Robotics and Autonomous Systems 132 (2020) 103606

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

A novel path planning methodology for automated valet parking based on directional graph search and geometry curve

Zhaobo Qin [a,*], Xin Chen [a], Manjiang Hu [a], Liang Chen [a], Jingjing Fan [b]

[a] State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha, 410082, China
[b] Intelligent Transportation Key Laboratory, North China University of Technology, Beijing, 1000144, China

ARTICLE INFO

Article history:
Received 6 March 2020
Received in revised form 8 April 2020
Accepted 15 July 2020
Available online 28 July 2020

Keywords:
AVP
Path planning
Path coordination
Directional graph search
Geometry curve

ABSTRACT

The paper presents a novel path planning methodology based on the directional graph search and the geometry curve for the Automated Valet Parking (AVP) system. The whole path planning methodology is divided into three parts including the global path planning, the path coordination strategy and the parking path planning. Firstly, the global path planning is triggered to find a path from the parking slot entrance to the rough location of the assigned parking spot. A novel directional Hybrid A* algorithm is proposed to generate the global path efficiently without redundant searches, such as the dead end. Afterwards, the path coordination strategy gives a transitional path to connect the end node of the global path to the parking planning start node. The transitional path is composed of geometry curves including arcs and line segments based on the optimal parking start node. Finally, the parking path planning generates a parking path to guide the vehicle from parking start node to the parking space. A modified C-type vertical parking path planning algorithm is utilized to generate the parking path, offering flexibility for choosing the parking start node. Simulation results based on Matlab and PreScan show that it takes less time for the proposed path planning algorithm to generate a feasible path for the AVP system compared with the general planning algorithm. The novel AVP path planning algorithm also has the potential for practical use.

1. Introduction

The Automated Valet Parking (AVP) is one of the most relevant and straightforward applications of the autonomous vehicles. It will potentially improve the parking efficiency and customers' driving experience. For AVP, path planning is the main challenge. The path planning platform for autonomous vehicles is built on the first algorithm in path planning for robotics [1], and many studies have been developed based on the A* algorithm. To reduce the search time, the heuristic function is modified depending on the obstacle density [2]. To improve the optimality of the path formed by the grid edges, the Theta* algorithm transports information along the grid edges without constraining the paths to the grid edges [3]. However, a disadvantage of the Theta* algorithm is that lots of invalid node checks waste much time. To avoid the unnecessary check, the Lazy theta* algorithm can only select one node to check in an expansion cycle [4]. Placing weight in the heuristic function is employed in the WA* algorithm which generates less expanded nodes [5]. Although the WA* algorithm is more time-saving than the A* algorithm, the generated path is not guaranteed to be optimal. To improve the optimality of the planned path, the AWA* algorithm adds a procedure on the basis of the WA* algorithm: after the path is generated, the nodes whose heuristic function value is less than the length of the path are expanded to generate a better path [6]. Three new techniques using weights to accelerate heuristic search of grid map are presented. The first technique is based on the iteration of a heuristic search algorithm. The second technique is based on the length between the start node and goal node. The last technique is based on the travel cost [7]. Jump Point Search (JPS) algorithm can further improve the way of expanding nodes and perform much better than A* in dense obstacle area [8]. The any-angle path finding algorithm generates the optimal path by searching over sets of states represented as intervals [9]. All the aforementioned algorithms are based on a grid map, and thus the generated paths cannot satisfy the vehicle kinematics constraints.

Therefore, many path planning algorithms satisfying kinematics constraints have been proposed. Dubins [10] proposes the shortest path planning approach for vehicles using the minimum turning radius. The method is further improved by Reeds and Shepp [11] who take both forward and backward motions into consideration. The common disadvantage of both algorithms is that the scheme is inapplicable for parking because obstacle-free environments are required. Clothoid curve is an improvement of Reeds and Shepp trajectory [12]. Hafida [13,14] generates a set

* Corresponding author.
E-mail addresses: qzb@hnu.edu.cn (Z. Qin), jjfan@ncut.edu.cn (J. Fan).

https://doi.org/10.1016/j.robot.2020.103606
0921-8890/

of clothoid curves in the center reference frame related to the vehicle and evaluates curves by several criteria.

In order to generate a path that satisfies the kinematics and collision-free constraints, the Hybrid A* algorithm is introduced in DARPA Urban Challenge in 2007 [15,16]. Although the Hybrid A* algorithm can provide non-holonomic path from the start to the target position, the path is not ensured to be optimal and may waste much time on the dead end. A hybrid path planner which improves the smoothness of the generated path has been developed applying the Adaptive Hybrid A* and A* [17]. Improving the heuristic function is also a method to realize a less time consuming graph search in consideration of the vehicle motion model [18]. However, all the aforementioned improvements on the Hybrid A* algorithm have a common problem: considerable time may be spent on the dead end. In addition to graph search algorithms, geometry curve is another research focus, especially in the parking path planning. By using the straight line segments and circular arcs, Hou proposes three algorithms including CC (Circle–Circle) type algorithm, CSC (Circle–Straight Line–Circle) type algorithm and iterative parking algorithm [19]. As for vertical parking, Sun introduces two algorithms including C-type parking algorithm and Y-type parking algorithm [20]. In order to generate a smooth path, a new curve element named linearly steering spiral is used for autonomous parking system [21]. Vorobieva applies the clothoid curve to parallel parking [22–24]. Optimization-based approaches are also presented. A novel hierarchical controller that combines the Hybrid A* and optimization-based collision avoidance is proposed to generate smooth, collision-free, and feasible dynamic path effectively [25]. A parking path planning method is presented combining rapidly-exploring random tree (RRT) with non-holonomic constraints and kinematics model vehicle. Although the generated path is feasible, it requires large space while the parking process [26]. It is worth noting that the aforementioned methods rarely focus on the range of the parking start region which is the important information to guarantee parking successfully.

Since the DARPA City Challenge was held in 2007, a series of autonomous parking vehicles have gradually appeared. For example, the European V-CHARGE project coordinates three modules for motion planning, including the global planning, local planning and parking planning. The global planning module computes topological plans through the graph-search. The local planning module uses a fast local motion planner to compute motion commands. The parking planner uses a three-stage planning cascade with increasing complexity [27]. Although the generated path contains only a minimal number of cusps and uses little space, the curvature is discontinuous. The AutoPLES project extends the existing algorithms RRT-Connect and RRT*, resulting in a new planning approach that outperformed both [28]. However, the parking path requires a slow velocity to satisfy high precision. For AVP system, efficiency is a significant issue. Sedighi combines the Hybrid A* and visibility diagram planning to find the shortest possible non-holonomic path and avoid redundant searches [29]. However, the method is difficult to deal with collision-free constraints for a narrow parking space. Wang proposes a path coordination strategy to merge the global path and the parking path. Nevertheless, the parking start node of each parking space is fixed, which limits the flexibility of the automated parking process [30].

This paper therefore presents a novel path planning algorithm for the AVP system which is composed of three parts: global path planning, path coordination strategy and parking path planning. The proposed path planning algorithm consumes less time by avoiding invalid searches. It can also coordinate global path and parking path smoothly and generate the parking path with more flexibility. The main contributions of the paper are as follows:

(1) A novel path planning methodology is proposed for the AVP system with adequate scene adaptation and safety guarantee. The path planning algorithm applies a novel path coordination strategy to merge the global path and the parking path. The strategy generates the new node in the same way with the Hybrid A*, however, the heuristic values of new nodes are calculated with the information of the optimal parking start node. The strategy can not only make a smooth transition on the path from the global planning to the parking planning, but also guarantee the near-optimal parking start node, which makes the parking process successful.

(2) By combining the JPS algorithm with the Hybrid A* algorithm, the directional Hybrid A* algorithm is presented for the global path planning, which uses the JPS algorithm to provide the reference direction for the Hybrid A* algorithm. The proposed method can generate a similar path compared with the general Hybrid A* algorithm, while consuming much less time.

