Research Article

Parking Space Detection and Path Planning Based on VIDAR

Yi Xu ^{1,2}, Shanshang Gao ¹, Guoxin Jiang ¹, Xiaotong Gong ¹, Hongxue Li ³,
Xiaoqing Sang ¹, Liming Wang ¹, Ruoyu Zhu ¹ and Yuqiong Wang ¹

¹School of Transportation and Vehicle Engineering, Shandong University of Technology, Zibo 255000, China

²Collaborative Innovation Center of New Energy Automotive, Shandong University of Technology, Zibo 255000, China

³School of Vehicle and Energy, Yanshan University, Qinhuangdao 066000, China

Correspondence should be addressed to Yuqiong Wang; wangyuqiong@sdut.edu.cn

Received 23 July 2021; Revised 27 October 2021; Accepted 3 November 2021; Published 2 December 2021

Academic Editor: Shahram Payandeh

Copyright © 2021 Yi Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The existing automatic parking algorithms often neglect the unknown obstacles in the parking environment, which causes a hidden danger to the safety of the automatic parking system. Therefore, this paper proposes parking space detection and path planning based on the VIDAR method (vision-IMU-based detection and range method) to solve the problem. In the parking space detection stage, the generalized obstacles are detected based on VIDAR to determine the obstacle areas, and then parking lines are detected by the Hough transform to determine the empty parking space. Compared with the parking detection method based on YOLO v5, the experimental results demonstrate that the proposed method has higher accuracy in complex parking environments with unknown obstacles. In the path planning stage, the path optimization algorithm of the A* algorithm combined with the Bezier curve is used to generate smooth curves, and the environmental information is updated in real time based on VIDAR. The simulation results show that the method can make the vehicle efficiently avoid the obstacles and generate a smooth path in a dynamic parking environment, which can well meet the safety and stationarity of the parking requirements.

1. Introduction

The increasing number of vehicles leads to increasingly serious parking problems in modern cities. It is a quite challenging task for drivers to park their cars in a limited space. Therefore, the research of automatic parking systems has become a hot issue in the field of intelligent transportation. Parking space detection and path planning are the key technologies to realize automatic parking.

Parking space detection is the primary task of the automatic parking system. With the development of image technology and the improvement of sensor accuracy, vision-based automatic parking system has gradually become the main trend for automatic parking system due to its low cost and high detection speed and accuracy. Various visual-based parking technologies have been proposed in recent years. Wang et al. [1] used four fisheye cameras to construct a real-time visual automatic parking system surrounding the car body and used the Radon transform to extract the parking space features and detect the parking space. Suhr and Jung

[2] proposed a monocular vision-based parking space marker recognition algorithm that used the peak value pair to perform line segment detection and cluster identification of marker lines in Hough space. This method could still effectively detect the parking space under the condition that the adjacent vehicles seriously blocked the parking space marker. The above-mentioned algorithms are not suitable for complex parking space; the accuracy of these algorithms is significantly affected by the transformation parameters. In recent years, target detection methods based on machine learning have been widely applied to parking space detection due to their high accuracy and speed in target recognition [3–5]. Xu and Hu [6] used the image collected by the four-way fisheye camera mounted on the car as input and adopted the YOLO (you only look once) v3 network structure to directly detect the parking space in the image. The algorithm has high recall and precision and can realize real-time detection. Kirtibhai Patel and Meduri [7] presented a Faster R-CNN-based object detection scheme to automatically map the parking space in a parking lot, instead of manually

mapping them. The results show that the method decreases the human effort needed by up to a compelling 86%. Vu and Huang [8] proposed a multitask convolutional neural network with spatial transform for parking space detection. This method crops the local image area adaptively by integrating a convolutional spatial transformer network (STN) and considers the status estimation by designing a multitask loss function, which can reduce the error detection rate. The machine learning method has high detection speed and accuracy for known obstacles such as vehicles, but the misdetection rate of parking space detection is high for parking environment with unknown obstacles. Particularly in some unmanaged open-air parking spaces with obstacles such as stones, garbage boxes, and traffic cones, the accuracy of parking detection algorithm based on machine learning method will be significantly affected. VIDAR [9] is a visual-IMU-based detection and ranging method. The IMU (inertial measurement unit) is used to obtain the moving distance and attitude of the camera, and the height of the object in the image can be calculated through reverse perspective transformation. This method can detect various types of obstacles in the parking environment, and the change of external parking environment factors does not significantly influence the accuracy of parking space detection. In order to improve the accuracy of parking space detection, VIDAR is applied to parking space detection in this paper.

In addition to parking space detection technology, path planning is also a key technology to realize automatic parking of intelligent vehicles. The purpose of path planning is to find a feasible and safe path from the current state to the target state. The existing path planning algorithms applied to the automatic parking systems are mainly divided into two types: geometric methods and graph methods. The geometric method considers the minimum turning radius under the geometric constraints of the environment and solves the problem by combining geometric primitives, such as straight lines, circles, and spirals [10–14]. However, most of the studies that applied geometric methods to the automatic parking system did not consider the limitations of parking spaces, which did not work well in a narrow space environment. For the limited parking space environment, Yang et al. [15] and Maekawa et al. [16] generated a smooth parking track by reducing the turning radius and smoothing the path. Li et al. [17] used the preview algorithm to detect the curvature outliers in parking paths to solve the problem of planned curvature discontinuity between straight lines and arcs. The above-mentioned methods are simple but can only be operated under predefined conditions. In addition, these methods only targeted one or two of the three traditional parking spaces, lacking generality and complicated real-time calculation.

