A Safety Navigation Method Integrating Global Path Planning and Local Obstacle Avoidance for Self-Driving Cars in a Dynamic Environment

Jin-kai Yin[1,2] **Wei-ping Fu[1,*]**

[1]School of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an, Shaanxi 710048, China

[2] College of Applied Engineering, Henan University of Science and Technology, Sanmenxia, Henan, 472000, China

*Corresponding author. Tel.: +86 131 8618 6562; fax: +029 8231 2626.

E-mail addresses: Yjk516@163.com(Jin-kai Yin), weipingf@xaut.edu.cn(Wei-ping Fu)

Abstract: In this paper, a novel method for obtaining high-quality paths for self-driving cars in underground parking lots is proposed. Self-driving cars require fast and accurate planning of collisionless path. When the self-driving car arrives at the parking lot, the car downloads the layout from the intelligent system of the parking lot and is assigned a parking space, then the location of the designated parking space and the car are provided by the intelligent system. A global path is planned by the global algorithm according to the location of the parking space and the car. If dynamic or unknown obstacles are detected in the process of moving along the global path, the parameters of obstacles can be estimated by the obstacle-detection algorithm. According to obtained parameters, the local obstacle avoidance path can be planned by the behavioral dynamics method. After completing obstacle avoidance, the car will return to the global path and continue to move toward the target parking space. Finally, the proposed method is simulated by MATLAB, and the results show that the car can safely park in the target parking space. This method simultaneously satisfies the smooth and the real-time requirements of path planning.

Keywords: self-driving car; global path planning; local obstacle avoidance; uncertain dynamic obstacle; dynamic environment.

---

1. Introduction

At present, the self-driving car has aroused extensive attention among many research institutions, and it has also been applied to military, transportation, and other fields [1]. The performance of the path planning determines the intelligence of a self-driving car and is one of the most important core technologies among the related technologies of the self-driving car. The purpose of path planning is to find a continuous and collisionless path from the initial position to the target position, and the path must satisfy the environment constraints, real-time constraints and the kinematic and dynamic constraints of the self-driving car.

Many traffic situations are existed in urban scenarios, the cluttered parking lot is a challenging place to navigate. Domokos Kiss and Gábor Tevesz [2] presented a global planning method for car-like vehicles, producing paths with continuous curvature. The method uses straight segments, CC_{in}-C-CC_{out} triplets (CC-turns), and elementary paths to generate a feasible and human-like solution even in narrow environments. However, it is limited to a pre-set environment. Jong Min Kim and Kyung II Lim [3] adopted an improved Reeds-Shepp curve algorithm for an effective forward and backward auto-stop system. The advantage of the proposed auto-stop system is that the vehicle control system has simple command data which lets the vehicle move forward, backward, and laterally and also tracks travel distance values. However, obstacle detecting is neglected during the parking process.

Guanqun Wang and Tsuneo Naanishi [4] modelled a parking lot with a time-varying graph, and the proposed system applies a time varying shortest-path algorithm and dynamically tunes arc transit times based on planned vehicle routes as well as traffic flow sensor data. Simulation results showed an average travel time reduction of more than 40% for each in the best case. However, in the proposed system, when a road is blocked at time t, for the simplicity of implementation, the transit time of the corresponding arc is chosen to increase at time t by a constant, two times the speed limit of the parking lot. In some cases, this method may lead to an over-tuning problem which means the system tunes an arc's transit time to an unnecessarily large value so that the time-varying shortest-path algorithm cannot output the actual optimal route.

In response to the above deficiencies, in this paper, a safety navigation method integrating global path planning and local obstacle avoidance will be established to make the planned path suitable for the self-driving car. The flow chart of the safety navigation method is shown in Figure 1.

A great amount of path-planning algorithms have been taken from the self-driving car to face the challenges of road networks and driving rules. The most relevant path planning algorithms implemented in motion planning for self-driving car are described below.

The A* algorithm is a graph searching algorithm that enables a fast node search due to the implementation of heuristics. Its most important design aspect is heuristic reducing computation time. But the planning path is not continuous and has many turns [5,6]. The probabilistic road map (PRM) is a graph-based search method that randomly selects N nodes in the planning space. The nodes are then connected, the connection lines with the obstacles are removed, and a path is obtained [7]. The Dijkstra algorithm finds the shortest path in a series of nodes or grid. Suitable for global planning in structured and unstructured environments. But the algorithm is slow in vast areas due to the important amount of nodes. The search is not heuristic. The resulting path is not continuous. Not suitable for real time applications [8-10].

The above-mentioned path planning algorithms have advantages in solving general planning problems, but they must model the obstacles in a deterministic space and these constructed models are very complex. Therefore, these algorithms do not satisfy the needs of self-driving car path planning in a dynamically complex environment.

The Rapidly Exploring Random Tree (RRT) is a data structure algorithm. Its unique advantages can be directly applied to nonholonomic planning and motion planning. The algorithm takes the given initial point as the root node of the random tree and searches quickly and efficiently for a feasible space according to the current environment. Therefore, the RRT algorithm is a randomized algorithm that can explore large space in a relatively short time, which is fast and efficient for the path planning of the self-driving car [11-14].

This paper is structured as follows: Section 2 introduces the global path-planning algorithm. The local obstacle avoidance algorithm is introduced in Section 3. Simulation results and discussion are presented in Section 4. Finally, the conclusion and future work are given in Section 5.

2. Global Path-Planning Algorithm

In recent years, the RRT algorithm has been widely applied and researched in the field of self-driving car path planning. Because its random sampling strategy does not need to preprocess the state space and has fast velocity in the process of searching, this algorithm can effectively solve the problem of path planning in a complex environment. However, there are some defects: (1) the global uniform sampling strategy may lead to unnecessary cost and slow convergence rate; (2) the randomness of the algorithm generates an unsmooth path which may not be directly executed by the non-holonomic constrained self-driving car.