(3) The optimal parking start node corresponding to any feasible heading angle is calculated and taken as a reference to the target parking start node based on the collision constraints and the optimal function. Compared with the existing parking algorithm, of which the parking start heading angle must be 0, the proposed parking method is more flexible

The paper is organized as follows. Section 2 introduces the novel path planning methodology and the basic kinematics model. Section 3 demonstrates the directional Hybrid A* algorithm which combines the JPS algorithm with the Hybrid A* algorithm. Section 4 describes the path coordination strategy. Section 5 shows the algorithm of parking path planning and the process of calculating the optimal parking start node. In Section 6, simulation results of the proposed path planning algorithm are introduced. Finally, conclusions are drawn in Section 7.

2. The path planning methodology

The paper takes the scenario of vertical parking slot as an example. The proposed path planning methodology for the AVP system is shown in Fig. 1, consisting of global path planning, path coordination strategy and parking path planning.

When the parking command is received, the global parking path planning firstly needs to find a path following the JPS algorithm. The areas around the nodes in the path are treated as target areas for the Hybrid A*. According to the guide of direction, the Hybrid A* can generate paths to the target areas orderly until arriving at the last target area. Path coordination is then started. Receiving the end node of the global path and the optimal parking start node, the path coordination strategy uses the geometry curves to generate a transitional path. Furthermore, the end node of the transitional path is adopted as the parking start node. In terms of the information of the parking start node, parking path planning can generate the specific parking path by C-type algorithm. The final path for AVP is obtained by combining the global path, the transitional path and the parking path.

The parking process in the paper is conducted based on Ackerman steering geometry [31], whose concept is to have all four wheels rolling around a common node during a turn without slipping. The common node is known as the instantaneous center of curvature in Fig. 2.

$$\frac{W_x}{L_x} = \cot t_1 - \cot t_2 \tag{1}$$

where W_x is the length of the wheel track. L_x is the length of the wheelbase. t_1 is the steering angle of the inner wheel. t_2 is the steering angle of the outer wheel.

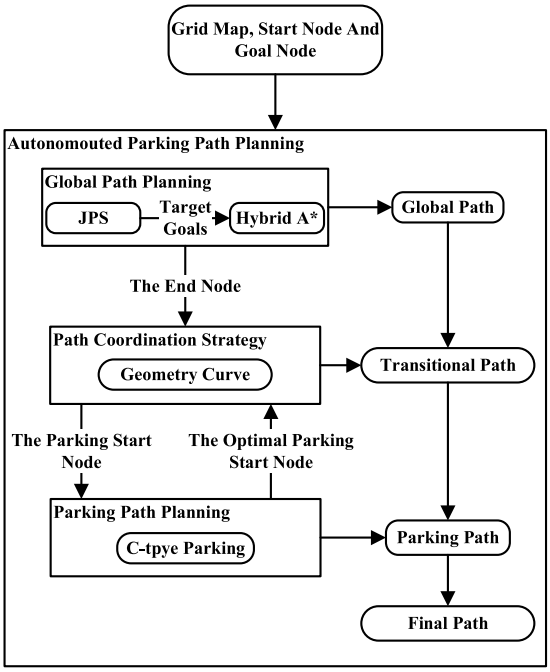


Fig. 1. Process of the path planning methodology.

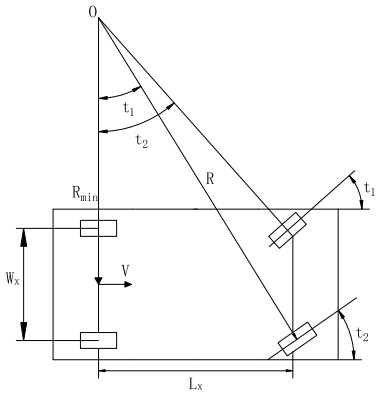


Fig 2. Ackerman steering geometry

$$\frac{W_x}{L_x} = \cot t_1 - \cot t_2$$

Fig. 2. Ackerman steering geometry.

Ackerman steering geometry is the most basic principle and the basis of calculation and derivation while studying the automated parking system in the paper. Therefore, the vehicle mentioned in the paper is supposed to make a circular motion at a certain turning angle while steering.

Based on the Ackerman steering geometry, the direction of velocity V at the center node of the rear axle is parallel to the car body. In the parking process, it is considered that the motion state of the rear axle center can indicate the motion state of the vehicle, and thus the paper takes the center of the rear axle as the path planning reference node.

3. The global path planning

3.1. Jump point search (JPS)

JPS can be implemented as an optimization of the A* algorithm with a major change on the method of generating the successors.

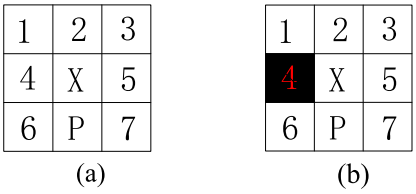


Fig. 3. Illustration of forced neighbor.

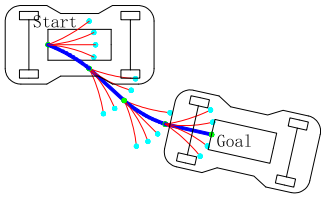


Fig. 4. Expanding method of Hybrid A* algorithm.

According to the pruning and jumping rules, JPS can avoid a lot of useless calculations on invalid nodes [32].

Pruning Rules: Given a node X, reached by a parent node P, the neighbors of X will be pruned and the successors will be selected following the rules:

1. Assuming that there are no obstacles around X, for a neighbor node n_2, there exists a path $\pi' = (p, n_1, n_2)$ that is shorter than or equal to the path $\pi = (p, x, n_2)$.
2. In the case of the obstacle, the path $\pi' = (p, n_1, n_2)$ is not valid.

If there exists a node n_2 which satisfies the two rules above, n_2 is referred as a successor of X.

The two rules is further illustrated in Fig. 3(a) and (b). In Fig. 3(a), no obstacles exist and the path $\pi' = (p, 4, 1)$ is equal to path $\pi = (p, x, 1)$. But when the obstacle exists as in Fig. 3(b), the path $\pi' = (p, 4, 1)$ is not valid. So the node 1 is referred as a successor of X. According to the explanation, only when the obstacles exist in a node's neighborhood can the node have a successor.

Jumping Rules: JPS expands neighbor from eight directions. Expansion in each direction will stop once the jumping node is found. Details are given in [8].

JPS can find the target node and generate corresponding path by the jumping nodes. JPS inserts less node in the open list compared with A*, which is doubly beneficial as (i) the number of operations is reduced. (ii) the number of nodes in each list is reduced, making each list operation simpler.

3.2. Hybrid A*

The obvious difference between the Hybrid A* and A* is the mode of generating adjacent nodes. Hybrid A* expands nodes by applying several steering actions (max-left, no-turn, max-right) to the continuous state associated with the node. New adjacent nodes are generated through the kinematic model of the vehicle. Fig. 4 illustrates how Hybrid A* algorithm expands nodes by the steering actions of the vehicle.