The graph-searching algorithm applied to the automatic parking system is developed from the existing robot path planning algorithm. The graph-searching methods include Hybrid-A* or RRT [18–20]. These methods first construct a graph from the environment and then find an optimal path in the graph. These methods apply vehicle kinematics constraints based on

the existing path planning methods of mobile robots. The most common planning methods for mobile robots are the traditional grid method based on the heuristic genetic algorithm, the artificial potential field method [21–23], and the A* algorithm based on node search [24]. All these methods have a strong ability of path finding and strong adaptability of path planning in various complex parking environments. However, in the field of automatic parking, these methods need to consider vehicle kinematics constraints. Dolgov et al. [25] proposed a vehicle kinematics algorithm (Hybrid-A*) that was applied in DARPA's Urban Challenge. Zheng and Liu [18] adopted the RRT algorithm to search the parking path of vehicles with constraints. Target preference and double RRT strategies were used to optimize the algorithm, which improved the path search efficiency. However, the path generated by this method is not smooth, and the path quality is often poor. In order to solve the above problem, Zhang et al. [26] designed a smooth parking path planning method under irregular and narrow conditions based on the Hybrid-A* algorithm and the OBCA algorithm. However, the above planning methods do not consider the changes in the parking environment. Thus, it is difficult to apply these methods to the actual parking scene because the path cannot be adjusted during the parking process.

To improve the practicability of path planning algorithm, Shen et al. [27] adopted the dynamic window method based on the improved A* algorithm and combined the dynamic characteristics of intelligent vehicles to conduct local obstacle avoidance and path planning for vehicles in view of the unknown and changing environmental information around the vehicles. As explained so far, the application of machine learning methods in automatic parking systems has the defect of missing the detection of generalized obstacles. Parking paths and detecting parking spaces may encounter obstacles. In addition, the path planning by the geometric method is only for one or two of the three traditional parking spaces, which lacks generality and complicated real-time calculation. In this paper, the VIDAR method is used to detect the generalized obstacles, and the Hough method is used to complete parking space detection. Combined with the A* algorithm and the Bezier method, a smooth trajectory in line with vehicle motion constraints can be generated in various complex parking environments. In the process of automatic parking, VIDAR is used to update the environmental information in real time to achieve dynamic planning and improve the reliability and safety of planned paths. This method has universality in various complex parking environments.

The remainder of this paper is organized as follows. Section 2 focuses on the process of parking space detection based on VIDAR. In Section 3, we introduce the formation of the path, which is generated from the A* algorithm combined with the Bezier curve. In Section 4, experimental results of parking space detection and the simulation results of path planning are given. In Section 5, we draw the conclusion of the algorithm in this paper.

2. Parking Space Detection Based on VIDAR

VIDAR is used to detect generalized obstacles in complex parking space environment, which can effectively improve the accuracy of parking space detection. Parking space detection based on VIDAR mainly includes three parts: obstacle detection based on VIDAR, obstacle extraction, and parking space detection.

2.1. Obstacle Detection Based on VIDAR. In a machine vision system, the pinhole imaging model (see Figure 1) is often used to determine the mathematical relationship between the object point and the imaging point of the object in the world coordinate system. The corresponding coordinate of the intersection point of the obstacle and the road plane in the imaging plane coordinate system is (x, y) . The coordinate origin of the plane coordinate system is (x_0, y_0) , which is the intersection point between the image plane and the optical axis of the camera. The effective focal length of the camera is f , the installation height of the camera is h , the pixel size of the camera is μ , and the pitch angle of the camera is ∂ .

The horizontal distance d between the object point and the camera can be obtained by

$$d = \frac{h}{\tan(\partial + \arctan[(y_0 - y)\mu/f])}. \quad (1)$$

The obstacle detection method based on VIDAR is consistent with the principle of the camera ranging model, which is based on the further analysis of the height difference between object points belonging to 3D obstacles and road plane when the monocular camera moves. The obstacle detection method based on VIDAR collects two adjacent frames of images when the monocular camera is moving and uses the height difference between the obstacle points and the ground points to screen the obstacles. The imaging schematic diagram of the static obstacle is shown in Figure 2. Assume that the imaging point of the first obstacle on the imaging plane is A . Due to the movement of the camera, the axis on the imaging plane moves from the axial direction, the imaging point of the obstacle after moving is B , and A' and B' are, respectively, the road plane points corresponding to points A and B . The distance index of feature points on a horizontal plane is Δl , and the horizontal distance from the camera to A' and B' is d_1 and d_2 . d_1 and d_2 can be calculated by

$$d_1 = d_2 + \Delta d + \Delta l. \quad (2)$$

It can be seen from Figure 2 that if object points belong to ground points with no height, the relationship between d_1 and d_2 is $d_1 = d_2 + \Delta d$. If the object point belongs to 3D obstacle with height, the relationship between d_1 and d_2 is $d_1 = d_2 + \Delta d + \Delta l$. Therefore, Δl can be used to determine whether the object point belongs to obstacle or not. The judgment method of feature points is as follows: if $|\Delta l| > u$, the object point belongs to the 3D obstacle; if $|\Delta l| \leq u$, the object point does not belong to 3D obstacle. Since Δl can be

FIGURE 1: Schematic diagram of pinhole imaging.

FIGURE 2: Schematic diagram of static obstacle imaging.

calculated by using IMU to obtain camera attitude data, VIDAR can quickly accurately and quickly detect obstacles.

2.2. Obstacle Extraction. The MSER (Maximally Stable External Regions) algorithm is widely used in image registration and region matching. In this paper, the feature points are extracted using the image region matching method based on MSER [28]. The method based on VIDAR is used to remove the nonobstacle points with no height from the feature points. Thus, the obstacles in the images can be detected quickly.

VIDAR is used to screen the matching points of two adjacent frames in the automatic parking movement, and the set of obstacles after screening is C , $C = \{c_1, c_2, \dots, c_n\}$. The matched images after screening are shown in Figure 3.

