leq \sigma_{\max} = \frac{1}{L} \omega_{\max} \leq \frac{1}{L \cos^2 \phi} \omega_{\max}.$$

In the expansion of Bi-RRT nodes, the steering radius of vehicle should be considered.

3.3. Improved Bi-RRT Algorithm. Based on the above analysis of vehicle path planning problem, this paper proposes a set of path planning algorithms suitable for autonomous vehicles based on the RRT framework. The steps of the algorithm are shown as the improved Bi-RRT algorithm. There are two main improvements in the algorithm, as Figure 7 shows. First, the sampling space is limited by the vehicle steering constraint, which can avoid searching in the entire sampling space, thus speeding up the convergence speed of the algorithm. The second point is to restrict the generation of nodes according to the steering constraints, so that the generated paths are more in line with the dynamic constraints of the vehicle.

3.3.1. A Probabilistic Target Bias Sampling Strategy Based on Steering Constraints. Rapidly-exploring Random Tree planning method or its expansion is a random search in the global scope during the random sampling process, which will lead to a lot of unnecessary searches and waste computing resources. To solve this problem, in order to speed up the random tree to the target point, the improved method used in this paper is as follows: in each growth process of the random tree, the sampling space is limited based on the steering constraint, thereby avoiding a lot of unnecessary searches. As shown in Figure 8, for the entire T-tree, when looking for the nearest neighbour, it is not necessary to search the entire tree. The permissible steering area for the vehicle is determined in the generated subtree, and then the nearest point is found from the corresponding area.

To enable the initial state and the target state to meet faster, this paper uses the target bias strategy in the initial expansion stage of the random number. The random probability is used to determine whether \mathbf{q}_{rand} is the target point or the random point. Set a target bias threshold \mathbf{q}_{prob} , and then get a stochastic number from zero to one when expanding. When $0 < p < \mathbf{q}_{\text{prob}}$, the random tree grows toward the target point; when $\mathbf{q}_{\text{prob}} < p < 1$, the random tree grows in a random direction. This strategy can explore the space faster and more effectively.

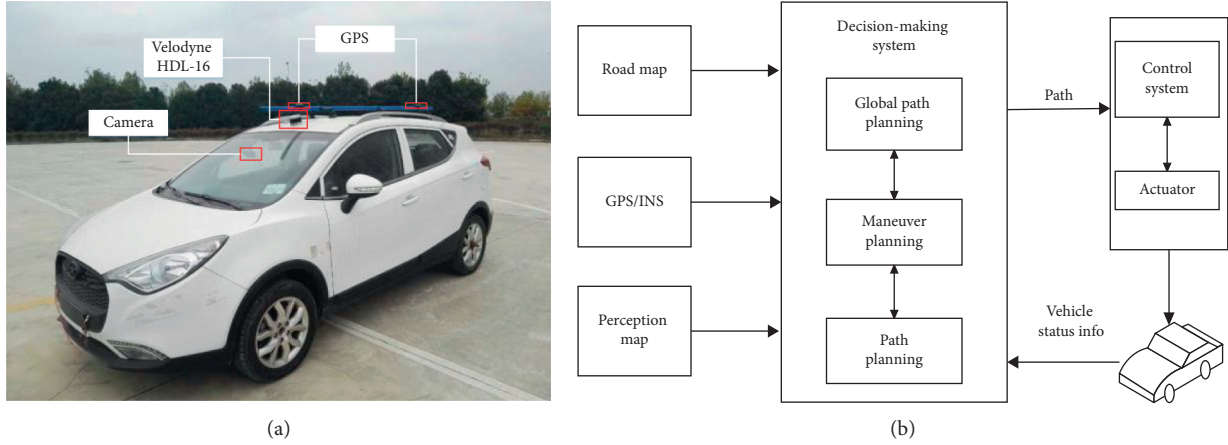


FIGURE 1: (a) The sensor configuration of the autonomous vehicle. (b) The software architecture of planning system.

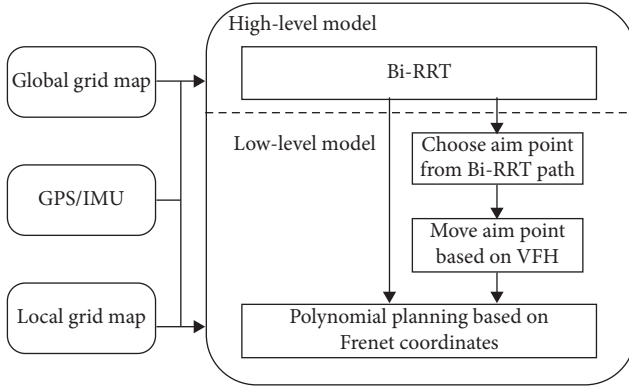


FIGURE 2: The two-layered model architecture.

Algorithm 1 Basic RRT algorithm

```

1:  $V \leftarrow \{q_{init}\}; E \leftarrow \emptyset; G \leftarrow \{V, E\}; i \leftarrow 0;$ 
2: for  $i = 1$  to  $N$  do
3:    $q_{rand} \leftarrow \text{sample}(i); i \leftarrow i + 1;$ 
4:    $q_{nearest} \leftarrow \text{Nearst}(G, q_{rand});$ 
5:    $q_{new} \leftarrow \text{Extend}(q_{nearest}, q_{rand});$ 
6:   if  $\text{ObstacleFree}(q_{nearest}, q_{new})$  then
7:      $V \leftarrow V \cup \{q_{new}\};$ 
8:      $E \leftarrow E \cup \{q_{nearest}, q_{new}\};$ 
9:   end if
10: end for
11: return  $G$ 

```

FIGURE 3: Basic RRT algorithm.

3.3.2. The Extension Strategy with Steering Constraints.

Considering the dynamic constraints of vehicles, this paper presents that, when searching for the nearest point, firstly, reduce the sampling space based on the steering constraint of vehicle, and then K-D tree is applied to search the nearest points using the Euclidean distance. Finally, the closest point can meet the maximum curvature constraint. As shown in Figure 9, although the Euclidean distance between q_i and q_{rand} is closer, because $\theta_i > \theta_j$, q_j is selected as the nearest neighbour node. In this way, the