The old node $N_{i-1}(x_{i-1}, y_{i-1}, \theta_{i-1})$ (x is the abscissa, y is the ordinate and θ is the heading angle of the vehicle) generates the new node $N_i(x_i, y_i, \theta_i)$ [33] as in Eqs. (2) to (6):

$$\beta = \left(\frac{L_{arc}}{W_x}\right) \times \tan\alpha \tag{2}$$

$$R_i = \frac{L_{arc}}{\beta} \tag{3}$$

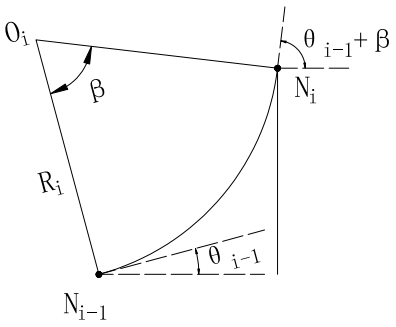


Fig. 5. Generate the new node by Hybrid A*

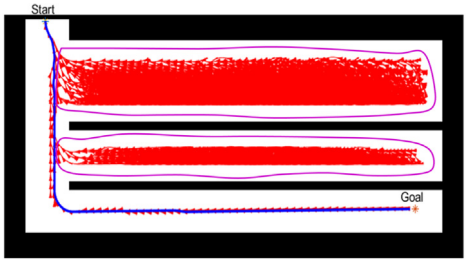


Fig. 6. Redundant expanding of the Hybrid A*.

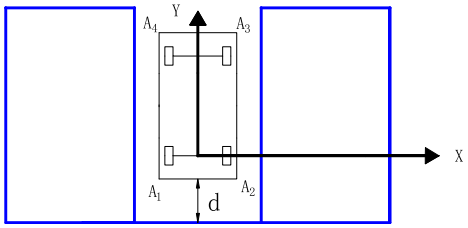


Fig. 7. The end position of vertical parking and the coordinate system.

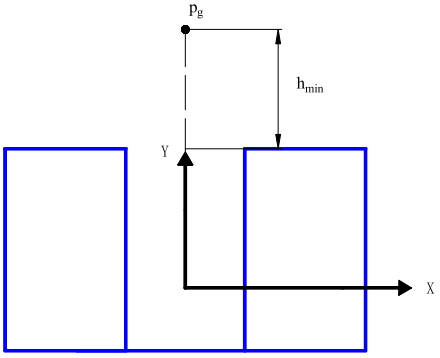


Fig. 8. Global target position.

$$x_i = x_{i+1} + R_i\left(\sin\left(\theta_{i-1} + \beta\right) - \sin\left(\theta_{i-1}\right)\right) \tag{4}$$

$$y_i = y_{i+1} + R_i\left(\cos\left(\theta_{i-1}\right) - \cos\left(\theta_{i-1} + \beta\right)\right) \tag{5}$$

$$\theta_i = \theta_{i-1} + \beta \tag{6}$$

where W_x is the wheelbase. L_{arc} is the desired travel length. α is the steering angle of the vehicle.

3.3. Merging hybrid A* with JPS

The most significant advantages of JPS are less complexity of operation and less time consumption during the search. In the parking slot, the obstacles are usually dense, which is suitable for JPS to implement. However, as the JPS ignores the non-holonomic characteristics of the vehicle, the generated path is infeasible for vehicles to track, which is a non-negligible disadvantage of JPS.

As a significant feature of the Hybrid A*, it takes the non-holonomic characteristics of the vehicle into account. But the Hybrid A* algorithm sometimes wastes much time in the dead-end, as shown in Fig. 6. The invalid searches are marked with pink curve. In the parking slot, intersections always exist, which means that the traditional Hybrid A* algorithm is not applicable for quick search.

In order to generate a feasible global path for AVP efficiently, the directional Hybrid A* path planning algorithm is proposed. Firstly, the JPS is implemented on the given grid map to obtain the referenced path to the target node. The path will provide the correct direction for the Hybrid A* to generate a feasible path considering the non-holonomic constraints of the vehicle. The steps are explained in detail as follows:

Input: Grid map of the parking slot (M), start position ($Start$), target parking space $\left(p_{ps}\right)$, steering actions of the vehicle (V_s) and the location of obstacles (O_m).

***Output*:** The global path $\left(Path_{global}\right)$ for an autonomous vehicle.

The steps to generate the global path are explained:

Step one: The grid map of the parking slot is provided by locating the start position, the target parking space and the obstacle position on that $\left(M\left(Start, p_{ps}, O_m\right)\right)$.

Step two: Define the coordinate system. For vertical parking, the target parking position is defined as in Fig. 7. The distance between the rear car body and the rear obstacle is d (reserved safe distance) and the left body is parallel to the outer boundary of the parking space. The coordinate system is established at the center of the vehicle rear axle and the parking target point is the origin of the coordinate system. The advantage of the coordinate system is that the abscissa and ordinate of the start position are exactly the lateral displacement and longitudinal displacement in the parking process, which is easy for calculating.

Step three: Define the target global position (Goal). The requirements of the global path planning are that the vehicle should not only be close to the parking space, but also adjust the heading angle while finishing the global path planning. Therefore, the abscissa of the target global path planning position is 0 and the distance from the ordinate to the parking space is $h_{\min}$ (The calculation of $h_{\min}$ is shown in Section 6) as shown in Fig. 8.

Step four: Implement JPS to find a path $\left(Path_{JPS}\right)$ from the start position to the desired target global position (Goal) as shown in Fig. 9. The nodes $(p_1, p_2, p_3 \ldots p_n)$ in the path in Fig. 10 are treated as the reference direction for further use.

Step five: Generate the target areas $(Area_1, Area_2, Area_3 \ldots Area_n)$. The nodes $(p_1, p_2, p_3 \ldots p_n)$ are regarded as the local target goals for the Hybrid A*. For the reason that the Hybrid A* can only find a path to the target area but not a specific target node, the target areas $(Area_1, Area_2, Area_3 \ldots Area_n)$ are defined by radius R_l in Fig. 11 (see Fig. 12).

As the nodes $(p_1, p_2, p_3 \ldots p_n)$ may be close to the obstacle, the target areas need to be large enough to avoid collision. The value of R_l can be calculated in Eq. (7):

$$R_l = \sqrt{\left(L_x + L_f\right)^2 + \left(\frac{W_c}{2}\right)^2} \tag{7}$$

where L_f is the length of the front overhang. L_x is the length of the wheelbase. W_c is the vehicle width.

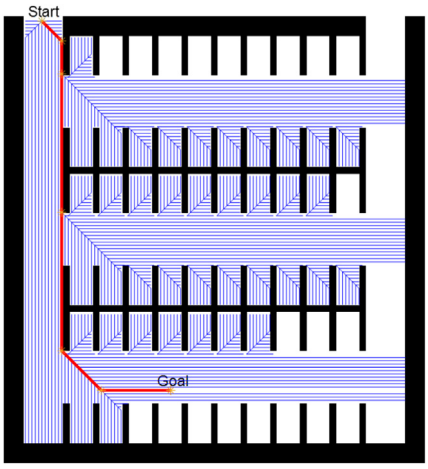


Fig. 9. Path of JPS.

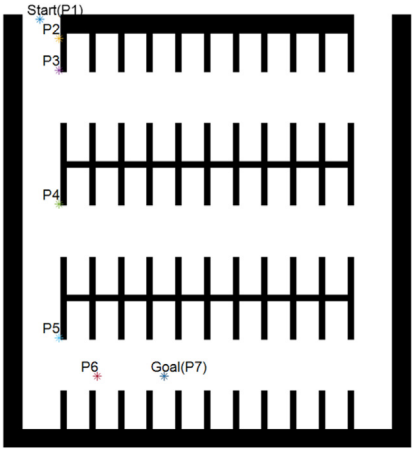


Fig. 10. Reference of node direction.

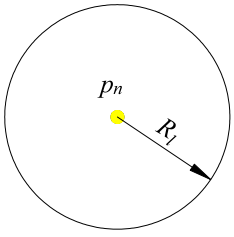


Fig. 11. Target area $Area_n$.

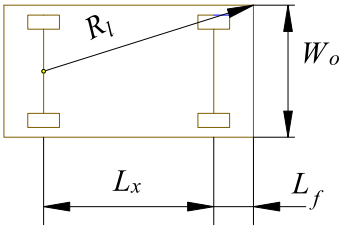


Fig. 12. The sketch of R_l.