In order to determine the space where the obstacle is in the image, the K -means clustering algorithm is used to perform clustering analysis for the coordinate data set C , which is composed of n obstacle matching points.

Cluster the obstacle point c_p ($c_p \in C$) in the data set and $\{u_1, u_2, \dots, u_K\}$, respectively, represent K clustering centres. K -means clustering algorithm is shown as follows [29]:

Step 1: initialize and select K objects as the centre of the initial cluster.

Step 2: calculate the Euclidean distance between the data and the clustering centre by formula (3), and divide the data with a close distance into one category:

$$S_i^{(t)} = \left\{ c_p: \|c_p - u_i\|^2 \leq \|c_p - u_j\|^2 \forall j, \quad 1 \leq j \leq K \right\}. \quad (3)$$

Step 3: recalculate the seed centre of this class by

FIGURE 3: Feature points matching graph based on VIDAR.

$$u_i^{(t+1)} = \frac{1}{s_i^{(t)}} \sum_{c_j \in s_i^{(t)}} c_j. \quad (4)$$

Step 4: traverse all obstacle point data and repeat step 2 and step 3 until the seed centre no longer changes.

The K value can be better determined by plotting the K -SSE curve and by finding the inflection point down. The elbow method was used to determine the selection of K value in the K -means algorithm, and the SSE value can be calculated by formula (5) to obtain the K -SSE curve [30]. According to the K -SSE curve analysis (see Figure 4(a)), there is a very obvious inflection point when $K = 2$. It can be seen that when $K = 2$, the data set clustering effect is the best. It is determined that $K = 2$ is the optimal K value for clustering. According to the optimal K value of K -SSE curve clustering, the clustering results of obstacle points in the parking space are obtained (see Figure 4(b)).

$$SSE = \sum_{K=1}^K \sum_{c_j \in s_j} \|c_j - u_K\|_2^2. \quad (5)$$

Reasonable expansion is made according to the position of the obstacle points around the edges of each K -means cluster. The yellow rectangular region is used to frame and divide the clustering region of each obstacle point (see Figure 5).

The clustering regions taken by yellow rectangles in Figure 5 are denoted as O , and the remaining area is the nonobstacle area, denoted as F .

2.3. Parking Space Detection. Hough transform method [31], the most mature straight line detection method, is used to detect the parking line in the nonobstacle area F . The basic idea of the Hough transform is to measure space point (x_i, y_i) into a parameter space curve. The points with the same parameter characteristics intersect in the parameter space after transformation, and the characteristic curve is detected by judging the degree of accumulation at the intersection points. Its duality formula is

$$\rho = x \cos \theta + y \sin \theta, \quad (6)$$

where ρ is the distance between the origin to the line closest point and θ is the angle between the x -axis and the line connecting the origin with that closest point.

Obstacles are extracted according to the above section, and Hough detection is carried out in nonobstacle areas. The detection edge line segments in the obstacle areas are removed adaptively to reduce the pixels involved in the transformation and the influence of the background interference line segment.

In the real world, due to illumination, unexpected occlusion and blurring of the parking line, and other factors, Hough transform usually cannot detect a complete line, but many line segments. In order to remove the noise interference of the detection of parking lines, we set up a library of linear equations by inherent characteristics of parking lines. The expression of the equation in the library is $y = kx + b$. The linear equation of each line segment is calculated by the coordinate of two endpoints of the line segment. Any edge line is taken as the reference line to judge the equation with the others. If the equation of two lines is approximately the same, the two lines are judged as a parking line. All parking lines can be determined until all detected edge lines are judged.

The line with the same equation is regarded as the same line, and the line is appropriately extended to obtain parking lines (see Figure 6(a)). According to the fixed characteristics of parking lines, four adjacent parking lines satisfying the conditions of parking lines are extracted as parking mark lines. The intersection points of the four extracted parking lines are taken as the four corners of the parking space, and the line segments connected by adjacent corners are taken as the parking lines. Four parking lines can be, respectively, expressed as C_1 , C_2 , T_1 , and T_2 (see Figure 6(b)).

Record the parking space composed of parking lines C_1 , C_2 , T_1 , and T_2 as P_{free} . The length of parking lines L_p ($P = C_1, C_2, C_3, C_4$) and the parking space P_{free} surrounded by four parking lines need to satisfy the following conditions

$$\begin{cases} O \cap P_{free} = \emptyset; \\ L_p > G_d, \end{cases} \quad (7)$$

where G_d is the threshold value of the length of the parking line and O for every obstacle clustering regions.

According to the detection results in Figures 5 and 6, the information of parking spaces and obstacles in the parking area can be obtained to realize the detection of empty parking space. Parking detection results are shown in Figure 7.

FIGURE 4: K -means cluster diagram. (a) Determine the value of K . (b) Obstacle points clustering (circles of the same color are divided into one class; red and blue circles belong to two separate classes).

FIGURE 5: Obstacle extraction.

FIGURE 6: Parking space extraction based on Hough transform. (a) Parking line detection. (b) Parking space extraction.

FIGURE 7: Parking detection based on VIDAR.

3. Path Planning

The planned path is composed of two partial paths. One is the constraint path planned by A* algorithm, and the other is the reverse path designed by parking space information.

3.1. Vehicle Constraints. The planned path is composed of two partial paths. One is the constraint path planned by A* algorithm, and the other is the reverse path designed by parking space information.

In automatic parking, kinematics and dynamics constraints should be considered in the study of vehicle movement. The parking space can be simplified due to the vehicle's low speed and small tire slippage, and the actual kinematics model can be carried out by combining the vehicle model parameters. The kinematics model of the vehicle can be simplified to a pure rolling, which is called a nonholonomic constraint model. The vehicle shape is simplified into a rectangle. The kinematics model of the vehicle is shown in Figure 8.