Algorithm 2 Bi-RRT algorithm

```

1:  $V_1 \leftarrow \{q_{init}\}; E_1 \leftarrow \emptyset; G_1 \leftarrow \{V_1, E_1\};$ 
2:  $V_2 \leftarrow \{q_{init}\}; E_2 \leftarrow \emptyset; G_2 \leftarrow \{V_2, E_2\}; i \leftarrow 0;$ 
3: for  $i = 1$  to  $N$  do
4:    $q_{rand} \leftarrow \text{sample}(i); i \leftarrow i + 1;$ 
5:    $q_{nearest} \leftarrow \text{Nearst}(G_1, q_{rand});$ 
6:    $q_{new} \leftarrow \text{Extend}(q_{nearest}, q_{rand});$ 
7:   if  $\text{ObstacleFree}(q_{nearest}, q_{new})$  then
8:      $V_1 \leftarrow V_1 \cup \{q_{new}\};$ 
9:      $E_2 \leftarrow E_2 \cup \{q_{nearest}, q_{new}\};$ 
10:     $q'_{nearest} \leftarrow \text{Nearst}(G_2, q_{new});$ 
11:     $q'_{new} \leftarrow \text{Extend}(q_{nearest}, q_{rand});$ 
12:    if  $\text{ObstacleFree}(q'_{nearest}, q'_{new})$  then
13:       $V_2 \leftarrow V_2 \cup \{q'_{new}\};$ 
14:       $E_2 \leftarrow E_2 \cup \{(q'_{nearest}, q'_{new})\};$ 
15:      while  $q'_{nearest} \neq q_{nearest}$  do
16:         $q''_{new} \leftarrow \text{Extend}(q'_{nearest}, q_{new});$ 
17:        if  $\text{ObstacleFree}(q''_{new}, q'_{new})$  then
18:           $V_2 \leftarrow V_2 \cup \{q''_{new}\};$ 
19:           $E_2 \leftarrow E_2 \cup \{(q''_{new}, q'_{new})\};$ 
20:           $q'_{new} \leftarrow q''_{new};$ 
21:        else break;
22:      end if
23:    end while
24:    end if
25:    if  $q'_{new} = q_{new}$  then return  $G$ 
26:    end if
27:    if  $|V_2| < |V_1|$  then  $\text{Swap}(V_1, V_2);$ 
28:    end if
29:  end if
30: end for

```

FIGURE 4: Basic Bi-RRT algorithm.

requirements for nonholonomic constraints of the vehicle will be satisfied by the obtained path. The metric function between random sampling point and adjacent point is defined as follows:

$$\text{Dist} = w_1 D + w_2 A, \quad (3)$$

where D and A are the normalization values of Euclidean distance and angle, respectively. Based on the experimental

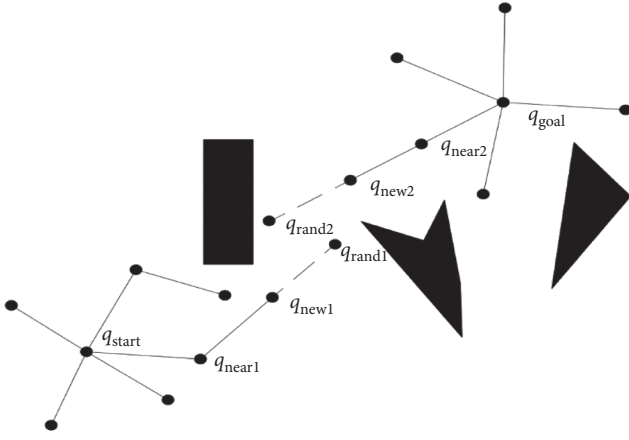


FIGURE 5: Schematic diagram of basic Bi-RRT algorithm.

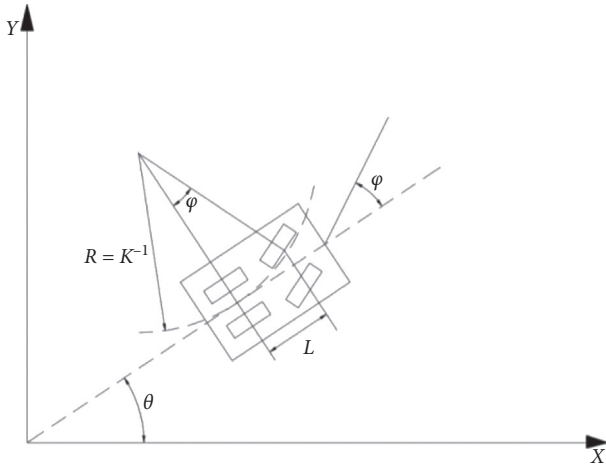


FIGURE 6: Vehicle kinematics model.

data, $w_1 = 0.55$ and $w_2 = 0.45$ were set in this paper. D and A , respectively, are

$$D = \frac{d_{\max} - d_i}{d_{\max}}; \quad (4)$$

$$A = \frac{\theta_{\max} - \theta_i}{\theta_{\max}}.$$

3.3.3. Postprocessing Method. Due to the random sampling of RRT algorithm, the generated paths are usually very tortuous and extremely unsmooth, especially in complex obstacle environments. It is difficult to effectively track them. Meanwhile, too many fold points will affect the ride comfort of the vehicle; it is unacceptable for path planning. Therefore, pruning and smoothing of the improved Bi-RRT path are required.

According to the maximum curvature constraint and the continuity of curvature, the safe nodes in the whole T-tree are deleted to achieve the pruning effect. As shown in Figure 10, q_1 to q_8 and q_{10} to q_{15} are safe, so q_1 to q_8 and q_{10} to q_{15} can be connected directly. If the pruned curve does not

TABLE 1: The parameters of vehicle.

Parameter	Definitions
L	Wheelbase
θ	The orientation of vehicle
φ	The front wheel steering wheel
R	The vehicle curvature radius
v	The vehicle linear velocity
ω	The vehicle yaw rate

Algorithm 3 Improved Bi-RRT Algorithm

```

1:  $V_1 \leftarrow \{q_{init}\}; E_1 \leftarrow \phi; G_1 \leftarrow \{V_1, E_1\};$ 
2:  $V_2 \leftarrow \{q_{init}\}; E_2 \leftarrow \phi; G_2 \leftarrow \{V_2, E_2\}; i \leftarrow 0;$ 
3: repeat
4:    $p \leftarrow \text{rand}(0, 1);$ 
5:   if  $p > p_{\text{prob}}$  then
6:      $q_{\text{rand}} \leftarrow \text{sample}(i); i \leftarrow i + 1;$ 
7:      $q_{\text{nearest}} \leftarrow \text{Nearst}(G_1, q_{\text{rand}});$ 
8:      $q_{\text{new}} \leftarrow \text{Extend}(q_{\text{nearest}}, q_{\text{rand}});$ 
9:   else  $q_{\text{rand}} = q_{\text{goal}}; i \leftarrow i + 1;$ 
10:  end if
11:  Same as Bi-RRT
12: until find a collision-free path from  $q_{\text{start}}$  to  $q_{\text{goal}}$ 
13: return  $S \leftarrow \text{PostProcess}(G)$ 
14: function NEARSTAREA( $G, q_{\text{rand}}$ )
15:   $G_{\text{Area}} \leftarrow \text{Area}(q_{\text{start}}, q_{\text{rand}});$ 
16:   $\text{Dist}_{\max} \leftarrow -\infty;$ 
17:  for all  $q_i$  in  $G_{\text{Area}}$  do
18:     $\text{Dist}_{\max} \leftarrow \text{Dist}(q_i, q_{\text{rand}});$ 
19:    if  $\text{Dist} > \text{Dist}_{\max}$  then
20:       $\text{Dist}_{\max} \leftarrow \text{Dist}; q_{\text{nearest}} \leftarrow q_i;$ 
21:    end if
22:  end for
23: return  $q_{\text{nearest}}$ 
24: end function

```

FIGURE 7: Improved Bi-RRT algorithm.