Step six: Generate the path to the target areas successively. In the beginning, $Area_2$ is adopted as the target area. Once the current expanding node arrives at $Area_2$, all the nodes in the open list are removed and the target area is changed into $Area_3$. The above procedure is repeated until the current expanding node arrives at the final target area $Area_n$.

After the process of traceback, the final global planning path $Path_{global}$ is generated. The significant advantage of the proposed global path planning is that it takes the non-holonomic characteristics of the vehicle into account and avoids redundant searches especially in the dead-end.

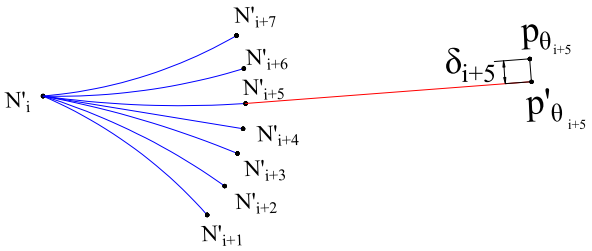


Fig. 13. Illustration of the defined error value.

4. The path coordination strategy

The global path planning can generate a path from the start node to the global target area which is close to the parking space. However, the end node of the global path $p_{\text{end}}(x_{end}, y_{end}, \theta_{end})$ may not be a suitable parking start node. In order to connect the global path planning to the parking path planning, the path coordination strategy needs to be implemented after the global path planning to generate a transitional path $Path_{tran}$.

Each feasible heading angle θ_i corresponds to an optimal parking start node p_{θ_i} (the range of θ_i and the calculation of p_{θ_i} are further illustrated in Section 6). The end node of the global path $p_{\text{end}}(x_{end}, y_{end}, \theta_{end})$ may be distant from its corresponding optimal parking start node $p_{\theta_{end}}$, and therefore the node p_{end} needs to be expanded in the same method of the Hybrid A* to generate new nodes.

As in Fig. 4, each expansion generates five new nodes in the global path planning. The number of expanded nodes is increased to seven in the path coordination strategy, as in Fig. 13. The advantage of large quantity of expanded nodes is to speed up the process of finding an appropriate transitional path.

The steps to create the transitional path are explained:

Step one: Preset a threshold value E.

Step two: Expand the current node in the same way with the Hybrid A* to generate the new nodes $\left(N_1', N_2' \ldots N_i'\right)$.

Step three: Put all the new nodes into the open list. The information of each node N_i' $(x_i, y_i, \theta_i, \delta_i)$ includes the abscissa value x_i, the ordinate value y_i, the heading angle θ_i and the heading angle error value δ_i. The error value δ_i is explained as follow:

If the heading angle θ_i is not feasible, the error value δ_i will be infinity; otherwise, N_i' is the current node as in Fig. 13. $N_{i+1}{}', N_{i+2}{}' \ldots N_{i+7}{}'$ are the new nodes generated by N_i'. Take $N_{i+5}{}'\left(x_{i+5}{}', y_{i+5}{}', \theta_{i+5}{}'\right)$ as an example to illustrate the error value δ. The node $p_{\theta_{i+5}}$ is the optimal node corresponding to $\theta_{i+5}{}'$. The red line is tangent to the arc $N_i'N_{i+5}{}'$ and the node $p_{\theta_{i+5}}{}'$ is the closest node to the node $p_{\theta_{i+5}}$ in the line. The distance between $p_{\theta_{i+5}}$ and $p_{\theta_{i+5}}{}'$ is δ_{i+5}.

The physical significance of the value δ is the quality level of the node. If a node's value of δ is low enough, the node is able to be expanded to generate the transitional path.

Step four: Check each new node to find out whether there exists a node whose value of δ is lower than E: if not, select the node with the minimum value of δ to be the current node and

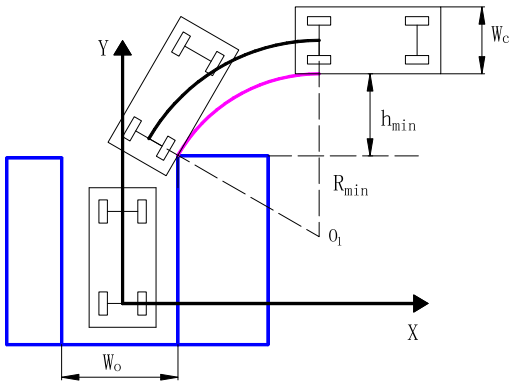


Fig. 14. The sketch of $h_{\min}$.

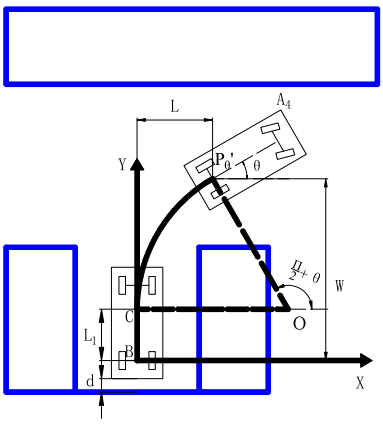


Fig. 15. The sketch of the C-type algorithm.

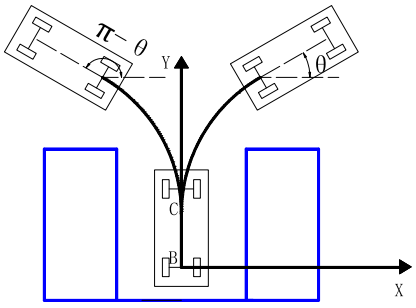


Fig. 16. The symmetry parking process of both sides.

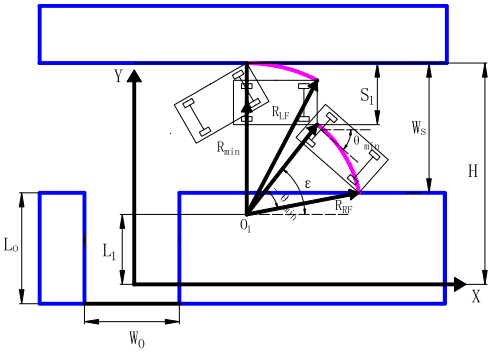


Fig. 17. The sketch of the minimum parking start heading angle.

return to *Step two*; if so, select the node with the minimum value of δ and expand by the line segment until the line arrives at the closest node p_θ' to the optimal parking start node p_θ, as shown in Fig. 13. The node $p_\theta'\left(x_\theta', y_\theta', \theta\right)$ is finally treated as the parking start node.

The transitional path from the global path to the parking path can be generated. The obvious advantage of the coordination strategy is that the selection of the parking start node is flexible and the parking start node is guaranteed to be near-optimal.

5. The parking path planning

After implementing the path coordination strategy, the transitional path and the parking start node are obtained. According to the parameters of the parking space and the vehicle contour, the parking path can be generated for the assigned parking spot. The paper takes vertical parking scenario as an example, and the vehicles can travel in both directions on the road.

The paper applies C-type parking algorithm to generate the parking path. However, the parking start node is always fixed for each parking space and the parking start heading angle must be zero in most research [20]. In addition, the distance between the parking start node and the obstacle must be bigger than $h_{\min}$ as shown in Fig. 14. Thus C-type parking algorithm is modified in the paper in order to make the selection of the parking start node more flexible. For each parking start heading angle, the optimal parking start node can be calculated utilizing the proposed method based on collision-free constraints.$h_{\min}$ is regarded as a parameter of the target global node, as mentioned in Section 4.

$$h_{\min} = R_{\min} - \sqrt{\left(R_{\min} - \frac{W_c}{2}\right)^2 - \left(R_{\min} - \frac{W}{2}\right)^2} \tag{8}$$

where $R_{\min}$ is the minimum turning radius.