In view of the defects of the classical RRT algorithm, some scholars have improved the algorithm. To improve the search efficiency, Kuffner and LaValle [15] proposed the bidirectional search tree (Bi-RRT) algorithm, i.e., two trees are simultaneously generated at the initial point and the target point to accelerate the convergence rate of the algorithm. In view of the unsmooth path generated by the randomness of the classical RRT algorithm, Fraichar and Scheuer [16] used the method of a convolution curve to smooth the path. However, the convolution curve method does not have a closed-form solution, so the path of the self-driving car cannot promptly and accurately be planned. Lau and Sprunk [17] planned the path of a self-driving car using a Quintic Bézier curve, but the curvature continuity of the path and the constraints of the self-driving car are neglected. Javad and Mansour [18] utilized randomized sampling methods such as the RRT or its derivatives to plan a prior path, which aims to solve the disadvantages of the artificial potential field method (APF), which include local minima and oscillation. A. H. Qureshi [19-20] introduced APF into the RRT* algorithm to accelerate the rate of convergence and to significantly reduce the number of iterations compared with the classical RRT algorithm. By combining reachability-RRT and resolution-complete RRT, Leonard Jaillet [21] improved the success rate of the search, especially the narrow channel, and reduced the number of nodes in searching. However, the constraints of the self-driving car are ignored by these methods, which may lead to the planned path that does not meet the requirements of the car.

2.1 Improved RRT Algorithm

To solve the strong randomness of the generating node of the classic RRT, the idea of gravitational force in APF [22] is introduced into the classic RRT algorithm (hereinafter referred to as A-RRT). The gravitational force guides random trees to grow toward the direction of the target point, which is shown in Figure 2.

The core idea of this improvement is to introduce the target gravitational function $G(n)$ into each node n in the path. n represents the nth point of x_{new} growing from the initial point x_{init}. x_{rand} and x_{goal} represent the random node and the target point, respectively. ρ represents the search step length.

In the gravitational function $G_x = dU_x/d_x = k_p\left|x_{goal} - x_{near}\right|$, x represents the current position vector of the self-driving car, k_p represents the coefficient of the gravitational field, x_{near} represents the nearest point, and $\left|x_{goal} - x_{near}\right|$ represents the absolute value of the geometric distance between the node x_{near} and the target point x_{goal}.

According to the above-mentioned growing process of nodes, the target gravitational function $G(n)$ suitable for RRT algorithm can be constructed as:

$$G(n) = \rho \cdot k_p \cdot \frac{x_{goal} - x_{near}}{\left\|x_{goal} - x_{near}\right\|} \tag{1}$$

When a new leaf node is added through the A-RRT algorithm, the target gravitational function will influence the selection of the new node by calculating the gravity from each node to the target, and then the random tree is guided toward the target to grow [23].

According to Equation (1), the Equation of generating new nodes that introduced the idea of gravitation can be obtained as:

$$x_{new} = x_{near} + \rho\left\{\frac{x_{rand} - x_{near}}{\left\|x_{rand} - x_{near}\right\|} + k_p \times \frac{x_{goal} - x_{near}}{\left\|x_{goal} - x_{near}\right\|}\right\} \tag{2}$$

2.2 Constraint Condition

To make a planned path be effectively applied to the self-driving car, the path can be tracked and then the path must meet the road environment constraints. Assuming B_l and B_r as the left and right boundaries of the road, the generated random tree nodes should be within the boundaries. The coordinates of the node position should meet Equations (3) and (4):

$$B_r \le t_y \le B_l \tag{3}$$

$$P_{ini} \le t_x \le P_{tar} \tag{4}$$

where P_{ini} is the initial point of each extension , P_{tar} is the target point of each extension.

Considering that a car has geometric shape, the width of the car can be expressed as D. Since Equation (3) is the coordinates restriction in the y direction of the nodes, this can be expressed as:

$$B_r - D/2 \le t_y \le B_l + D/2 \tag{5}$$

Assuming the center of mass of the self-driving car moving along the planned path, the curvature of the planned path cannot change too much to ensure stability during driving. If the actual front wheel steering angle has a maximal value $\theta_{\max}$, the connection line between the child node x_b and its parent node x_a is $x_a x_b$, and the connection line between the parent node x_a and its parent node x_{init} is $x_{init} x_a$, the angle β between $x_a x_b$ and $x_{init} x_a$ must satisfy $\beta < \theta_{\max}$. Generally, the θ value is between $30° \sim 40°$. Then the angle constraint can be expressed as:

$$\arctan\left((K_1 - K_2)/(1 + K_1 K_2)\right) < \beta \le \theta_{\max} \tag{6}$$

where K_1 is the curvature of the straight line $x_a x_b$, and K_2 is the curvature of the straight line $x_{init} x_a$.

To ensure that extended points do not intersect with obstacles, the method of elliptically enveloping obstacle and properly enlarging the safe ellipse is adopted to meet obstacle avoidance requirements. If the connection line between the new node and its parent node does not intersect with the safe ellipse, then the expanded new point satisfies the obstacle avoidance requirement. Taking the five-equal partition point $P(x, y)$ on the connection line, then the constraint equation can be expressed as:

$$(x - x_{obs})^2/(s \cdot a)^2 + (y - y_{obs})^2/(s \cdot b)^2 > 1 \tag{7}$$

where (x_{obs}, y_{obs}) are the coordinates of the obstacle, s is the safe ellipse magnification coefficient, the half length of the car $a = 2m$, and the half width of the car $b = 1m$. When s equals $\sqrt{2}$, the safe ellipse just right envelops the rectangular obstacle, Therefore, it is necessary to guarantee $s \ge \sqrt{2}$ from the perspective of security obstacle avoidance.