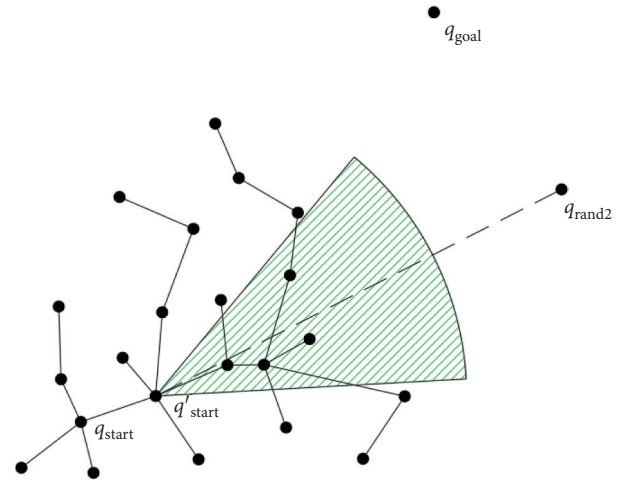


FIGURE 8: A search based on the steering constraint.

satisfy the maximum curvature constraint, the necessary nodes are inserted. The contrast effect before and after pruning is shown in Figure 11.

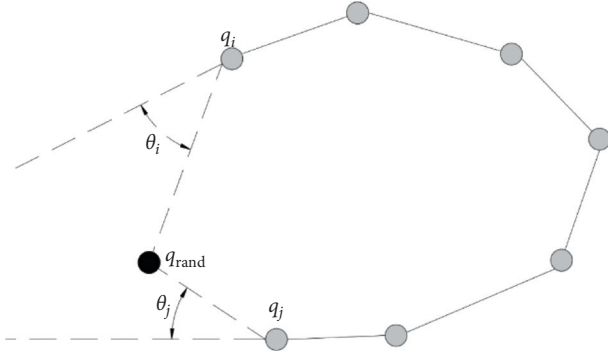


FIGURE 9: Schematic diagram for selecting the nearest node.

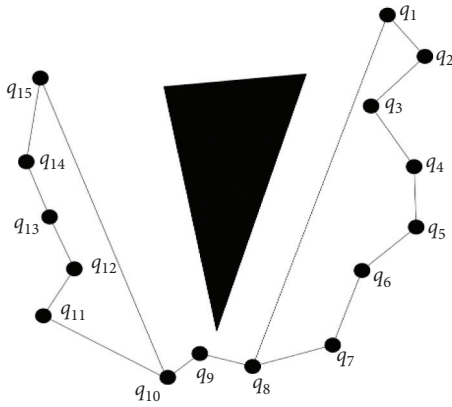


FIGURE 10: Pruning principle.

3.4. *Simulations of Improved Bi-RRT.* To test the effectiveness and adaptability of improved Bi-RRT planning method in high-level model, the simulations are applied. Assume that the grey and black points mean the start and end point separately. The black line refers to the planned path.

Figures 12 and 13 indicate the Bi-RRT and improved Bi-RRT planning results in simple or complex obstacle environments. It can be observed that the improved Bi-RRT method reduces expanding nodes successfully, so the calculation time is reduced. The final path not only satisfies the requirements of vehicle kinematic constraints but also is very smooth.

4. The Low-Level Path Planning Method Based on VFH-Guided Polynomial Planning in Frenet Coordinates

4.1. *Local Aim Point Selection Based on VFH.* In local path planning, the point that is chosen from the improved Bi-RRT path is treated as the local aim point. In dynamic environment, it is hard to make a precise description about the scenario in a big range, so the aim point must be adjusted in real time.

VFH algorithm will divide the environment around the vehicle into many small pieces, called angular sectors. In each angular sector, the risk magnitude can be calculated and then polar histogram is obtained. The best sector is

chosen according to the polar histogram and distance to origin aim point finally. This method is calculated fast.

Instead of using the entire grid map of sensor range, only a small circle around the vehicle can be used to determine the new direction of movement. This space is called the active window. The vehicle is located in the center of the window, the window can move with the vehicle, and its shape is a square or circle. In this section, the circle is introduced for VFH.

4.1.1. *Active Region.* Observing Figure 14, a moving circle window of $L_d \times L_d$ called active region is introduced. The sectors around the vehicle are not all approachable for vehicle, due to the kinematic constraints. When these constraints are considered, the new active regions are built [27]. In Figure 15, r_{\min} is the turning radius and s represents the searching step. The effective angle of VFH is defined as follows:

$$\phi \in \left[-\arcsin \frac{s}{2r_{\min}}, \arcsin \frac{s}{2r_{\min}} \right], \quad (5)$$

where ϕ denotes the sector angle. Figure 16 shows the occupancy grid map used in AFH method. All the perception data including the static and moving obstacles are rendered in the grid map. The red grids mean there are obstacles located in the region and the white grids indicate that these regions are obstacle-free. The sectors that consisted of the dark blue grids represent that it is more dangerous for vehicle to travel than other sectors, and the sectors including the light blue grids indicate that these regions are safe for autonomous driving.

4.1.2. *Polar Histogram.* The scanning radius of lidar is d_{\max} and the angular resolution of each sector is chosen as 10° in this paper. When the nearest distance between vehicle and obstacles $d < d_{\max}$, the risk magnitude of each region is defined as

$$H_i = c_v^2 (a - bd^2), \quad (6)$$

where c_v , a , and b are constant; $a = bd_{\max}^2$.

Polar histogram consists of risk magnitude with each sector. The direction of vehicle should be chosen from the sectors whose risk magnitudes are lower. According to the result of polar histogram and distance from the sector to the origin aim point, new aim point is moved to the most suitable region around the vehicle according to the following principle.

$$S_i = w_1 H_i + w_2 A_i, \quad (7)$$

where A_i means the distance between sectors and the origin aim point.

4.2. *Local Path Planning Based on Polynomial Planning Method in Frenet Coordinates.* The positions of the autonomous vehicle in the global coordinates system and Frenet coordinates system are shown in Figure 17. The

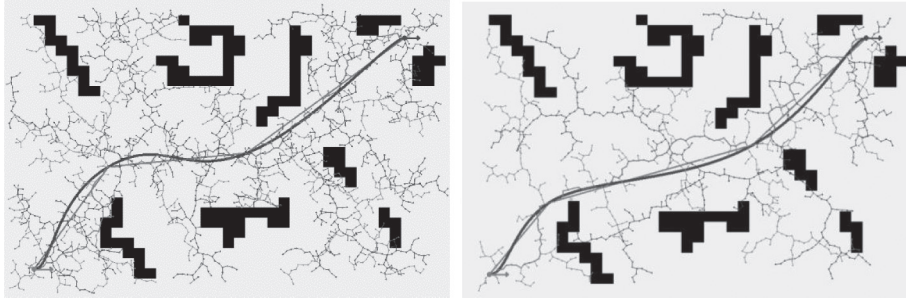


FIGURE 11: Schematic diagram of algorithm (a) before and (b) after pruning.