The process of the modified C-type parking is shown as follow:

The parking start node is p_θ' and the parking target node is B as in Fig. 15. The steering wheel turns to a certain fixed angle and begins to parking. After the vehicle travels in a section of circular arc $p_\theta'C$, the steering wheel returns to zero and the vehicle travels a straight line CB for a distance until reaching the target parking node B.

The mathematical models are shown in Eqs. (9) and (10):

$$L = R\left(1 - \sin\theta\right) \tag{9}$$

$$W = R\cos\theta + L_1 \tag{10}$$

where R is the radius of the circular arc.

According to the known values of L, W and θ which are provided by the parking start node $p_\theta'\left(x_\theta', y_\theta', \theta\right)$, the parking parameter R can be calculated by Eqs. (9) and (10).

By C-type parking algorithm, the parking path can be generated. In order to combine the transitional path and the parking path effectively, the methods of calculating the range of the parking start heading angle θ_i and the optimal parking start node p_{θ_i} are illustrated as mentioned in Section 5. As the parking process of both sides are symmetric in Fig. 16, only right parking is discussed in the paper.

5.1. The range of the parking start heading angle

There is no doubt that the maximum value of the parking start heading angle is $\frac{\pi}{2}$. The minimum value of the parking start heading angle $\theta_{\min}$ is shown in Fig. 17.

$$W_s - S_1 = R_{RF}\left(\sin\varepsilon - \sin\left(\varepsilon + \theta_{\min}\right)\right) \tag{11}$$

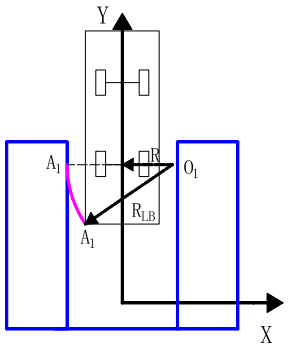


Fig. 18. Rear-left collision in vertical parking.

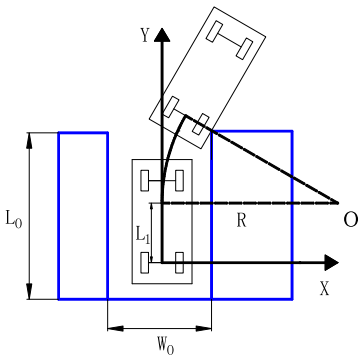


Fig. 19. Right side collision in vertical parking.

$$R_{RF} = \sqrt{\left(R_{\min} - \frac{W_c}{2}\right)^2 + \left(L_x + L_f\right)^2} \tag{12}$$

$$S_1 = \sqrt{\left(R_{\min} + \frac{W_c}{2}\right)^2 + \left(L_x + L_f\right)^2} - R_{\min} + \frac{W_c}{2} \tag{13}$$

$$\varepsilon = \frac{\pi}{2} - \arctan\left(\frac{L_x + L_f}{R_{\min} - \frac{W_c}{2}}\right) \tag{14}$$

where H is the distance between the left boundary and the end node. W_c is the width of the parking space. W_s is the width of the parking aisle.

5.2. The optimal parking start node

The precondition for calculating the optimal parking start node is getting the range of the parking start node for each heading angle. As shown in Eqs. (9) and (10), if the range of R and L_1 are defined, then the range of L and W could be defined with the given value of θ, which means that the range of the parking start node for θ is obtained. Therefore, the constraints of defining the range of R and L_1 need be proposed.

The geometry parameters R and L_1 of the path are calculable if the vehicle can follow the parking path without collision. It is important to define the range of the geometry parameters R and L_1 under collision-free constraints. The collision-free constraints are analyzed in the reverse parking process.

(1) While pulling out the parking space, the rear-left side of the vehicle A_1 may collide with an obstacle as shown in Fig. 18. The constraint in Eq. (15) must be satisfied.

To avoid the kind of collision, the following constraint must be satisfied.

$$\sqrt{\left(\left(R + \frac{W_c}{2}\right)^2 + L_b^2\right)} - R \le \frac{W_o}{2} \tag{15}$$

where L_b is the length of the rear suspension of the vehicle. W_o is the width of the parking space.

(2) While pulling out the parking space, the rear-right side of the vehicle may also collide with an obstacle as shown in Fig. 19. The constraint in Eq. (16) must be satisfied.

To avoid the kind of collision, the following constraint must be satisfied.

$$\sqrt{\left(\left(R + \frac{W_c}{2}\right)^2 + L_b^2\right)} - R \le \frac{W_o}{2} \tag{16}$$

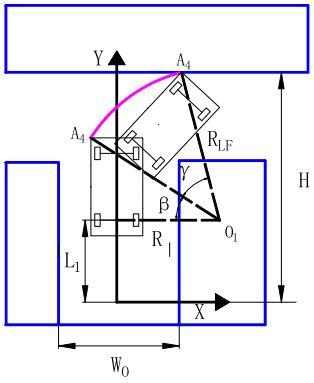


Fig. 20. Front-right collision in C-type parking.

(3) While pulling out the parking space, the front-left side of the vehicle A_4 may collide with an obstacle, as shown in Fig. 20. The constraints in Eqs. (17)–(19) must be satisfied.

To avoid the kind of collision, the following constraints must be satisfied.

$$A_{4_{y\max}} \le H \tag{17}$$

$$A_{4_{y\max}} = \max(L_1 + R_{LF}\sin(\pi - \beta - \gamma), \gamma \in (0, \frac{\pi}{2} - \theta)) \tag{18}$$

$$R_{LF} = \sqrt{\left(R + \frac{W_c}{2}\right)^2 + \left(L_x + L_f\right)^2} \tag{19}$$

where H is the distance between the left-side obstacle and the target parking node.

(4) While pulling out the parking space, the front-right side of the vehicle A_3 may collide with an obstacle as shown in Fig. 21. The constraints in Eqs. (20) and (21) must be satisfied.

To avoid the kind of collision, the following constraints must be satisfied.

$$L_1 + R\cos\theta - \sqrt{\left(\frac{W_c}{2}\right)^2 + \left(L_x + L_f\right)^2}\sin(\theta + \theta_1) \ge H_1 \tag{20}$$

$$\theta_1 = \arctan(\frac{W_c}{2(L_x + L_f)}) \tag{21}$$

where H_1 is the distance between the right-side obstacle and the parking target node.

For the whole parking process, four possible collisions shown above need to be analyzed. According to Eqs. (15)–(21), the constraints of geometry parameters of C-type parking algorithm can

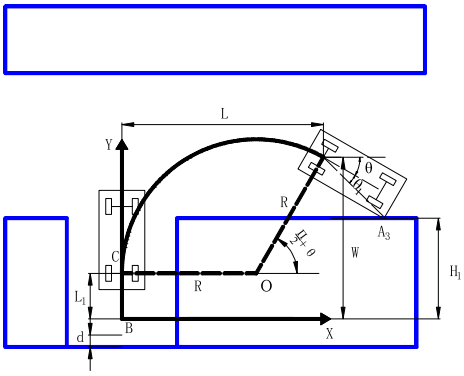


Fig. 21. Front-right side collision in vertical parking.

Table 1
Simulation parameters of the vehicle and the parking slot.