From Figure 8, we can get the following relationship:

$$\begin{cases} \dot{x} = v \cos \theta, \\ \dot{y} = v \sin \theta, \\ \dot{v} = a, \\ \dot{\theta} = \frac{v}{R}, \\ R = \frac{L}{\tan \phi}, \end{cases} \quad (8)$$

where (x, y) is the coordinate of the centre point of the rear wheel axle of the vehicle; v is the longitudinal velocity of the centre point of the front axle, and a is the acceleration; we define ϕ as the steering angle of the centre point of the front axle of the vehicle (the angle between front wheel direction and body direction); θ is the included angle between the centre axis of the vehicle and the horizontal direction; L is the front and rear wheel wheelbase; L_p is the distance between the front axle and the front end of the vehicle; L_r is the distance between the rear axle and the rear end of the vehicle; W is the width of the vehicle; R is the connection between vehicle instantaneous centre C and rear wheel axle.

It can be known from formula (8) that when the ϕ value of the current wheel angle is the maximum, the steering radius R of the vehicle at this time is the minimum, $R_{\min} = (L/\tan \phi_{\max})$. In plane coordinates, the path curve can be expressed as $y = f(x)$, and its curvature formula can be obtained as follows:

$$k = \frac{|y''|}{[1 + (y')^2]^{3/2}}, \quad (9)$$

where $k_{\max}$ is the maximum curvature of the curve and the constraint relationship between the curvature k of the curve path planning and the minimum steering radius $R_{\min}$ is

FIGURE 8: Vehicle kinematics diagram.

$$k_{\max} < \frac{1}{R_{\min}}. \quad (10)$$

3.2. Set Transition Target Point. In order to make full use of the known information, in the whole parking process of the car, the position information of the parking space can generally be obtained by the external sensor (see Section 2.3). The way of reverse path can be used to plan the last section of the automatic parking back into the storage path and set the intermediate transition target point $P_g(x_g, y_g)$.

Parallel, vertical, and tilted parking spaces are the most common parking spaces. This paper considers the three most common parking spaces as research scenarios. Assume that the vehicle has been parked in the parking space. The coordinate points at the position 0.75 of the long axis of the parking line and the middle position of the short axis are taken as the final parking point $P_e(x_e, y_e)$, which corresponds to the coordinates of the centre point of the rear axis in the vehicle kinematics model. $P_g(x_g, y_g)$ is the other endpoint of the reverse path, which is the target point based on A* algorithm. Three parking path models are as follows.

As shown in Figure 9, the three kinds of automatic parking reverse paths are all composed of straight lines and arcs. The relative position relationship between P_e and P_g on the reverse path can be deduced and determined by the basic path method in literature [26]. When the vehicle exits from the parallel parking space (see Figure 9(a)), it will first move a distance of d_p along the long axis parking line and then move to P_g by two arcs with continuous curvature. The radius of the first arc is R_1 and the arc angle is θ_1 ; the radius of the second arc is R_2 and the arc angle is θ_2 . When the vehicle exits from the vertical parking space (see Figure 9(b)), it will first move along the long axis parking line for a distance d_v and then move to the target point through an arc path with a radius of R_3 and an arc angle of θ_3 . When the vehicle exits from the tilted parking space (see Figure 9(c)), it will first move along the long axis parking line for a certain distance d_t and then move to the target point through an arc path with a radius of R_4 and an arc angle of θ_4 .

Through the reverse path of the above three kinds of parking spaces and combining with the location information

FIGURE 9: Reverse parking path planning. (a) Parallel parking. (b) Vertical parking. (c) Tilted parking.

of the target parking space extracted in Section 2.3, the appropriate target point position under the three kinds of parking spaces can be calculated. Automatic parking can be finished directly without adjusting parking posture by designing reversed driving-out paths, the path is smoother, and the safety margin is larger than the method by A^* algorithm.

3.3. Path Planning Based on A^* Algorithm

3.3.1. A^* Algorithm. The A^* algorithm is one of the most classical robot path tracking methods. It is more convenient to impose vehicle motion constraints based on A^* algorithm planning, making it easier to meet vehicle kinematics constraints [18, 27, 28]. The A^* algorithm is used to generate the planned path in the process of automatic parking of ego vehicles. The A^* algorithm is a heuristic search algorithm, which is the most effective method to find the shortest path in the static network. The process of A^* algorithm is as follows.

Set the starting point (x_0, y_0) as the initial node and the intermediate transition target point (x_g, y_g) (see Section 3.2) set above as the target node. The evaluation function is used to generate a global optimal path. Each time, the node with the smallest value of the evaluation function is selected, and the destination point (x_g, y_g) to be reached is found through the algorithm. Finally, the global optimal path from the initial position to the target position is generated. The cost estimation function of A^* algorithm is

$$f(n) = g(n) + h(n), \quad (11)$$

where $f(n)$ is the total cost estimation function from the initial node to the target node, $g(n)$ is the actual cost from the initial node to the current node n in the state space, and $h(n)$ is the optimal cost estimation of the path from the current node to the target node.

To ensure the safety of the automatic parking process, the method of obstacle edge expansion is adopted to avoid the collision between the outer edge of the vehicle and the obstacle. The expanded distance should be larger than half of the width of the vehicle, and a certain safe distance da should be reserved to avoid the planned path clinging to obstacles.

The maximum expansion of distance of obstacles is $(W/2) + da$ (see Figure 10).

As shown in Figure 11, the planned path based on A^* algorithm avoids the danger of collision with surrounding obstacles by means of obstacle extension. The initial path generated is I_1 .

3.3.2. Path Optimization. The initial path I_1 obtained by A^* algorithm is composed of multiple straight line segments. The path has too many abrupt turning points, which does not meet the requirements of vehicle kinematics. Therefore, it is necessary to optimize the path curve.