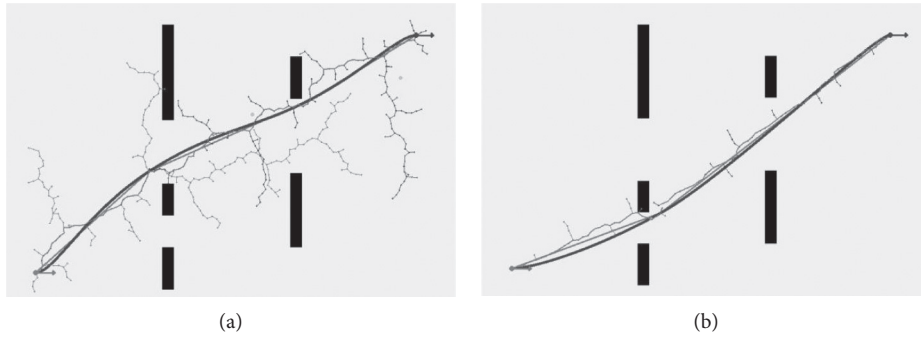


FIGURE 12: The basic and improved Bi-RRT algorithm in simple environment. (a) The basic Bi-RRT method. (b) The improved Bi-RRT method.

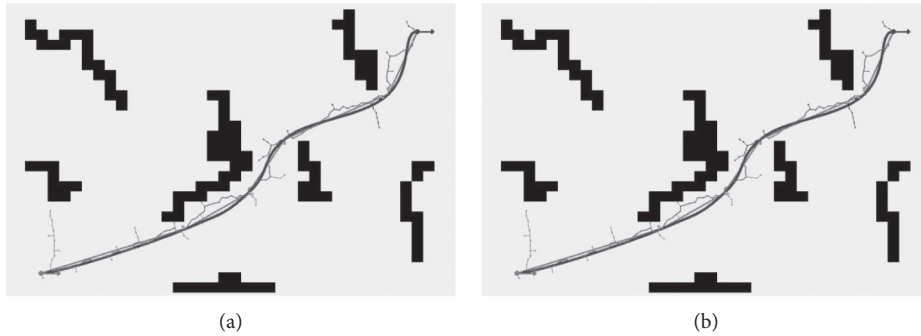


FIGURE 13: The basic and improved Bi-RRT algorithm in complex environment. (a) The basic Bi-RRT method. (b) The improved Bi-RRT method.

Frenet coordinates system is established based on a given reference line (denoted as T_{ref}). T_{ref} can be any curve; in this paper, it is the improved Bi-RRT path. Suppose that the coordinates of the autonomous vehicle in the global coordinates system are (x, y) , and the projection point is F from the vehicle position (x, y) to reference line. The distance between point F and vehicle position (x, y) is the lateral displacement d . The curve distance from the start point of the reference line to the projection point F is the longitudinal displacement s . (s, d) is used to describe the coordinates values of autonomous vehicle in Frenet coordinates system, and the mapping relationship is constructed as follows:

$$(x, y)T_{\text{ref}} \Leftrightarrow (s, d). \quad (8)$$

Based on Frenet coordinates system, the position state of autonomous vehicles is decomposed into the direction of s and d to describe the motion state of the vehicle. For each original coordinates point (x_v, y_v) , the corresponding mapping point (x_p, y_p) on the reference trajectory is determined, and the Frenet coordinates (s_v, l_v) are obtained by

$$l_v = \sqrt{(x_p - x_v)^2 + (y_p - y_v)^2}, \quad (9)$$

$$s_v = S(x_p, y_p).$$

Local path generation problem is divided into the direction of s (longitudinal) and d (lateral); in this paper, only the d direction is taken into consideration; thus the vehicle is assumed at a constant speed.

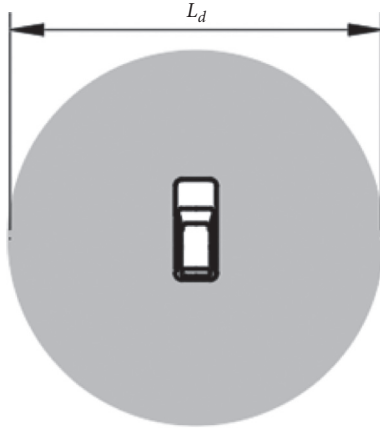


FIGURE 14: Original active regions.

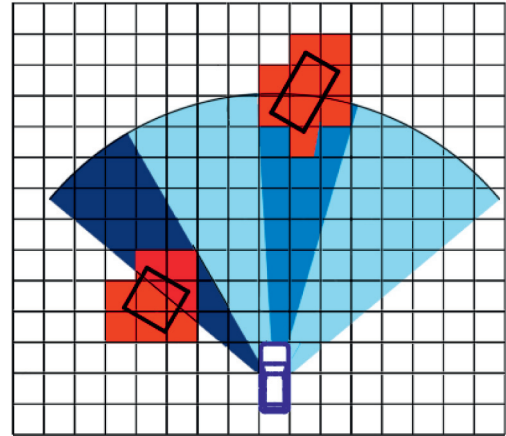


FIGURE 16: The local searching grid map.

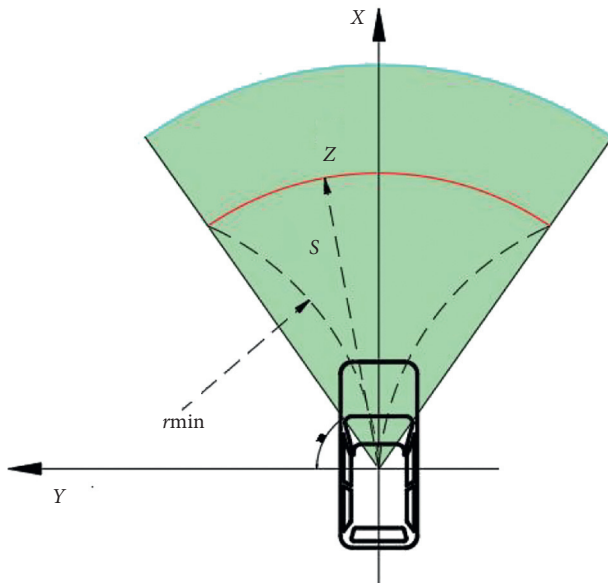


FIGURE 15: The active region of autonomous vehicle.

A polynomial of degree 5 generated in Frenet coordinates is used as the path planning method in low-level. The vehicle position is treated as the start point and the local aim point is chosen from the improved Bi-RRT path [28]. The polynomial path is calculated fast and smooth:

$$p(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 + \alpha_4 t^4 + \alpha_5 t^5. \quad (10)$$

Given the initial configuration $D_0 = \{d_0, \dot{d}_0, \ddot{d}_0\}$, target configuration $D_1 = \{d_1, \dot{d}_1, \ddot{d}_1\}$, and braking time T , the coefficient of quintic polynomial in d direction with respect to time t can be calculated. $d_0, \dot{d}_0, \ddot{d}_0$ represent lateral offset, lateral velocity, and lateral acceleration, respectively.