2.3 The Process of Smoothing the Path

The path planned by the classic RRT algorithm usually has a small range of twists and turns and is discontinuous. To make the path meet the stability and safety requirements of the car while moving, it is necessary to smooth the planned path. The B-spline can locally adjust the path without changing the entire path shape. According to the feature of the B-spline, the path planned by the classic RRT algorithm can achieve the purpose of the smooth path through the method of interpolation. The usually used B-spline is the cubic spline curve.

The proposed A-RRT algorithm does not need an accurate model of the global environment, which can greatly reduce the planning time and improve the real-time performance of the algorithm. Furthermore, the cubic B-spline curve is used to smooth the path generated by the A-RRT algorithm, which can ensure the curvature continuity of the path and satisfy the constraints of the self-driving car.

3. Local Obstacle Avoidance Using Behavioral Dynamics

The environmental information can be detected by the sensors of the self-driving car. When detecting a dynamic obstacle, the behavioral dynamics method is used to plan the local obstacle avoidance path for the self-driving car, and then the self-driving car begins to avoid the obstacle.

According to the behavioral dynamics theory [24, 25], the target can be represented as an attractor and the obstacle can be indicated as a repeller. The attractor can generate a virtual attraction between the target and the current position of the self-driving car in the process of moving. The repeller can generate a virtual repulsive force between the obstacle and the current position of the self-driving car in the process of moving. Firstly, the relationship between virtual attraction and behavioral variables can be established by making the target of the typical driving behavior an attractor. Secondly, the relationship between virtual repulsive force and behavioral variables can be structured by making the obstacle around the self-driving car a repeller. Finally, the behavioral dynamics model of self-driving car path planning can be established by combining the attraction and the repulsive force model.

According to traffic rules, the solid line does not allow the vehicle to cross, the dotted line can be driven over for a short time, but the car cannot ride on the line for a long time. The solid line is defined as strong constraint, and the dotted line is defined as a weak constraint. The strong constraint must be avoided while driving the car. The weak constraint should not be avoided if the car needs to change lanes or overtake another car while driving, and only when the car is being driven in the mode of lane-keeping, for driving safety, must a safe distance be kept between the car and lane.

3.1 Attraction Model

According to the typical driving behavior of the self-driving car, the relationship between the attraction of tending to the target and the behavioral variables including heading angle and velocity can be established [26].

The behavioral variables of the self-driving car are illustrated in Figure 3.

3.1.1 Heading Angle Attraction Model

The heading angle attraction model is established by the typical driving behavior of the self-driving car. The heading angle ψ of the self-driving car finally must be consistent with the direction of the target. Assuming the direction of the target is the attractor, then the heading angle of the self-driving car must satisfy $\psi_{tar} \in [-\pi/2, \pi/2]$ .The heading angle attraction equation of tending to the target can be established according to [27], which can be expressed as:

$$f_{tar} = f_{tar,\varphi}(\psi) = -\lambda_{tar,\varphi} \tan(\psi - \psi_{tar}) \tag{8}$$

where $\lambda_{tar,\psi}$ is the heading angle attraction strength factor, and the magnitude of the virtual attraction will be changed by adjusting $\lambda_{tar,\psi}$. ψ_{tar} is the heading angle between the target and the self-driving car in the world coordinate system $\psi_{tar} = \arctan((P_{tar,y} - P_{veh,y}) / (P_{tar,x} - P_{veh,x}))$. $P_{tar}(P_{tar,x}, P_{tar,y})$ and $P_{veh}(P_{veh,x}, P_{veh,y})$ are the positions of the target point and the self-driving car in the world coordinate system, respectively.

Figure 4 shows the local coordinate system on the self-driving car. The x'-axis direction is the moving direction along the axis of the car, and the y'-axis direction is perpendicular to the axis direction of the car. ϕ_{tar}' is the angle between the direction of vehicle velocity and the x' coordinate, which is a behavioral variable.

To ensure the final velocity direction in line with the axis of the car and the parking space, the specified parking space centerline is set as the target pose attractor.

The pose behavioral dynamics model [28] of the self-driving car can be established as:

$$\dot{\phi} = -\lambda\left|\phi' - \phi_{tar}'\right| \tag{9}$$

where λ is the pose attraction strength factor.

3.1.2 Velocity Attraction Model

In the path planning process of the self-driving car, it is necessary to consider the safe and dynamic characteristics of the car when establishing the velocity attraction model. The velocity of the car must have an upper limit. The contact time can be expressed as $T = d_{tar} / v$, where d_{tar} is the distance between the self-driving car and the target point. The contact time T cannot be too big nor too small, and must meet the safe distance of the car. $T_{\max}$ is the maximum of the contact time $T_{\max} = (d_{tar} - D_s / v)$, where D_s is the safe following distance. Furthermore, the acceleration of the self-driving car $\dot{v}$ must be smaller than the acceleration of the target car $\dot{v}_{tar}$, otherwise, there will be a collision.

According to [29], the velocity attraction equation of tending to the target can be established as:

$$f_{tar,v}(v) = -\lambda_{tar,v}(v - v_{tar})\exp[\frac{(v - v_{tar})^2}{2\sigma_v^2}] \tag{10}$$

where $\lambda_{tar,v}$ is the velocity attraction strength factor, and the size of the virtual attraction of the target point to the self-driving car can be changed by adjusting $\lambda_{tar,v}$. v_{tar} and D_{veh} are the expected velocity and the width of the self-driving car, respectively. σ_v is the range of the attractor, which can be expressed as:

$$\sigma_v = \arcsin(\frac{D_s + D_{veh}}{D_{veh}}) \tag{11}$$

When the target is the ahead moving car, then v_{tar} is the velocity of the target car. Otherwise, the self-driving car will drive at a constant velocity. The current velocity of the self-driving car v_c can be achieved according to the inertial navigation system. The distance S between the self-driving car and the target car can be measured by the millimeter wave radar, and the initial velocity v_o can be set.