To solve the problem above, the greedy strategy is used to smooth path trajectory I_1 ; specific algorithm steps are as follows: connect the target point (x_g, y_g) from the initial point (x_0, y_0) as the starting point. If obstacles exist in the path of connection, the next turning point (x_1, y_1) on the path is taken as the starting point, which is connected to the target point (x_g, y_g) . Repeat the above operation until the path of connection between the point (x_i, y_i) and the target point (x_g, y_g) does not pass through obstacles. Make $\begin{cases} x_i = x_g \\ y_i = y_g \end{cases}$; that is, the point (x_i, y_i) is regarded as the target point; repeat the above operation until the initial starting point (x_0, y_0) is connected with the target point (x_g, y_g) . The path I_2 optimized by the greedy strategy is as follows (see Figure 12). The blue line represents the initial path I_1 , and the red line represents the planning path I_2 optimized by the greedy strategy.

Then, the Bezier formula is used to smooth the planned path I_2 . The Bezier curve can fit any number of control points, and the degree of the Bezier polynomial is related to the position of the control points. Nodes are evenly inserted from P_0 to P_g . P_0 and P_g of the path are seen as data points, and the necessary turning points in the planning curve are seen as control points.

Start from P_0 ; a node is set at every interval distance s between the control point and the data point. The last node of a straight path is the necessary turning point J_i , $d_{J_i, J_{i-1}} \leq s$, and then starting from the necessary turning point J_i , a node

FIGURE 10: Obstacle expansion.

FIGURE 11: Path planning based on A* algorithm.

FIGURE 12: The generation process of greedy strategy curve.

is set for every fixed distance of path s , until it reaches to P_g . Suppose the number of control nodes is n . P_0 is denoted as J_0 , P_g is denoted as J_n , and the intermediate nodes are denoted as $J_1, J_2, \dots, J_{n-1}$. The node from J_0 to J_n is fitted by Bezier as shown in

$$B_n(t) = \sum_{i=0}^n C_n^i J_i (1-t)^{n-i} t^i, \quad t \in [0, 1]. \quad (12)$$

After inserting nodes according to the above method, all nodes are fitted by formula (13). We can obtain Bezier curve I_3 with continuous curvature. As shown in Figure 13, the blue line represents planning path I_2 and the green line represents smooth path I_3 fitted by Bezier curves. Compared with the planned path I_2 , the smooth path I_3 mainly changes near the key turning point, and the node positions in other straight path regions are unchanged. The node whose position remains unchanged after Bezier smoothing is taken as the new path node, and the node position set is denoted as P , $P = \{P_0, P_1, \dots, P_g\}$.

FIGURE 13: The generation of the Bezier curve optimization.

3.3.3. Real-Time Obstacle Avoidance Strategy. Considering that the environmental information in the actual parking environment is relatively complex and changeable, it is inevitable that unknown dynamic obstacles will appear in the movement process of the ego vehicle, which will change the environment. A real-time obstacle avoidance strategy is proposed in the parking process to realize the real-time obstacle avoidance of automatic parking. VIDAR is used to update the parking environment information. Real-time obstacle avoidance strategies are as follows.

Step 1: the intelligent vehicle moves from the starting point P_0 to the new node P_i at a constant speed v on the planned path, and the vehicle position coordinates after the first movement are P_1

Step 2: when the path moves to the new node P_i , use VIDAR to detect generalized obstacles on the path (see Section 2.1), and update the parking environment information

If any feature point on the planning path satisfies $|d_1 - \Delta d - d_2| < k$, it indicates that there is no new dynamic obstacle, and it continues to move to the new node P_{i+1} at a constant speed v along the original path

If the feature points on the planning path satisfy $|d_1 - \Delta d - d_2| > k$, it indicates that there are new dynamic obstacles. Let the next node closest to the vehicle P_{i+1} be P_0 , namely, $P_{i+1} = P_0$, and the coordinate position of the target point P_g remains the same. Generate a new parking path from node P_0 to P_g according to the path planning smoothing method (see Section 3.3.2). Continue moving at constant speed v to the target point on the new planning path.

Step 3: until $P_i = P_g$, it indicates that the vehicle has reached the target point according to the planned path and parking stops

3.4. Integrated Planning. Two monocular cameras installed on the front and rear cover in the centre of the ego vehicle are used to collect environmental information. The front-end monocular camera is used for the path planned by A* algorithm to obtain the images, and the camera on the trunk cover is used for the final reversed path to obtain the images. The overall process of parking detection and path planning based on VIDAR is shown in Figure 14.

The overall steps of parking space detection and path planning based on VIDAR are as follows:

FIGURE 14: Flowchart of parking space detection and path planning based on VIDAR.

- (1) Zhang Zhengyou's camera calibration method is used to obtain the camera parameters fixed on the detection vehicle, and the camera and IMU are used to carry out image acquisition and inertial data acquisition, respectively, during the parking movement to complete the update of camera parameters.
- (2) The obstacle points are detected by VIDAR, and the data set C composed of the obstacle points is clustered by the K -means clustering method. Each cluster region is denoted as the obstacle region O , and other regions are denoted as the nonobstacle region F .
- (3) In the nonobstacle area F , the improved Hough transform method is used to detect the parking space, and the detected parking space P_{free} satisfies the condition: $O \cap P_{\text{free}} = \emptyset$.
- (4) Based on the detected target parking space, the exit path is simulated as the parking path, and the parking trajectory can be inverted to determine in the parking area, and the appropriate intermediate target point (x_g, y_g) is set.
- (5) The starting point (x_0, y_0) is taken as the initial node, and (x_g, y_g) is taken as the target node. A^* algorithm is used to carry out path planning between the two nodes and then smooth the planned path.
- (6) When the vehicle is parking along the planned path at a constant speed v , the parking environment information is updated in real time. The vehicle moved to (x_g, y_g) by using the real-time obstacle avoidance strategy (see Section 3.3.3).
- (7) After arriving at the point (x_g, y_g) , the vehicle moves to the point (x_e, y_e) along the backward path of step (4) and finally completes the automatic parking.