5. Simulations of Two-Layered Path Planning Model

A normal scenario is designed to show the effect of the proposed method mentioned above. In this scenario, there is one obstacle in the way; the vehicles must avoid the obstacle.

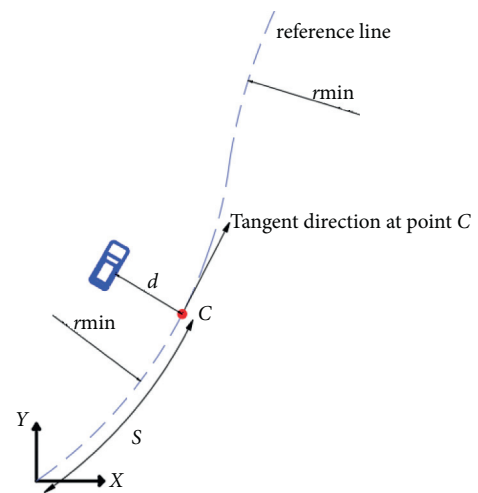


FIGURE 17: Frenet coordinates system.

TABLE 2: The parameters of risk magnitude.

Parameter	Value
C_v	2
a	400
b	1

In this simulation, the expansion step of improved Bi-RRT is set to 0.5 m and the angular resolution of VFH is set to 10° . The main parameters of polar histogram are shown in Table 2.

In Figure 18, the grey and black points mean the start and end point separately. The black line refers to improved Bi-RRT path. Although this method avoids the obstacle successfully, there may be no guarantees about the usage in dynamic environment.

Observing Figure 19(a), in local planning based on polynomial, the blue and green points indicate the origin and moved aim point. The red line represents the local path generated in Frenet coordinates. In normal situations, there is no need to shift aim point because the current sector is



FIGURE 18: The Bi-RRT planning result in high-level model.

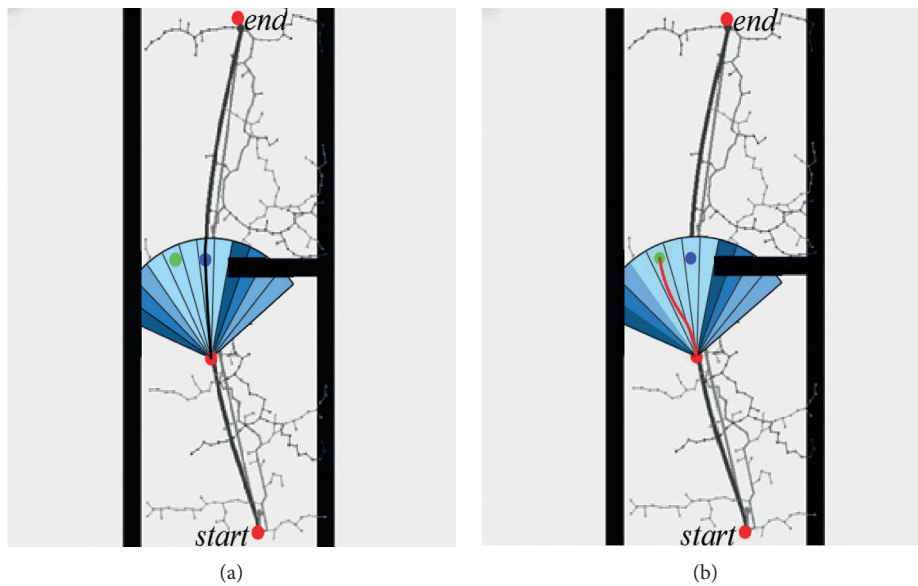


FIGURE 19: The main steps of low-level path planning methods.

the best direction. In Figure 19(a), the distance from origin aim point chosen from Bi-RRT path to obstacle is too near. According to the risk magnitude calculation results in different sectors in Figure 20, the suitable sector that is safe and keeps a short distance from origin aim point is chosen. The aim point is moved and the clearance between vehicle and obstacles is bigger, as shown in Figure 19(b).

6. Experimental Results and Discussion

We conduct a real experiment on the autonomous vehicle for the purpose of verifying the utilities of the path planning method. Figure 21 shows a typical unstructured environment, which consists of lots of obstacles. Three typical scenarios including different amounts of obstacles are used in the experiments. In these scenarios, the speed of vehicle has been set at 15 miles per hour. The low-level planning subsystem does a real-time mission when perception information is updated. Under these conditions, the global grid map is $50\text{ m} \times 50\text{ m}$ and the local grid map is $20\text{ m} \times 20\text{ m}$.

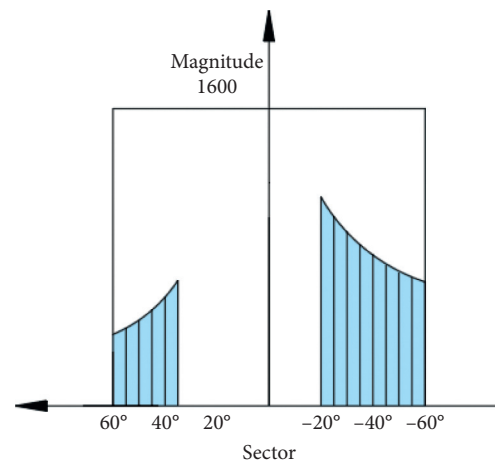
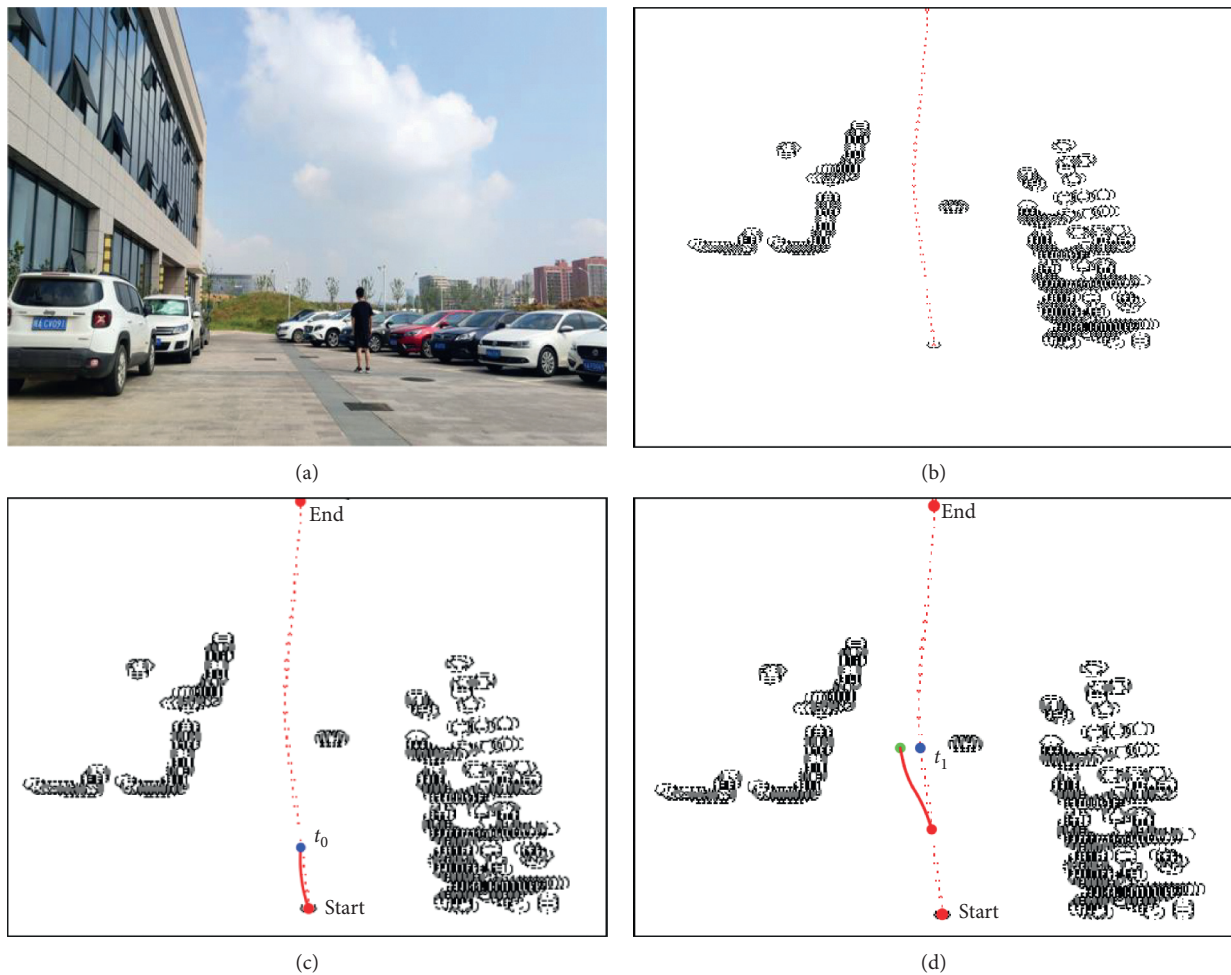