The velocity of the target car v_{tar} can be calculated by:

$$v_{tar} = \sqrt{2aS + v_c^2} \tag{12}$$

$$a = \frac{v_c - v_o}{T_{\max}} \tag{13}$$

3.2 Repulsive Force Model

According to the surrounding environment of the self-driving car, the relationship between the repulsive force and the behavioral variables including the heading angle and the velocity can be established.

3.2.1 Heading Angle Repulsive Force Model

If a static or moving obstacle is detected while moving toward the target point, the self-driving car must be able to safely avoid the obstacle and safely reach the target point. ψ_{obs} represents the repeller, which is the unstable point that turns the influence of an obstacle to zero in the behavioral dynamics method.

According to [30], the repulsive force equation of the heading angle can be established as:

$$f_{obs,i}(\psi_{obs,i}) = -\lambda_{obs,i}(\psi - \psi_{obs}) \times \exp(-Cd_{obs,i})\exp(-\frac{(\psi - \psi_{obs,i})^2}{2\sigma_{obs,i}^2}) \tag{14}$$

$$\psi_{obs,i} = \arctan(\frac{P_{obs,i}y - P_{veh}y}{P_{obs,i}x - P_{veh}x}) \tag{15}$$

where $\lambda_{obs,i}$ represents the heading angle repulsive force strength factor, and the repulsive force can be changed by adjusting $\lambda_{obs,i}$. C represents the repulsive coefficient of attenuation with increasing distance. $d_{obs,i}$ represents the distance between the obstacle and the self-driving car. $P_{obs,i}x$ and $P_{obs,i}y$ are the coordinates of the obstacle in the world coordinate system. $\sigma_{obs,i}$ represents the range of a repeller, which can be expressed as:

$$\sigma_{obs,i} = \arcsin(\frac{D_s + D_{veh}}{d_{obs,i} + D_{veh}}) \tag{16}$$

The heading angle repulsive force equation of multiple obstacles can be written as:

$$F_{obs} = \sum_i f_{obs,i}(\psi_{obs,i}) \tag{17}$$

3.2.2 Velocity Repulsive Force Model

In the path planning process, the velocity of the self-driving car is not only related to the distance d_{obs} between the current position of the self-driving car and the obstacle, but also related to the safe distance D_s . Under the premise of guaranteeing the minimal safe distance, the linear velocity of the self-driving car decreases with decreasing d_{obs} .

According to [31], the linear velocity repulsive force equation can be established as:

$$f_{obs,i}(v) = -\lambda_{obs,v}(v - v_{obs,i})\exp[\frac{(v - v_{obs,i})^2}{2\sigma_v^2}] \tag{18}$$

3.3 Behavioral Dynamics Model

According to the above established attraction model and repulsive force model of the heading angle and velocity, the behavioral dynamics model of the self-driving car can be established by the weighting of each attraction and repulsive force. In practical applications, each behavior needs to be coordinated and then used for vehicle behavior control.

Synthesized behavioral dynamics model that includes velocity and heading angle can be established as:

$$\dot{v} = f_v = w_{obs} f_{obs,i}(v) + w_{tar} f_{tar}(v) \tag{19}$$

$$\dot{\psi} = f_\psi = \gamma_{obs} F_{obs} + \gamma_{tar} f_{tar} \tag{20}$$

where w_{obs} , w_{tar} , γ_{obs} and γ_{tar} are the weight coefficients of the behavioral dynamics model. According to the weight of each behavior in the actual model, the force of each behavioral variable can be changed by altering the weight coefficient. To eliminate interference so that the target behavior and obstacle avoidance behavior occur simultaneously, the four weight coefficients are varying with the situation in the simulation process.

4. Simulation and Discussion

To illustrate the effect of the A-RRT algorithm, Figure 5 contrasts the planning results of the A* algorithm and the A-RRT algorithm. The initial parameters of the self-driving car are CarPos = [165, 20, 90, 5, 8, 90, 1] (initial x coordinate, initial y coordinate, heading angle, velocity, perceptive distance, perceptive angle, size of car), and the target position parameters are TargetPos = [25, 175, 90] (x and y coordinates of the target parking space). The obstacle position parameters are ObstaclePos = [90, 95]. The horizontal length is 200m. MATLAB

2014a is used for simulating the hybrid algorithm. The comparison of the results can be seen from the graph and the table:

Compared to the A* algorithm, the path of the A-RRT algorithm is smoother, there are no frequent large curvature changes. To objectively evaluate the performance of the algorithm, two algorithms were planned 30 times in the same experimental scene because of the randomness of A-RRT algorithm. Table 1 shows the partial results of the 30 experiments, which includes 5 planning time, 5 path length, times of success and mean square deviation of path curvature.

According to the results of the simulation experiments in Figure 5 and the data in Table 1, A* algorithm is better than A-RRT algorithm in planning time and path length, but A-RRT algorithm has significant advantages in the aspect of path smoothing and meeting car constraints. So the A-RRT algorithm is more suitable for the self-driving car.

Figure 6 illustrates the process of obstacle avoidance. The self-driving car and the obstacles are located in initial position at t_1 moment. A millimeter wave radar and two ultrasonic radars are arranged on the head of the self-driving car. The detection range is expressed as a sector, and the angle of the sector is 120°. The car and the dynamic obstacle keep their own velocity vector before t_3 moment, and the car moves toward the target point at the speed of 5km/h. When the dynamic obstacle is detected at t_3 moment, it is indicated in red, and then the path of obstacle avoidance is planned by the behavioral dynamics method. The average time of 20 times obstacle avoidance path planning is 0.072s, and the planning time decreases with improving computer configuration. Furthermore, the whole time of human from discovery to brain judgement to manipulation of the hands and feet is called the reaction time, which is about 0.38s, so behavioral dynamics obstacle avoidance meets the requirement of real time.