4. Experiments and Evaluation

This section aims to verify the effectiveness of the method proposed in this paper. Firstly, the experimental platform of this paper is introduced. Secondly, in the parking detection stage, the target detection results of the proposed method are compared with the results of the YOLO v5 method, highlighting the important role of VIDAR in effectively reducing the rate of missed parking detection and improving the accuracy of parking detection. Finally, the reliability of the improved path planning algorithm based on the A^* algorithm is verified by simulation. The simulation results show that the proposed method can generate smooth paths in different parking spaces.

4.1. Experimental Vehicle Platform Construction. The algorithm proposed in this paper is developed and tested on the experimental platform (see Figure 15). The experimental platform takes a pure electric vehicle as the main body, which is also equipped with experimental equipment such as camera, IMU, electronic power steering gear, and laptop. Path simulation experiments are generated by using MATLAB. The geometry of the pure electric is known, vehicle length $L_r = 2.84$ m, vehicle width $W = 1.44$ m, and wheelbase $L = 2.25$ m. The monocular camera fixed in the centre of the front of the ego vehicle is used to collect the image of the parking environment. The mounting height of the camera is 0.715 m, and the camera pitch angle is 0 degrees. In the stage of camera calibration, Zhang Zhengyou's camera calibration method is used to obtain the relevant parameters of camera. IMU, as vehicle information acquisition equipment, can collect the information of the motion state of the ego vehicle. As a processor, the laptop can deal

FIGURE 15: Experimental platform.

with the output signal of each sensor in real time and decide the subsequent movement of the vehicle.

4.2. Parking Space Detection Accuracy Analysis. YOLO series is representative of the target recognition framework based on artificial intelligence deep learning technology. YOLO v5 is the most superior machine learning approach that has been widely used in current parking space recognition due to its high accuracy and speed in target recognition. The parking environment of the experiment was recorded using the Da Ying camera, and the camera attitude data was recorded by HEC295 IMU. The image data and the camera attitude data were processed by the obstacle detection methods of YOLO v5 and VIDAR.

In the real world, some parking spaces may have some nonstandard phenomena; for example, underground parking spaces may have debris (cardboard boxes, traffic cones, etc.) and accumulation phenomena. The outdoor parking spaces may have a parking space phenomenon, and parking spaces are occupied by chairs, stones, and other unknown obstacles. These conditions will affect the accuracy of parking space detection; especially, the obstacle detection method based on machine learning cannot detect positional obstacles. Four images of parking spaces occupied by unknown obstacles were selected, and obstacles were detected using YOLO v5 and the proposed method. The results are shown in Figure 16.

In this paper, the experiments were carried out in underground and open parking spaces. After three weeks of experiments, 872 experimental images of parking spaces were collected. YOLO v5 and the proposed VIDAR-based parking detection methods were used to detect the parking spaces in each collected parking image. Under the same experimental conditions, the detection results of the two methods are compared. Accuracy and recall rate were used as quantitative indexes. The precision P and the recall rate R can be obtained as

$$\begin{cases} P = \frac{TP}{TP + FP}, \\ R = \frac{TP}{TP + FN}, \end{cases} \quad (13)$$

where TP represents the number of samples that correctly detect empty parking space; FP represents the number of

samples wrongly detected as empty parking spaces; FN represents the number of samples that are actually empty parking spaces but wrongly detected as nonempty parking spaces.

Under the same experimental conditions, YOLO v5 and VIDAR parking detection methods are used for the comparative experiment of parking detection. Figures 17(a) and 17(b) show the obstacle detection results based on YOLO v5 and VIDAR, respectively, while Figures 17(c) and 17(d) show the parking detection results based on Hough transform.

It can be seen from Figures 17(a) and 17(c) that the result of the parking space by YOLO v5 is incorrect and the parking space with unknown obstacles is identified as empty. YOLO v5 can accurately detect known types of obstacles. Cartons are unknown target types in the training dataset of YOLO v5. Therefore, the method based on YOLO v5 cannot detect cartons as obstacles. In the case of unknown obstacles (carton, traffic marker, etc.) in the parking space, the accuracy of the parking detection method based on YOLO v5 is significantly affected. Figures 17(b) and 17(d) show that the method proposed in this paper can directly and accurately detect all generalized obstacles. Compared with the YOLO v5 method, the proposed method can effectively reduce the false rate of parking space detection.

From Tables 1–3, it can be seen that compared with the parking detection method based on YOLO v5, the method proposed in this paper has higher detection accuracy and recall rate, especially in the outdoor parking lot. This is because the outdoor parking lot may be unmanaged for a long time. Obstacles such as small rocks, garbage bags, traffic cones, pedestrians, and other obstacles often exist in the parking space, which greatly affects the accuracy of parking space detection methods based on YOLO. It can be seen from the above parking space experiment results that, compared with the YOLO method, the parking space detection method based on VIDAR has stronger accuracy and robustness.

4.3. Path Planning Experiment. Parking experiments were conducted in a grid diagram with generalized obstacles, and the A^* algorithm based on node search was used to plan the path. The Bezier curve can fit any number of control nodes when the generated path is smoothed by the Bezier formula (12). The optimization method of the Bezier curve is to insert a node every certain distance s on the generated path and finally fit all the nodes to complete the smooth optimization of the path. The set insertion distance s significantly impacts the smoothing result. When more nodes are inserted, the distance s between these nodes becomes shorter than before. It could further induce small deviation of the smooth path (generated by Bezier curve) from the original path, finally reducing the possibility of collision. When the distance between the nodes is too large, the smooth curve is prone to deviate from the existing straight path and collide with obstacles. When the distance between the nodes is too small, the vehicle curvature constraint formula cannot be satisfied, which will increase unnecessary calculation and overall

FIGURE 16: Obstacle detection results based on YOLO v5 and the method proposed in this paper.