Length of vehicle	5.21 m
Width of vehicle	2.04 m
Wheelbase	2.94 m
Front overhang	0.99 m
Rear overhang	1.28 m
Minimum turning radius	5.09 m
Maximum steering velocity	15 °/s
Parking velocity	0.8 m/s
Length of parking slot	68 m
Width of parking slot	64 m
Length of parking space	6 m
Width of parking space	3.5 m

be concluded as follows:

$$\begin{cases} R \geq R_{\min} \\ L_1 \geq 0 \\ \sqrt{\left(\left(R+\frac{W_c}{2}\right)^2 + L_b^2\right)} - R \leq \frac{W_o}{2} \\ \sqrt{\left(R-\frac{W_o}{2}\right)^2 + (L_1-(L_o-d-L_b))^2} - \left(R-\frac{W_c}{2}\right) \leq 0 \\ A_{4_{\text{ymax}}} \leq H \\ L_1 + R\cos\theta - \sqrt{\left(\frac{W_c}{2}\right)^2 + \left(L_x+L_f\right)^2}\sin(\theta+\theta_1) \geq H_1 \end{cases} \tag{22}$$

The range of unknown variables R and L_1 can be calculated through the constraints above. In order to get the optimal parking start node for θ from the calculated range, a cost function is proposed in Eq. (23):

$$f = R + R(\frac{\pi}{2} - \theta) + L_1 \tag{23}$$

The constraints corresponding to the cost function are shown in Eq. (22), and thus the optimal value of R and θ can be calculated. After putting the optimal values into Eqs. (9) and (10), the optimal parking start node p_θ corresponding to heading angle θ can also be obtained.

6. Simulation results

A novel path planning algorithm based on directional graph search and geometry curves for AVP has been explained. To further verify the feasibility of the presented methodology, simulations are carried out based on Matlab/Simulink and PreScan. An electric vehicle model provided by PreScan is used for the simulation. The parameters of referred vehicle and parking lot are shown in Table 1.

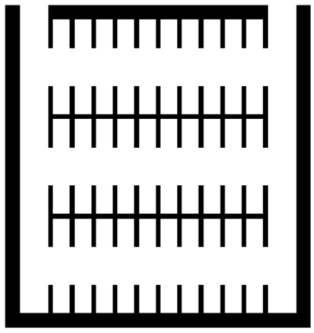


Fig. 22. The parking slot.

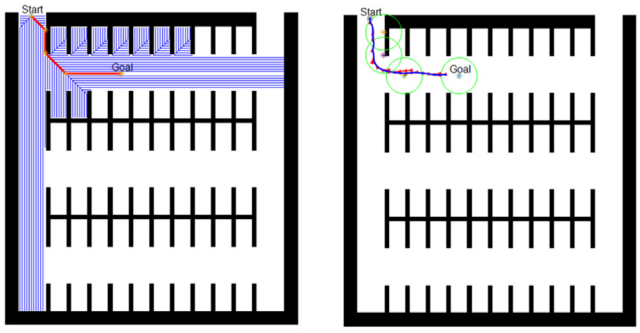


Fig. 23. The process of the directional Hybrid A* algorithm in the first condition.

Table 2
Comparison results.

	Time spent		Number of nodes	
	Directional Hybrid A*	General Hybrid A*	Directional Hybrid A*	General Hybrid A*
The first condition	0.277 s	0.272 s	150	250
The second condition	0.359 s	0.696 s	595	3985
The third condition	0.390 s	0.931 s	1145	5315

The simulation verification mainly focus on three aspects: the comparison between the directional Hybrid A* algorithm and the general Hybrid A* algorithm, the path planning and path tracking results.

6.1. The global path planning

The parking slot in the simulation has 60 vertical parking spaces in Fig. 22. The length of the parking slot is 68 m, and the width of the parking slot is 64 m. The same size of each parking space has the length of 6 m and the width of 5 m respectively.

In order to make a sufficient comparison between the directional algorithm and the general Hybrid A* algorithm, the results under three conditions are presented. The performance is shown as follows: Figs. 23(a), 25(a), and 27(a) display the search process of JPS under each condition. Figs. 23(b), 25(b), and 27(b) show the search process with the guided direction based on JPS. Figs. 24, 26 and 28 display the search process of the general Hybrid A* under each condition. The results of consuming time and the number of expanded nodes are compared in Table 2 using a workstation with Intel@2.30 GHz.

From the comparison results in Figs. 23, 28 and Table 1, the modified Hybrid A* algorithm can always perform better than the general Hybrid A* algorithm in the parking slot. The difference between two algorithms is small in the first condition, because

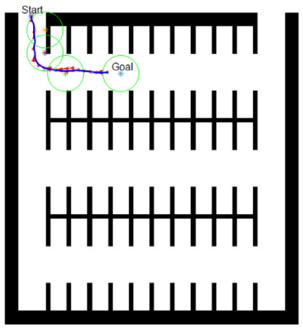


Fig. 24. The process of the general Hybrid A* in the first condition.

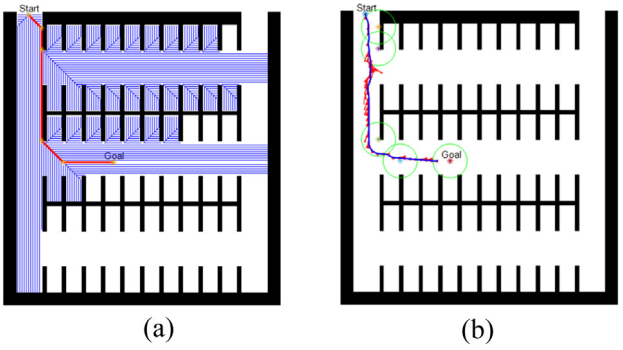


Fig. 25. The process of the directional Hybrid A* in the second condition.

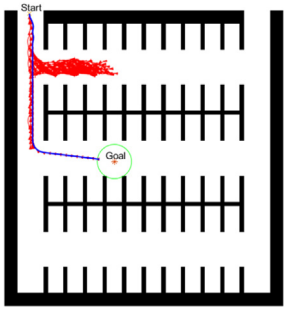


Fig. 26. The process of the general Hybrid A* in the second condition.

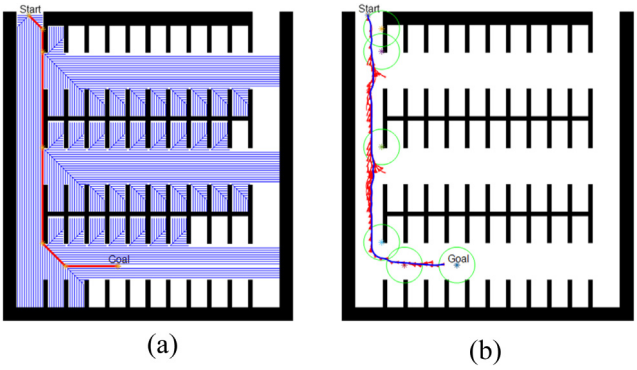


Fig. 27. The process of the directional Hybrid A* algorithm in the third condition.

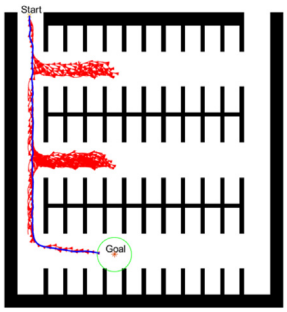


Fig. 28. The process of the general Hybrid A* in the third condition.

the parking space is close to the start position. In the second and third conditions, the novel algorithm has a much better performance than the general Hybrid A* algorithm. Therefore, the proposed modified Hybrid A* algorithm is more efficient than the general Hybrid A* algorithm in the parking slot which exists intersections. The advantage will be more obvious in a more complex parking slot.

6.2. The path planning and path tracking results

In order to verify the feasibility of the whole path planning methodology, three representative parking spaces are selected to generate the corresponding paths accordingly.

In Figs. 29, 30 and 31, Start1 represents the parking slot entrance. Goal1 represents the global target node calculated as in Fig. 5 and Equation (8). The green area is the global target area, which is a circle corresponding to the Goal1. Start2 represents the parking start node and the target parking node is Goal2.

Figs. 29(a), 30(a) and 31(a) display the planned path from the parking slot entrance to each assigned parking space.

Figs. 29(b), 30(b) and 31(b) show the details of the transitional path. The path coordination strategy is implemented after the planned path arriving at the global target area. New nodes are generated and a feasible node is selected to be expanded by the line segment. The smooth and simple transitional path is generated after the line arrives at Start2.