Monte Carlo method is a stochastic simulation method that uses random numbers or some kind of probability phenomenon to simulate real-world problems. Since the self-driving car may randomly stop during parking to the designated parking space, therefore, Monte Carlo method can be used to simulate the path between the random initial state and the target state. As the path length is proportional to the navigation error of the goal point, the navigation path planning problem can be reduced to the shortest path problem from the initial state to the target state under the premise of the maximum allowable error. The process of safety navigation is shown in Figure 7.

Figure 7(a) shows the environment of the underground parking lot of the Wal-Mart International Shopping Center. The area of the underground parking lot is 26,000 m^2, and there are 800 parking spaces. The size of the parking space is 5.5×2.4 m. Moreover, the width of the parking space line and the lane line are 9 cm, and the lane width is 6 m. Figure 7(b) shows the layout of the parking lot. Figure 7(c) shows the global and the partial enlarged drawing of the planning path. The entrance coordinate of the parking lot is [885, 55], which is expressed in green, and it can be used as the initial point. The designated parking space coordinate is [658, 110], which is expressed in yellow, and it can be used as the target point. The global path can be planned by using the A-RRT algorithm, and then, the proposed obstacle detection algorithm is used to estimate the obstacle parameters when the dynamic obstacle is detected. The black rectangle represents the dynamic obstacle, the larger rectangle that wraps the dynamic obstacle represents the area of potential collision. According to the initial point coordinates and foregone values, the parameters of the dynamic obstacle can be calculated. The possible radius and the velocity of the dynamic obstacle are in the range of 0.5-0.8m and the range of 3.5-4m/s, respectively. The estimated parameters of the dynamic obstacle are transmitted to behavioral dynamics for obstacle avoidance, for which the self-driving car obstacle avoidance process can be clearly seen in Figure 7(c), and then the self-driving car returns to the global path. Finally, the self-driving car safely reaches the designated parking space.

5. Conclusion

For the typical self-driving scene of the parking lot, an efficient trajectory planning framework for the self-driving car is presented. The APF is introduced into the classical RRT algorithm to accelerate the convergence speed and obtain an optimal solution, the convergence rate has accelerated nearly four times, the smoothness and curvature continuity of the path have been greatly improved as shown in Table 1, and the improved algorithm only requires 68% of original iterations to find a solution. Moreover, constraints of the road and the self-driving car were considered during nodes expansion, making the planning path meet the self-driving car requirements. The behavioral dynamics method is used to plan an obstacle avoidance path based on the dynamic obstacle parameters, and the average time of 20 times replanning path is 0.072s, which meets the real-time requirement. The MATLAB experimental results show that the hybrid path planning model has the good real-time performance and reliability, and the self-driving car along the desired path can safely bypass the obstacle and reach the target point. The future work is to apply the method on a real self-driving car to perform real-life tests and performance measurement.

Conflict of Interests

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

Acknowledgments

This work is supported by Project (10872160) of the National Natural Science Foundation of China, Shaanxi Province Important Disciplines of the Vehicle Engineering Construction of China, Scientific Research Projects of Shaanxi Province Education Office Key Laboratory (13JS070), and Scientific Research Project in Shaanxi Province Department of Education (14jk1796).

Reference

[1] Li, A.J., Li, S.M. and Li, D.R. “On the trajectory planning's key technologies for intelligent vehicle”, Mechanical Science and Technology for Aerospace Engineering,32(7), pp.1022-1026(2013).
[2] Domokos, K. and Gábor, T. “Autonomous Path Planning for Road Vehicles in Narrow Environments: An Efficient Continuous Curvature Approach”, Journal of Advanced Transportation, 2017(2), pp.1-28(2017).
[3] Kim, J.M., Lim, K.I. and Kim, J.H. “Auto Parking Path Planning System Using Modified Reeds-Shepp Curve Algorithm”, 2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence, pp.311-315(2014).
[4] Wang, G.Q., Tsuneo, N. and Akira, F. “Time-Varying Shortest Path Algorithm with Transit Time Tuning for Parking Lot Navigation”, TENCON 2015-2015 IEEE Region 10 Conference, pp.1-6(2015).
[5] Du, M.B., Mei, T. and Chen, J.J. “RRT-based Motion Planning Algorithm for Intelligent Vehicle in Complex Environments”, ROBOT, 37(4), pp. 443-450(2015).
[6] Dolgov, D., Thrun, S., Montemerlo, M., et al. “Path planning for autonomous vehicles in unknown semi-structured environments”, The International Journal of Robotics Research, 29(5), pp. 485–501(2010).
[7] Elbanhawi, M. and Simic, M. “Sampling-Based Robot Motion Planning: A Review”, IEEE Access, 2(1), pp. 56-77(2014).
[8] Likhachev, M. and Ferguson, D. “Planning Long Dynamically-Feasible Maneuvers for Autonomous Vehicles”, International Journal of Robotics Research (IJRR), 2009.
[9] Kushleyev, A. and Likhachev, M. “Time-bounded lattice for efficient planning in dynamic environments”, 2009 IEEE International Conference on Robotics and Automation, pp. 1662–1668(2009).