FIGURE 17: Parking detection based on VIDAR and YOLO v5. (a) Obstacle detection based on YOLO v5. (b) Obstacle detection based on VIDAR. (c) Parking detection based on YOLO v5. (d) Parking detection based on VIDAR.

TABLE 1: The results in the parking space detection experiment of outdoor parking lot.

Method	Input	P (%)	R (%)
YOLO v5	3525	82.4	85.4
VIDAR	3525	91.8	89.7

TABLE 2: The results in the parking space detection experiment of underground parking lot.

Method	Input	P (%)	R (%)
YOLO v5	3765	91.4	87.4
VIDAR	3765	92.4	90.4

TABLE 3: The total results of both in the parking space detection experiment.

Method	Input	P (%)	R (%)
YOLO v5	7290	86.7	86.4
VIDAR	7290	92.1	90.0

FIGURE 18: Determination of optimal control node distance s . (a) $s = 0.4$, (b) $s = 0.8$, and (c) $s = 1.5$.TABLE 4: Simulation results of $s = 0.4$, $s = 0.8$, and $s = 1.5$ in the same environment.

Method	Avg. time (s)	Avg. number of nodes in path	Success rate of 30 environmental scenarios (%)
$s = 0.4$	1.177	84	97.5
$s = 0.8$	1.175	51	84.7
$s = 1.5$	1.173	27	63.2

calculation time. The node distance parameters $s = 0.4$, $s = 0.8$, and $s = 1.5$ were used to analyze the simulation results as shown in Figure 18.

Table 4 shows that there is a slight difference in the path generation time for node distance parameters $s = 0.4$, $s = 0.8$, and $s = 1.5$. However, the success rate of collision-free path generation is much higher when $s = 0.4$ than that when $s = 0.8$ and $s = 1.5$. Thus, $s = 0.4$ is the optimal node distance parameter.

Therefore, the path optimized by the Bezier formula (12) sets a control node at every interval distance 0.4. The simulation results of path planning of the algorithm are shown in Figure 19. The blue line segment in Figure 19(a) represents the initial trajectory generated by the A^* algorithm, the green line segment in Figure 19(b) is the path after greedy strategy, and the red line segment in Figure 19(c) shows the path optimized by the proposed algorithm. The optimized planned trajectory eliminates the redundant abrupt turning points, can effectively avoid the obstacles, and has the advantage of smoothness that meets the kinematics requirements of the vehicle in actual motion.

Another advantage of the path planning algorithm in this paper is that the path planning is real-time. In this paper, VIDAR is used to directly and quickly detect generalized obstacles. In order to cope with changes in the parking environment, the VIDAR algorithm is used to update the parking environment information in real time; then, a real-time obstacle avoidance strategy is added in the process of path planning (see Section 3.3.3). In the process of driving along the predetermined trajectory, the ego vehicle can avoid obstacles in real time and replan the path after forecasting the collision situation, which can effectively reduce the occurrence of a collision. This method proposed is reliable and accurate and can generate a safe obstacle avoidance path under a dynamic environment. As shown in Figure 20, after the moving obstacle is detected (red square), the simulation results of parking paths comparison with real-time obstacle avoidance strategy are shown in (a) and (b). It can be seen that the algorithm in this paper can effectively improve the safety of automatic parking motion by adding a real-time obstacle avoidance strategy.

FIGURE 19: Path optimization based on the A* algorithm. (a) Before optimization. (b) Greedy strategy optimization. (c) Bezier optimization.

FIGURE 20: Schematic diagram of simulation results before and after adding real-time obstacle avoidance strategy. (a) Collision situation. (b) Obstacle avoidance situation.

FIGURE 21: Three kinds of parking space path planning simulation diagram. (a) Plan path in parallel parking space. (b) Plan path in vertical parking. (c) Plan path in tilted parking.

The proposed path planning algorithm can generate the planned path in the parking environment with generalized obstacles. The algorithm is tested in three typical parking spaces, namely, vertical parking space, parallel

parking space and tilted parking space, and parking (see Figure 21). The simulation results show that the path is composed of the path planned by A* algorithm and the reverse path designed by parking space information. Not

only can the path generated avoid the obstacles effectively, but also the path on the turning point is smooth; this method can meet the continuity of the parking requirements very well.

5. Conclusion

In this paper, VIDAR is introduced into parking space detection to detect the generalized obstacles, which significantly reduces the missed detection of unknown obstacles and improves the accuracy of parking space detection. An improved path planning method based on the A* algorithm is proposed. The generated trajectory is smooth and meets the kinematic constraints of vehicles. The proposed method is suitable for various parking spaces and meets the requirements of automatic parking movement. At the same time, the parking environment information is updated by VIDAR to make the path planning real time, which ensures the reliability of path planning.

The method proposed in this paper requires a large amount of computation. Therefore, reducing the computation time and improving the efficiency of the proposed method will be the next research direction. At the same time, the automatic parking system mainly consists of three parts, including parking space detection, path planning, and tracking control. Parking detection and path planning have been introduced in detail in this paper, and the next research direction will mainly focus on tracking control function of automatic parking so as to achieve a complete automatic parking system.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 51905320, the China Postdoctoral Science Foundation under Grants 2018M632696 and 2018M642684, the Shandong Key R&D Plan Project under Grant 2019GGX104066, and SDUT & Zibo City Integration Development Project under Grant 2017ZBXC133.