Figs. 29(c), 30(c) and 31(c) display the details of the planned parking path.

The planned path for each parking space demonstrates that the global path, the transitional path and the parking path are connected successfully and smoothly. In order to further test whether the paths are collision-free, the path in Fig. 30 is selected as an example for joint validation together with path following control.

A path following control module is added with the MPC controller, which is not the focus of the paper. Comparison between the planned path and the tracked path is shown in Fig. 32.

The vehicle occupancy area during the whole AVP process is shown in Fig. 33.

The result of path planning and path following co-simulation further demonstrates that the proposed path planning methodology can generate a feasible path to guide the vehicle from the parking slot entrance to the desired parking space.

7. Conclusions

The paper presents a path planning methodology based on the directional graph search and geometry curves, which merges the global path, the transitional path and the parking path planning

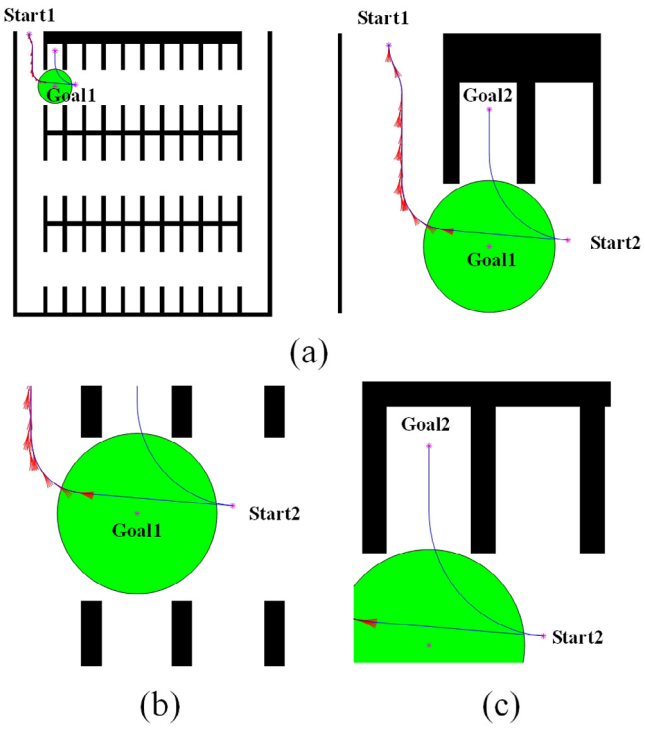


Fig. 29. The planned path for the first parking spaces.

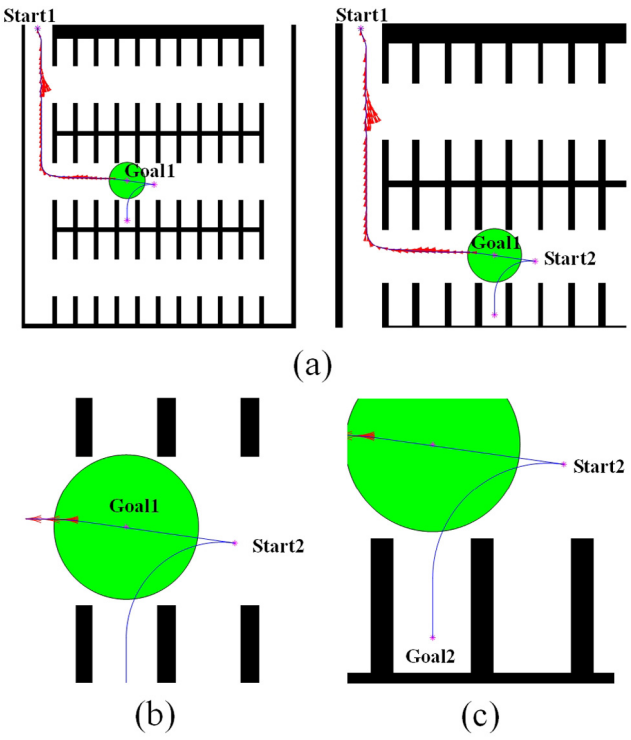


Fig. 31. The planned path for the third parking space.

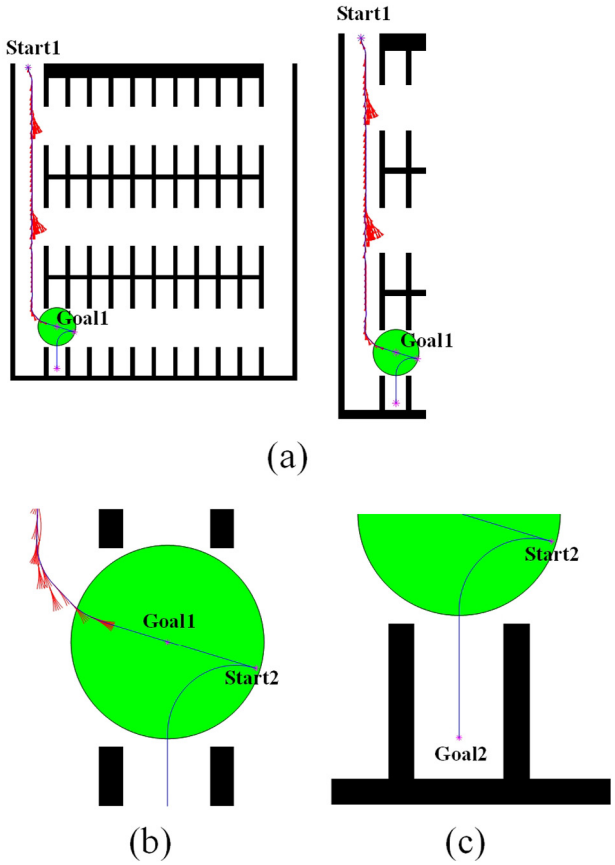


Fig. 30. The planned path for the second parking space.

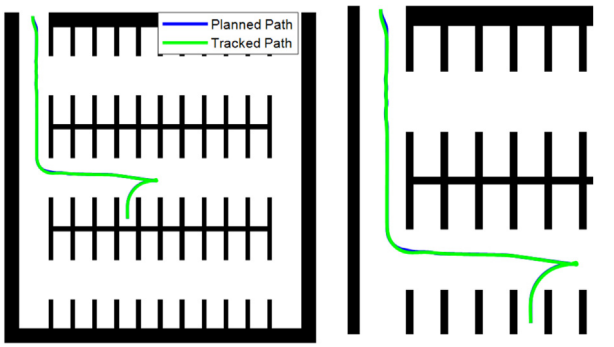


Fig. 32. Comparison between planned path and tracked path.

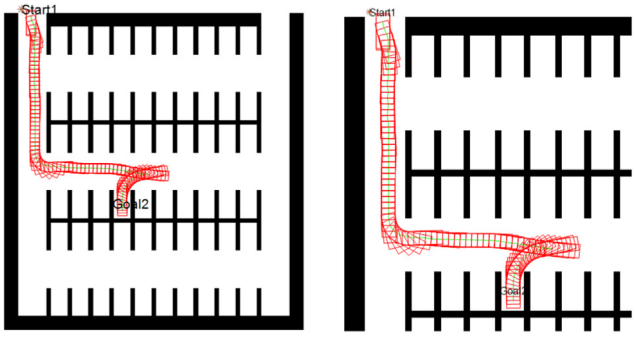


Fig. 33. The vehicle occupancy area during AVP.

for AVP. Experimental results show that the proposed methodology can generate a feasible path for the AVP directly and effectively which has the potential for practical use. Main conclusions can be given:

(1) A novel global path planning algorithm, the directional Hybrid A* algorithm, which integrates the JPS and Hybrid A* together, is proposed. Simulation results have shown that it performs much better than the general Hybrid A*, with less time consumption and expanded nodes.