[10] Guo, Q., Zhang, Z. and Xu, Y. “Path-planning of automated guided vehicle based on improved Dijkstra algorithm”, 2017 29th Chinese Control and Decision Conference (CCDC),2017.
[11] Dong, Y. and Camci, E. “Faster RRT-based Nonholonomic Path Planning in 2D Building Environments Using Skeleton-constrained Path Biasing”, Journal of intelligent & Robotic Systems, 89(3-4), pp. 387-401(2018).
[12] Otte, M. and Frazzoli, E. “RRTX: Asymptotically Optimal Single-Query Sampling-Based Motion Planning with Quick Re-planning”, International Journal of Robotics Research,35(7), pp.1-35(2015).
[13] Song, X.L., Zhou, N., Huang, Z.Y., et al. “An Improved RRT Algorithm of Local Path Planning for Vehicle Collision Avoidance”, Journal of Hunan University (Natural Science), 44(4), pp.30-37(2017).
[14] Jeon, J. H., Cowlagi, R.V., Peters, S.C., et al. “Optimal motion planning with the half-car dynamical model for autonomous high-speed driving”, 2013 American Control Conference, pp. 188–193(2013).
[15] Kuffner, J.J. and LaValle. S.M. “RRT-connect: An efficient approach to single-query path planning”, IEEE International Conference on Robotics & Automation, pp. 995-1001(2002).
[16] Fraichard, T. and Scheuer, A. “From Reeds and Shepp’s to continuous curvature paths”, IEEE Transactions on Robotics, 20(6), pp. 1025-1035(2004).
[17] Lau, B., Sprunk, C. and Burgard, W. “Kinodynamic motion planning for mobile robots using splines”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2427-2433(2009).
[18] Amiryan, J. and Jamzad, M. “Adaptive Motion Planning with Artificial Potential Fields Using a Prior Path”, The 3rd RSI International Conference on Robotics and Mechatronics, pp. 731-736(2015).
[19] Qureshi, A.H., Iqbal, K.F., Qamar, S.M., et al. “Potential Guided Directional-RRT* for Accelerated Motion Planning in Cluttered Environments”, 2013 IEEE International Conference on Mechatronics and Automation, pp. 519-524(2013).
[20] Qureshi, A.H., Mumtaz, S. and Iqbal, K.F. “Adaptive Potential Guided Directional-RRT*”, The IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1887-1892(2013).
[21] Jaillet, L., Hoffman, J. and Berg, J.V.D. “EG-RRT: Environment-Guided Random Trees for Kinodynamic Motion Planning with Uncertainty and Obstacles”, 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2646-2652(2011).
[22] Sun, L.B., Liu, Y., Sun, J.Z., et al. “Path Planning Model Based on Mixed Perception Information”, Computer Engineering, 36(10), pp. 32-35(2010).
[23] Lv, W.X., Zhao, L.J., Wang, K., et al. “Efficient exploration of unknown environments with RRT-boundary constraint”, Hua Zhong Univ. of Sci. &Tech. (Natura Science Edition), 39, pp. 366-369(2011).
[24] Y, S.Q., Fu, W.P. and Li, D.X. “Application of Dynamic System Theory to Mobile Robot Navigation”, Mechanical Science and Technology for Aerospace Engineering, 29, pp. 100-104(2010).
[25] Y, S.Q., Fu, W.P., Li, D.X., et al. “Research on Application of Genetic Algorithm for Intelligent Mobile Robot Navigation Based on Dynamic Approach”, IEEE International Conference on Automation and Logistics, pp. 898-902(2007).
[26] Bicho, E., Mallet, P. and Schoner, G. “Using attractor dynamics to control autonomous vehicle motion”, Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Societh, pp. 1176-1181(1998).
[27] Fu, W.P., Zhang, P.F. and Yang, S.Q. “Behavioral dynamics of mobile robot and rolling windows algorithm to path planning”, Computer Engineering and Applications, 45, pp. 212-214(2009).
[28] Wang, W.Y., Fu, W.P., Wei, M.M., et al. “Behavior dynamics method for the motion planning of the end-effector of autonomous manipulator”, Journal of Xi’an University of Technology, 32(4), pp. 468-474(2016).
[29] Fu, W.P., Hao, D.P., Yang, S.Q., et al. “Study on the navigation method of behavior dynamics in mobile robot”, Mechanical Science and Technology for Aerospace Engineering, 32(10), pp. 1488-1491(2013).
[30] Han, G.N., Fu, W.P., Hao, D.P., et al. “Study on the Motion Planning Method of Intelligent Vehicle Based on the Behavior Dynamics”, Mechanical Science and Technology for Aerospace Engineering, 34(2), pp. 301-306(2015).
[31] Han, G.N., Fu, W.P. and Wang, W. “The Study of Intelligent Vehicle Navigation Path Based on Behavior Coordination of Particle Swarm”, Computational Intelligence and Neuroscience, pp.1-10(2016).

Jin-kai Yin is a Ph.D. student in mechanical and electronic engineering at XI'AN University of Technology. He holds M.S. degree in Vehicle Engineering from HENAN University of Science and Technology. His research interests are path planning and decision making of intelligent vehicle.

Wei-ping Fu is Professor and Ph.D. Supervisor at XI'AN University of Technology. He holds B.S. and M.S. degrees from SHAANXI Institute of Machinery, and Ph.D. degree in this field from XI'AN JIAOTONG UNIVERSITY. His research interests include: intelligent vehicle and intelligent robot.