FIGURE 20: The polar histogram based on risk magnitude.

The generated grid maps are plotted according to program. The obstacles are marked with black circles and the red-dotted line indicates the improved Bi-RRT path. The



FIGURE 21: A typical unstructured environment.

FIGURE 22: The results of two-layered path planning method in scenario 1. (a) The experiment scenario. (b) The improved Bi-RRT planning result in high-level model. (c) The planning result of low-level model at time t_0 . (d) The planning result of low-level model at time t_1 .

blue and green points are showing the original and moved aim points when necessary. The local path of vehicle which is executed finally is shown as a red solid line.

When following the path generated in high-level model in Figures 22(b), 23(b), and 24(b), the vehicle may be unsafe due to the incorrect perception results or tracking error. The vehicle must do a real-time planning based on the planning method in low-level model.

In scenario 1, assume that t_0 is the start time. At time t_0 , the local planning result is shown in Figure 22(c). The chosen aim

point from Bi-RRT path is not moved, because the risk magnitude of original travelling direction is less than others. At time t_1 , the vehicle is close to the obstacles, so the aim point is moved to the most suitable region based on the evaluation function. Figure 22(d) shows the planning result in local map.

In scenario 2, at time t_1 and time t_2 , Figures 23(c) and 23(d) indicate the local planning results. When approaching the obstacles, the risk magnitude is increased; the aim points are shifted to the safe location in local grid map.

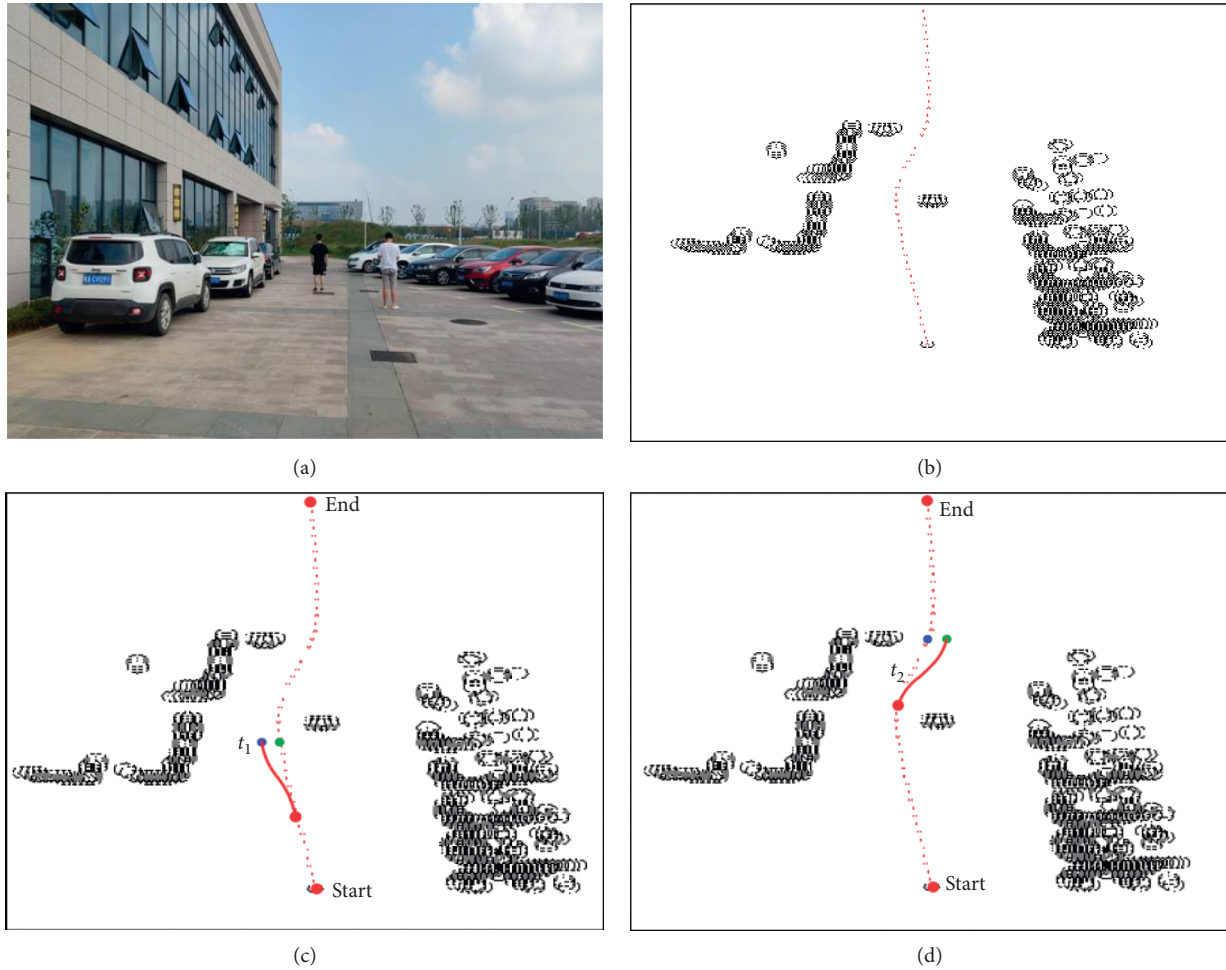


FIGURE 23: The results of two-layered path planning method in scenario 2. (a) The experiment scenario. (b) The improved Bi-RRT planning result in high-level model. (c) The planning result of low-level model at time t_1 . (d) The planning result of low-level model at time t_2 .