(2) The presented path coordination strategy can not only switch from the global path to the parking path smoothly, but also guarantee that the parking start node is near-optimal, which increases parking success rate and ensures the optimality.

(3) Based on geometry curves, the proposed parking path planning algorithm promotes the C-type parking algorithm, making the parking start node more flexible and improving the scenario applicability. Furthermore, the presented method of calculating the optimal start node provides vital information for choosing the suitable parking start node. Parking geometry curves of other types can also be improved in the same way, which needs to be investigated in future works.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

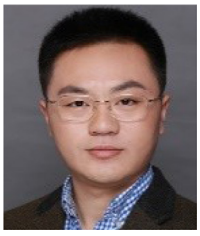
The authors would like to appreciate the financial support of Key R&D Program of Hunan Province, China under 2019GK2151 and the National Key R&D Program of China under 2017YFC0821102.

References

[1] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. 4 (2) (1968) 100–107.
[2] A.K. Guruji, H. Agarwal, D.K. Parsediya, Time-efficient A* algorithm for robot path planning, Proc. Technol. 23 (2016) 144–149.
[3] A. Nash, K. Daniel, S. Koenig, et al., Theta*: Any-angle path planning on grids, AAAI 7 (2007) 1177–1183.
[4] A. Nash, S. Koenig, Any-angle path planning, AI Mag. 34 (4) (2013) 85–107.
[5] I. Pohl, Heuristic search viewed as path finding in a graph, Artif. Intell. 1 (3–4) (1970) 193–204.
[6] E.A. Hansen, R. Zhou, Anytime heuristic search, J. Artificial Intelligence Res. 28 (2007) 267–297.
[7] Z.A. Algfoor, M.S. Sunar, A. Abdullah, A new weighted pathfinding algorithms to reduce the search time on grid maps, Expert Syst. Appl. 71 (2017) 319–331.
[8] D.D. Harabor, A. Grastien, Online graph pruning for pathfinding on grid maps, in: Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.
[9] D.D. Harabor, A. Grastien, D. Öz, et al., Optimal any-angle pathfinding in practice, J. Artificial Intelligence Res. 56 (2016) 89–118.
[10] L.E. Dubins, On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents, Amer. J. Math. 79 (3) (1957) 497–516.
[11] J. Reeds, L. Shepp, Optimal path for a car that goes bothforward and backward, Pacific J. Math. 145 (2) (1990) 367–393.
[12] T. Fraichard, A. Scheuer, From Reeds and Shepp's to continuous-curvature paths, IEEE Trans. Robot. 20 (6) (2004) 1025–1035.
[13] H. Mouhagir, R. Talj, V. Cherfaoui, et al., Integrating safety distances with trajectory planning by modifying the occupancy grid for autonomous vehicle navigation, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2016, pp. 1114–1119.
[14] H. Mouhagir, V. Cherfaoui, R. Talj, et al., Trajectory planning for autonomous vehicle in uncertain environment using evidential grid, IFAC-PapersOnLine 50 (1) (2017) 12545–12550.
[15] D. Dolgov, S. Thrun, M. Montemerlo, J. Diebel, Practical search techniques in path planning for autonomous driving, in: Proc. of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR-08), 2008.
[16] D. Dolgov, S. Thrun, M. Montemerlo, J. Diebel, Path planning for autonomous Vehicles in unknown Semi-structured Environments, Int. J. Robot. Res. 29 (5) (2010) 485–501.
[17] F. Esposto, J. Goos, A. Teerhuis, et al., Hybrid path planning for non-holonomic autonomous vehicles: An experimental evaluation, in: 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), IEEE, 2017, pp. 25–30.
[18] K. Tu, S. Yang, H. Zhang, et al., Hybrid A* based motion planning for autonomous vehicles in unstructured environment, in: IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, parking: 1-4.
[19] Xiaoyang Hou, Research on Automatic Parallel Parking System Based on Route Planning (Master thesis), Department of Automotive Engineering, Tsinghua University, Beijing, 2015.
[20] S. Si, Research on Steering Control Strategy of Autonomous Vertical Parking System Master thesis, Department of Automotive Engineering, Tsinghua University, Beijing, 2015.
[21] Y. Yi, Z. Lu, Q. Xin, et al., Smooth path planning for autonomous parking system, in: 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2017, pp. 167–173.
[22] H. Vorobieva, N. Minoiu-Enache, S. Glaser, et al., Geometric continuous-curvature path planning for automatic parallel parking, in: 2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC), IEEE, 2013, pp. 418–423.
[23] H. Vorobieva, S. Glaser, N. Minoiu-Enache, et al., Automatic parallel parking with geometric continuous-curvature path planning, in: 2014 IEEE Intelligent Vehicles Symposium Proceedings, IEEE, 2014, pp. 465–471.
[24] H. Vorobieva, S. Glaser, N. Minoiu-Enache, et al., Automatic parallel parking in tiny spots: Path planning and control, IEEE Trans. Intell. Transp. Syst. 16 (1) (2014) 396–410, Gómez-Bravo F, Cuesta F, Ollero A, others.
[25] X. Zhang, A. Liniger, A. Sakai, et al., Autonomous parking using optimization-based collision avoidance, in: 2018 IEEE Conference on Decision and Control (CDC), IEEE, 2018, pp. 4327–4332.
[26] K. Zheng, S. Liu, RRT based path planning for autonomous parking of vehicle, in: 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), IEEE, 2018, pp. 627–632.
[27] U. Schwesinger, M. Bürki, J. Timpner, et al., Automated valet parking and charging for e-mobility, in: 2016 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2016, pp. 157–164.
[28] S. Klemm, M. Essinger, J. Oberländer, et al., Autonomous multi-story navigation for valet parking, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2016, pp. 1126–1133.
[29] S. Sedighi, D.V. Nguyen, K.D. Kuhnert, Guided hybrid A-star Path Planning Algorithm for Valet Parking Applications, in: 2019 5th International Conference on Control, Automation and Robotics (ICCAR), IEEE, 2019, pp. 570–575.
[30] Y. Wang, F. Jiang, Y. Luo, et al., A Topological Map-Based Path Coordination Strategy for Autonomous Parking, SAE Technical Paper, 2019.
[31] S. Cameron, P. Probert, Advanced Guided Vehicles: Aspects of the Oxford AGV Project, World Scientific, 1994.
[32] D.D. Harabor, A. Grastien, Improving jump point search, in: Twenty-Fourth International Conference on Automated Planning and Scheduling, 2014.
[33] R. Rajamani, Vehicle Dynamics and Control, Springer Science & Business Media, 2011.

Zhaobo Qin received the B.S. degree and Ph.D. degree in mechanical engineering from Tsinghua University, Beijing, China in 2013 and 2018. He is currently the research associate professor in automotive engineering at Hunan University. His research interest includes automated parking, ADAS and autonomous driving technologies.

Xin Chen received the B.S. degree in mechanical engineering from Qingdao University of Technology, Qingdao, China in 2018. He is currently pursuing the M.Sc. degree in Vehicle Engineering at Hunan University. His research interest includes path planning and automated parking, ADAS.



Manjiang Hu received the B.S. degree and Ph.D. degree in mechanical engineering from Jiangsu University, China in 2009 and 2014. He is currently the research professor in automotive engineering at Hunan University. His research interest includes active safety and intelligent vehicles.

Liang Chen received the B.S. degree in vehicle engineering from Northeastern University, Liaoning, in 2019. He is currently pursuing the M.Sc. degree in Vehicle Engineering at Hunan University. His research interest includes vehicle dynamics and control.

Jingjing Fan received the Ph.D. degree in mechanical engineering from Tsinghua University, Beijing, China in 2011. He is currently the associate professor in mechanical engineering at North China University of Technology. His research interest includes the development of hybrid electric vehicles and ADAS.