Figure 1. Flow chart of the safety navigation method
Figure 2. The growing process of nodes
Figure 3. The behavioral variables of the self-driving car
Figure 4. The coordinate system of the car
Figure 5. The chart of algorithm effect comparison
Figure 6. Obstacle avoidance using the behavioral dynamics method
Figure 7(a)**.** The underground parking lot
Figure 7(b). The layout of the underground parking lot
Figure 7(c). The global and the partial enlarged diagram
Table 1. Comparison of Simulation Experimental Data

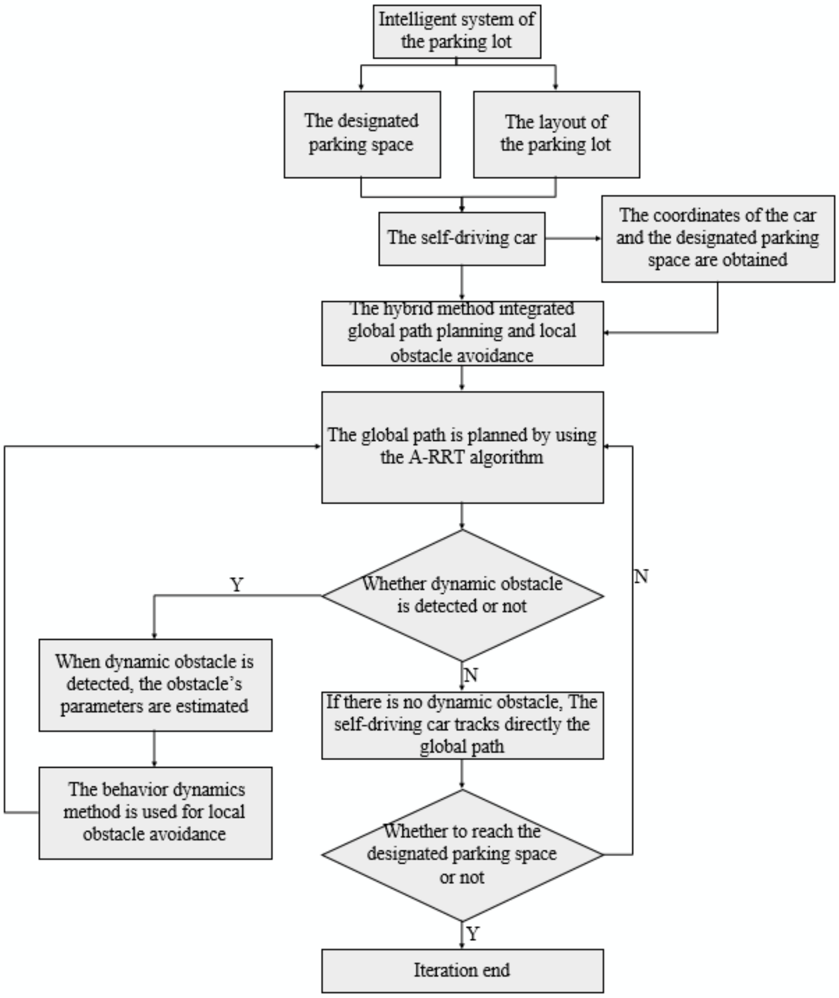


Figure 1. Flow chart of the safety navigation method

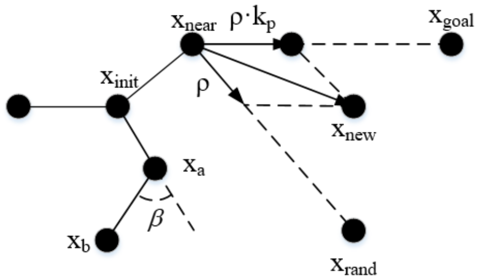


Figure 2. The growing process of nodes

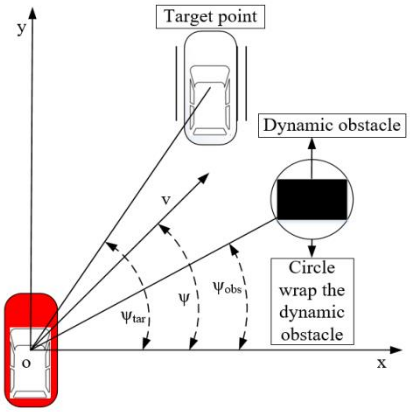


Figure 3. The behavioral variables of the self-driving car

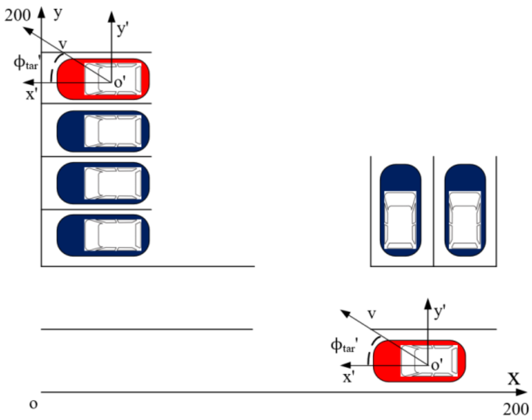