Scenario 3 is more complex than other scenarios. This region is small, while one person is around the end point, which makes the outlet narrower. At time t_1 , time t_2 , and time t_3 , the aim points based on improved Bi-RRT are both moved, as Figures 24(c)–24(e) show.

The main evaluation indicators of path planning results are listed in Tables 3 and 4. Table 3 shows the improved Bi-RRT planning indicators in high-level model and Table 4 shows the polynomial planning indicators in low-level model.

It can be observed that, in these scenarios of Table 3, the max curvatures of these paths are 0.09 m^{-1} , 0.1 m^{-1} , and 0.11 m^{-1} separately, and the maximum steering curvature of our autonomous vehicle is 0.22 m^{-1} . These results all meet the nonholonomic requirement of vehicle. Although the generated Bi-RRT paths avoid the obstacles successfully, the distances to the obstacle in Table 3 are not big enough to ensure the safety of vehicle.

Observing Table 4, the max curvature, minimum distance to the obstacles, and planning time in local planning are shown. At different times in these scenarios, the maximum curvatures of the generated paths are also satisfying

the vehicle nonholonomic constraint. Due to the fact of planning in polynomial, the curvature is continuous so that the local path is smooth. Compared to Table 3, the minimum distance to obstacles is bigger due to the moved aim point; all the planning results have a safe clearance to obstacles. The local planning results are smooth and safe. According to Tables 3 and 4, the planning time of two-layered model is less than 100 ms, which conforms to the requirement of real-time planning.

For the sake of verifying the effectiveness of the proposed method mentioned above, a variation of RRT method, RRT*, has been implemented for comparison. Scenario 3 which is most complex is used. Figure 25 shows the planning result of RRT*. Three main indicators of two methods are shown in Table 5. Observing the planning speed, two-layered planning method is slower than RRT* method, and this may be caused by the fact that the two-layered planning model has to plan twice. Because the RRT* method is not considering the steering radius when expanding new nodes, this method does not satisfy the kinematic constraints of vehicle. At last, considering the distance between paths and obstacles, the two-layered model is much safer.

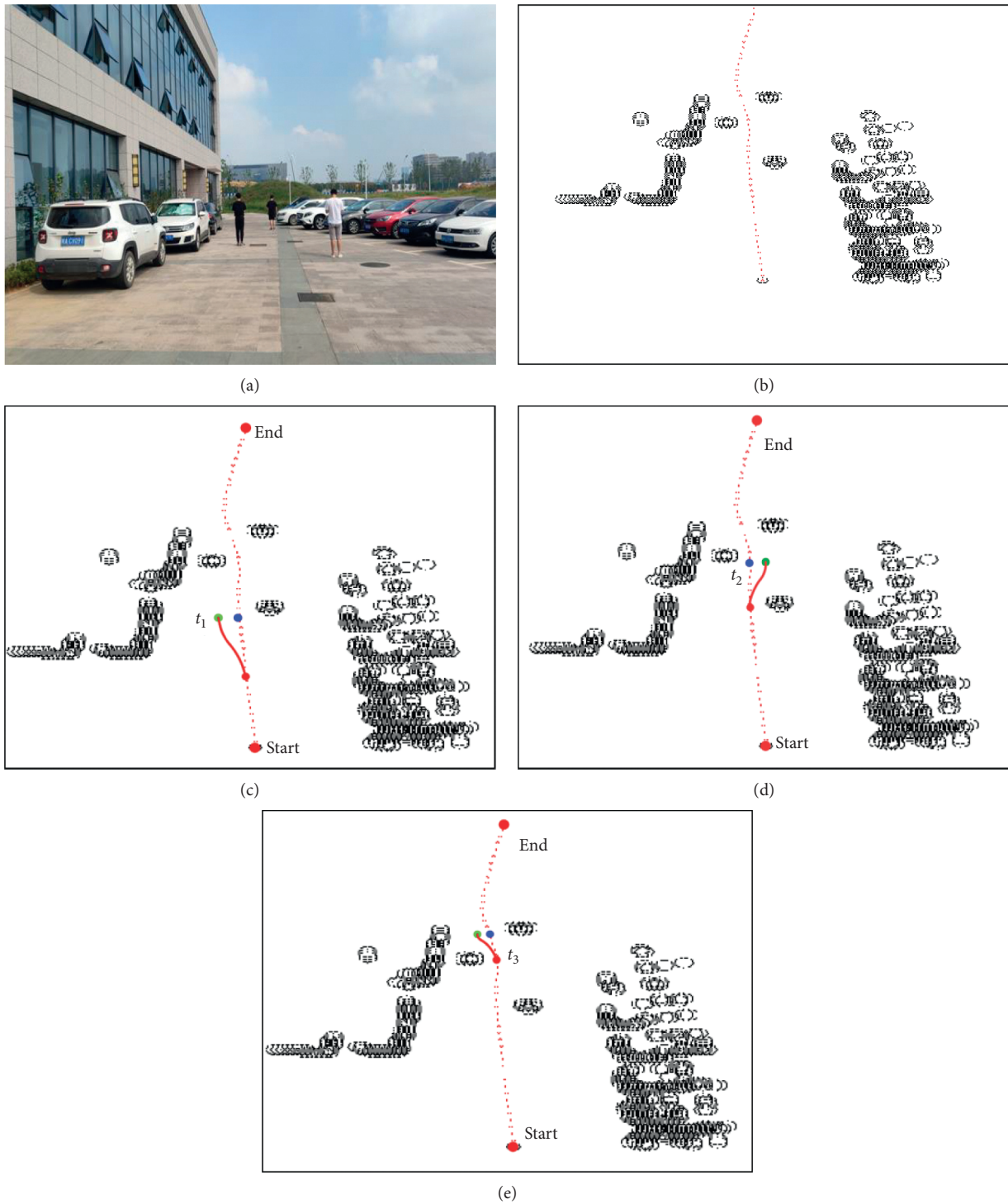


FIGURE 24: The results of two-layered path planning method in scenario 3. (a) The experiment scenario. (b) The improved Bi-RRT planning result in high-level model. (c) The planning result of low-level model at time t_1 . (d) The planning result of low-level model at time t_2 . (e) The planning result of low-level model at time t_3 .

TABLE 3: The main evaluation indicators of path planning results in high-level model.

Scenario	Indicator		
	Planning time (ms)	Minimum distance to obstacle (m)	Max curvature (m^{-1})
1	23.16	1.18	0.09
2	25.76	1.09	0.1
3	28.32	0.95	0.11

TABLE 4: The main evaluation indicators of path planning results in low-level model.