References

- [1] C. Wang, H. Zhang, M. Yang, X. Wang, L. Ye, and C. Guo, "Automatic parking based on a bird's eye view vision system," *Advances in Mechanical Engineering*, vol. 6, Article ID 847406, 2014.
- [2] J. K. Suhr and H. G. Jung, "Full-automatic recognition of various parking slot markings using a hierarchical tree structure," *Optical Engineering*, vol. 52, no. 3, p. 7203, 2013.
- [3] J. Y. Chen and C. M. Hsu, "A visual method for the detection of available parking slots," in *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff, Canada, October 2017.
- [4] L. Li, L. Zhang, X. Li, X. Liu, Y. Shen, and L. Xiong, "Vision-based parking-slot detection: a benchmark and a learning-based approach," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 10, no. 3, Hong Kong, China, 2017.
- [5] X. An, H. Deng, X. Shi et al., "Detection method for parking spaces based on mini-convolution neural network," *Journal of Computer Applications*, vol. 38, no. 4, pp. 935–938, 2018.
- [6] C. Xu and X. Hu, "Real time detection algorithm of parking slot based on deep learning and fisheye image," *Journal of Physics: Conference Series*, vol. 1518, Article ID 012037, 2020.
- [7] R. Kirtibhai Patel and P. Meduri, "Faster R-CNN based automatic parking space detection," in *Proceedings of the 3rd International Conference on Machine Learning and Machine Intelligence*, Hangzhou, China, 2020.
- [8] H. T. Vu and C. C. Huang, "A multi-task convolutional neural network with spatial transform for parking space detection," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017.
- [9] Y. Xu, S. Gao, S. Li et al., "Vision-IMU based obstacle detection method," in *Proceedings of the International Conference on Green Intelligent Transportation System and Safety*, Changchun, China, 2017.
- [10] A. A. Zhdanov, D. M. Klimov, V. V. Korolev, and A. E. Utemov, "Modeling parallel parking a car," *Journal of Computer and Systems Sciences International*, vol. 47, no. 6, pp. 907–917, 2008.
- [11] D. Kim, W. Chung, and S. Park, "Practical motion planning for car-parking control in narrow environment," *IET Control Theory & Applications*, vol. 4, no. 1, pp. 129–139, 2016.
- [12] S. Zhang, M. Simkani, and M. H. Zadeh, "Automatic vehicle parallel parking design using fifth degree polynomial path planning," in *Proceedings of the 2011 IEEE Vehicular Technology Conference (VTC Fall)*, San Francisco, CA, USA, September 2011.
- [13] J. Van Den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.
- [14] Y. Zhu, J. Li, M. Feng, and Y. Xu, "Autonomous parking path planning algorithm for intelligent vehicles based on numerical optimization," *Journal of Military Transportation University*, vol. 21, no. 11, 2019.
- [15] W. Yang, L. Zheng, Y. Li, Y. Ren, and Y. Li, "A trajectory planning and fuzzy control for autonomous intelligent parking system," *SAE Technical Paper*, vol. 1, 2017.
- [16] T. Maekawa, T. Noda, S. Tamura, T. Ozaki, and K.-i. Machida, "Curvature continuous path generation for autonomous vehicle using B-spline curves," *Computer-Aided Design*, vol. 42, no. 4, pp. 350–359, 2010.
- [17] C. Li, H. Jiang, S. Ma, S. Jiang, and Y. Li, "Automatic parking path planning and tracking control research for intelligent vehicles," *Applied Sciences*, vol. 10, no. 24, pp. 91–115, 2020.
- [18] K. Zheng and S. Liu, "RRT based path planning for autonomous parking of vehicle," in *Proceedings of the 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, Enshi, China, 2018.
- [19] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Applied Soft Computing*, vol. 77, pp. 236–251, 2019.

- [20] X. Wang, H. Shi, and C. Zhang, "Path planning for intelligent parking system based on improved ant colony optimization," *IEEE Access*, vol. 8, pp. 65267–65273, 2020.
- [21] X. Q. Xu and Q. B. Zhu, "Multi-artificial fish-swarm algorithm and a rule library based dynamic collision avoidance algorithm for robot path planning in a dynamic environment," *Dianzi Xuebao (Acta Electronica Sinica)*, vol. 40, no. 8, pp. 1694–1700, 2016.
- [22] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*, pp. 396–404, Springer, New York, NY, USA, 2011.
- [23] Z. Feng, S. Chen, Y. Chen, and N. Zheng, "Model-based decision making with imagination for autonomous parking," in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, China, 2018.
- [24] D. Wang, "Indoor mobile-robot path planning based on an improved *A algorithm," *Journal of Tsinghua University (Science and Technology)*, vol. 2012, pp. 1085–1089, 2012.
- [25] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.
- [26] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, Miami, FL, USA, 2018.
- [27] G. Shen, J. Shao, D. Tan, Y. Niu, and S. Gao, "Optimal hybrid path planning considering dynamic and static characteristics of intelligent vehicles," in *Proceedings of the International Symposium on Pervasive Systems, Algorithms and Networks*, Naples, Italy, 2019.
- [28] A. Śluzek, "Improving performances of MSER features in matching and retrieval tasks," in *Proceedings of the European Conference on Computer Vision*, Amsterdam, Netherlands, 2016.
- [29] M. Cui, "Introduction to the K-means clustering algorithm based on the elbow method, accounting," *Auditing and Finance*, vol. 1, no. 1, pp. 5–8, 2020.
- [30] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, "Integration k-means clustering method and elbow method for identification of the best customer profile cluster," *IOP Conference Series: Materials Science and Engineering*, vol. 336, no. 1, Article ID 012017, 2018.
- [31] S. Jiang, Y. Lu, Y. Chen, and X. Zou, "Parking-lines detection based on an improved Hough transform," in *Proceedings of the International Conference on Robotics and Machine Vision*, vol. 10613, Kitakyushu, Japan, 2017.