Figure 4. The coordinate system of the car

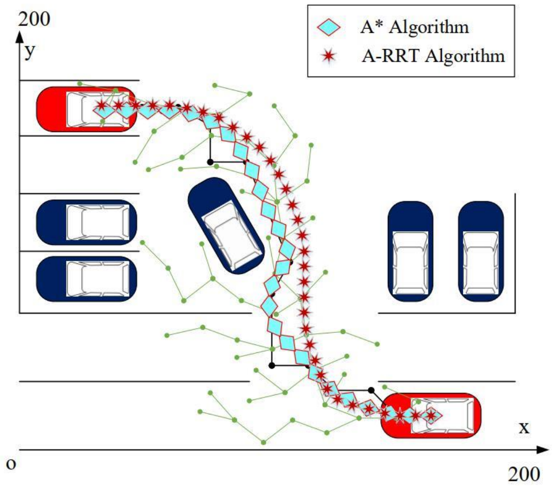
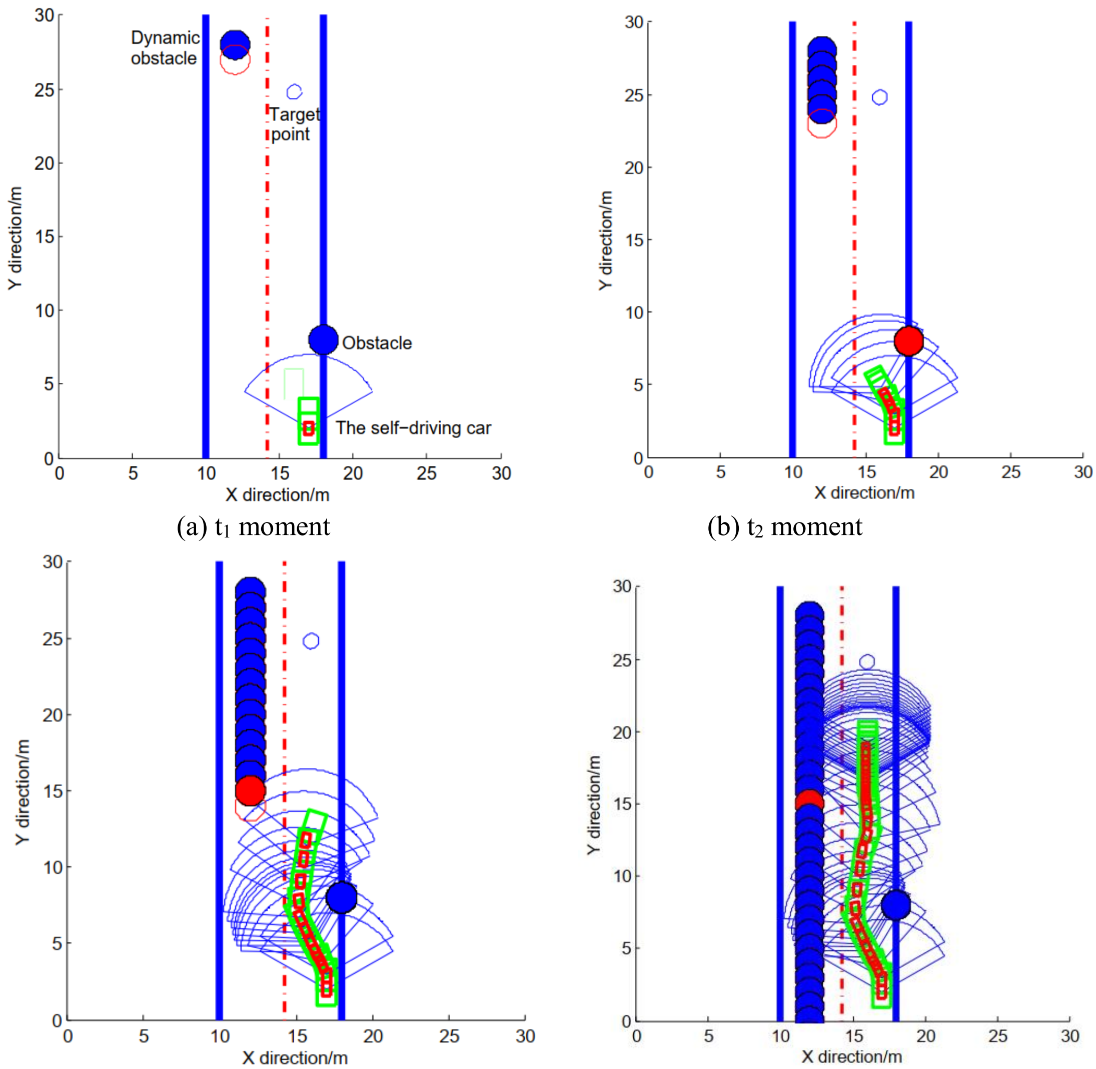


Figure 5. The chart of algorithm effect comparison

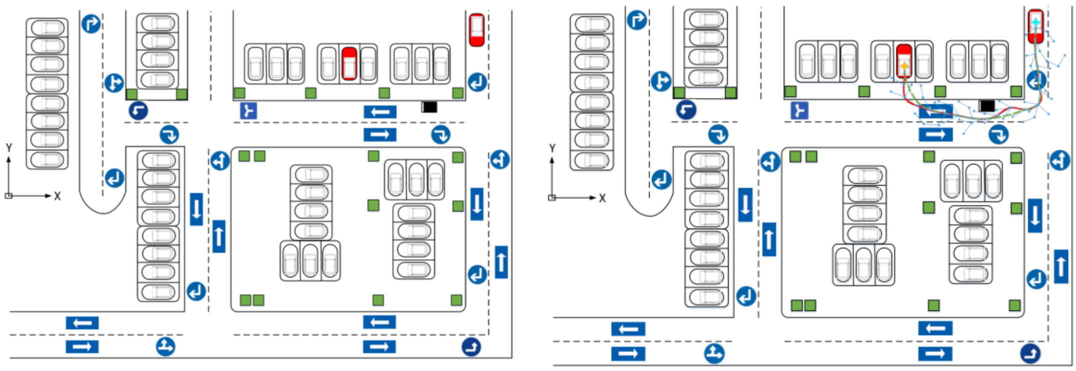


(a) t_1 moment (b) t_2 moment

(c)t_3 moment (d) t_4 moment

Figure 6. Obstacle avoidance using the behavioral dynamics method

(a)The underground parking lot



(b) The layout of the underground parking lot (c) The global and the partial enlarged diagram

Figure 7. The process of safety navigation

Table 1. Comparison of Simulation Experimental Data

30 experiments	A*	A-RRT
Planning Time /s	0.166	0.181
	0.189	0.237
	0.262	0.284
	0.381	0.464
	0.155	0.152
the length/m	130.922	132.423
	131.246	132.68
	129.854	130.262
	130.573	133.025
	130.248	130.883
Successful Times	30	30
Mean Square Deviation of Path Curvature	0.138	0.094