Moment	Indicator		
	Planning time (ms)	Minimum distance to obstacle (m)	Max curvature (m^{-1})
Scenario 1, t_0	38.74	5	0.056
Scenario 1, t_1	40.12	1.46	0.092
Scenario 2, t_1	41.56	1.38	0.094
Scenario 2, t_2	41.38	1.26	0.12
Scenario 3, t_1	40.79	1.40	0.096
Scenario 3, t_2	41.67	1.32	0.099
Scenario 3, t_3	42.68	1.22	0.13

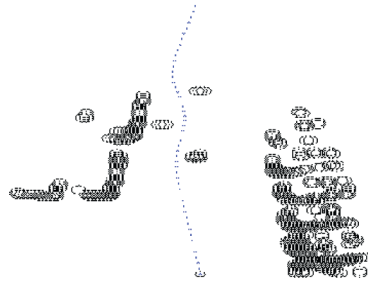


FIGURE 25: The planning result of RRT*.

TABLE 5: The main evaluation indicators of two contrastive path planning results.

Scenario	Indicator		
	Planning time (ms)	Minimum distance to obstacle (m)	Max curvature (m^{-1})
RRT*	32.56	1.08	0.30
Two-layered planning	71.32	1.22	0.13

7. Conclusions

For solving the path planning problem of vehicle when navigating in complex dynamic environments, this paper proposed a two-layered planning model. The high-level model produces a base path and the low-level model provides precise navigation according to the base route computed previously.

The planning results show that this method can realize the navigation of autonomous vehicle with safe and smooth path under the condition of satisfying the nonholonomic constraint of vehicle in complex environment. The vehicle navigates successfully and properly in environments with different amounts of obstacles. Through the experiments, the planning time is not fast enough in a high speed; future works will focus on improving the planning speed and applying more scenarios for testing.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors gratefully acknowledge the help of the team members, whose contributions were essential for the development of the intelligent vehicle. The authors would also like to acknowledge the School of Automotive and Transportation Engineering, Hefei University of Technology, for

supporting this study. This work was supported by the National Natural Science Foundation of China (Grant nos. 51805133 and 61803138).

References

- [1] H. Gao, B. Cheng, J. Wang et al., "Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4224–4231, 2018.
- [2] C. Urmson, J. Anhalt, D. Bagnell et al., "Autonomous driving in urban environments: boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.
- [3] J. Leonard, J. How, S. Teller et al., "A perception-driven autonomous urban vehicle," *Springer Tracts in Advanced Robotics*, vol. 65, pp. 163–230, 2009.
- [4] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, "A predictive controller for autonomous vehicle path tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 92–102, 2009.
- [5] M. Kulich, V. Kozák, and L. Přeucil, "Comparison of local planning algorithms for mobile robots," *Modelling and Simulation for Autonomous Systems*, Springer International Publishing, New York, NY, USA, 2015.
- [6] N. Ganganath and C. T. Cheng, "A 2-dimensional ACO-based path planner for off-line robot path planning," in *Proceedings of the International Conference on Cyber-Enabled Distributed Computing & Knowledge Discovery*, IEEE, Beijing, China, October 2013.
- [7] R. Kala and K. Warwick, "Planning of multiple autonomous vehicles using RRT," in *Proceedings of the IEEE International Conference on Cybernetic Intelligent Systems*, IEEE, Las Vegas, NV, USA, August 2012.
- [8] N. Ganganath, C.T. Cheng, and K. Chi Tse, "Rapidly replanning A*," in *Proceedings of the International Conference on Cyber-Enabled Distributed Computing & Knowledge Discovery*, IEEE Computer Society, Chengdu, China, October 2016.
- [9] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the IEEE International Conference on Robotics & Automation*, IEEE, Washington, DC, USA, May 2002.
- [10] Y. Kuwata, G. A. Fiore, J. Teo et al., "Motion planning for urban driving using RRT," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, IEEE, Nice, France, September 2008.
- [11] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.
- [12] S. Suresh, J. Poornaselvan, and C. Divyapreya, "Optimal path planning approach to grid environment," *Pollack Periodica*, vol. 6, no. 1, pp. 131–140, 2011.
- [13] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1599–1616, 2012.
- [14] L. Makarem and D. Gillet, "Decentralized coordination of autonomous vehicles at intersections," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 13046–13051, 2011.
- [15] A. Chebly, G. Tagne, R. Talj et al., "Local trajectory planning and tracking for autonomous vehicle navigation using clothoid tentacles method," in *Proceedings of the IEEE International Symposium on Intelligent Vehicles (IV)*, IEEE, Seoul, South Korea, June 2015.
- [16] J. Moreau, P. Melchior, S. Victor et al., "Reactive path planning in intersection for autonomous vehicle," *IFAC-PapersOnLine*, vol. 52, no. 5, pp. 109–114, 2019.
- [17] M. L. Tazir, O. Azouaoui, M. Hazerchi et al., "Mobile robot path planning for complex dynamic environments," in *Proceedings of the International Conference on Advanced Robotics*, IEEE, Istanbul, Turkey, July 2015.
- [18] X. Li, Z. Sun, D. Cao, D. Liu, and H. He, "Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles," *Mechanical Systems and Signal Processing*, vol. 87, pp. 118–137, 2017.
- [19] Y. Gu, Z. Li, Z. Zhang, J. Li, and L. Chen, "Path tracking control of field information-collecting robot based on improved convolutional neural network algorithm," *Sensors*, vol. 20, no. 3, p. 797, 2020.
- [20] J. Kuffner and S. LaValle, "RRT-connect: an efficient approach to single-query path planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 995–1001, San Francisco, CA, USA, April 2000.
- [21] X. Z. Zang, W. T. Yu, L. Zhang et al., "Path planning based on Bi-RRT algorithm for redundant manipulator," in *Proceedings of the International Conference on Electrical, Automation and Mechanical Engineering (EAME 2015)*, Phuket, Thailand, July 2015.
- [22] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.
- [23] A. Babinec, D. Martin, F. Duchoň, and A. Vitko, "Modifications of VFH navigation methods for mobile robots," *Procedia Engineering*, vol. 48, pp. 10–14, 2012.
- [24] M. Werling, J. Ziegler, S. Kammel et al., "Optimal trajectory generation for dynamic street scenarios in a Frenet frame," in *Proceedings of the IEEE International Conference on Robotics & Automation*, IEEE, Anchorage, AK, USA, May 2010.
- [25] G. Zhen, G. Xin, L. I. Qian et al., "Driver follower control model with vehicle target velocity based on preview follower theory," *Journal of Jilin University of Technology*, vol. 32, pp. 1–5, 2002.
- [26] Z. Li, L. Chen, Q. Zheng, X. Dou, and Y. Lu, "Control of a path following caterpillar robot based on a sliding mode variable structure algorithm," *Biosystems Engineering*, vol. 186, pp. 293–306, 2019.
- [27] P. Qu, J. Xue, L. Ma et al., "A constrained VFH algorithm for motion planning of autonomous vehicles," in *Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Seoul, South Korea, June 2015.
- [28] S. J. Wang, *Research of Trajectory Planning for Autonomous Driving Based Frenet Coordinates and Sampling*, Lanzhou University of Technology, Lanzhou, China, 2019.