Bi-RRT* based Parking Planning with Fast Path Collision Check Method

12232434 Sun Yaowei

摘要—To solve the issues of time-consuming collision detection and low sampling efficiency in the Bi-RRT* algorithm for parking problems, an optimized path collision detection method is proposed. Additionally, a road edge sampling method based on road information is introduced to enhance the sampling efficiency. Simulation results indicate that the improved path collision detection method outperforms the original method in terms of effectiveness, and the effect of road edge sampling method is also good.

***Index Terms*—Bi-RRT, Reeds-Shepp, sample optimize, parking, collision check**

I. Introduction

As an important part of autonomous driving, automatic parking technology aims to park vehicles safely and accurately in designated parking spaces without human intervention. With the increasing severity of urban traffic congestion and parking difficulties, automatic parking technology has received widespread attention. Automatic parking systems can help drivers easily complete parking tasks in narrow parking spaces, improve driving convenience, reduce parking accidents, and increase parking efficiency.

Rapidly-exploring Random Trees (RRT) is a randomization algorithm for robot path planning and other exploration problems. It is particularly suitable for dealing with complex issues, such as high-dimensional spaces and dynamic constraints. Moreover, it can quickly find feasible paths in large, complex environments, even though these paths may not be optimal. Bi-RRT, a variant of RRT, simultaneously builds two search trees from both the initial state and the target state.

In parking planning problem, Bi-RRT* constructs a road exploration tree starting from the vehicle's location and a narrow parking space exploration tree starting from the parking spot. They expand towards each other until the two trees meet in the state space. Bi-RRT* often finds a feasible path faster than RRT.

II. Relate Works

Researchers have adopted various methods to solve parking problem, including geometry-based methods, spline curve-based methods, optimization-based methods, and graph search-based methods. These methods need to consider the kinematic and dynamic constraints of vehicles, parking space limitations, and obstacles in the surrounding environment. With the rapid development of autonomous driving technology, automatic parking planning faces more challenges, such as real-time performance, stability, adaptability, and reliability in complex scenes.

In parking planning problem, especially in narrow Spaces, it is very important to consider the steering radius limits and incomplete sliding constraints. The Reed-Shepp curve provides a way to satisfy these constraints, making the generated path more realistic and feasible.

The Bi-RRT* algorithm is an asymptotically optimal path planning method proposed by Klemm[1] in 2015. Building on the RRT* algorithm, Bi-RRT* introduces a dual-tree strategy to enhance search efficiency. Yao et al. [1] optimized the sampling of the starting and ending trees in the Bi-RRT* framework to improve the sampling efficiency of the Bi-RRT* algorithm in parking environments. Additionally, the straight-line connection between two points in the RRT is replaced with a Reed-Shepp curve connection, ensuring the feasibility of the path.

Yao adopted the approach from literature [3] and proposed a fast collision detection model for vehicle path planning. Although this method may sacrifice some precision, it can quickly detect vehicle collisions at individual points. However, this method does not address the collision optimization of paths. In planning, each collision detection typically involves a section of paths rather than a single point. To address this issue, this paper proposes a path collision detection method based on a binary hierarchical sequence. This method improves the efficiency of collision detection by adjusting the sequence of single-point collision detections along the path.

Simultaneously, although Yao employs the obstacle-avoidance sampling method for the starting point search tree, the feasible sampling points obtained by this method are predominantly located in the middle of the road, which may result in collisions with the road and an increased number of gear shifts for path switching. To address this issue, this paper proposes a road edge sampling method. The road edge sampling rule effectively utilizes lane space for vehicles and reduces the number of forward and backward movements as well as the path length.

III. Method

Bi-RRT* was developed for robot path planning and could not satisfy the nonholonomic motion constraints of vehicles. By modifying the Bi-RRT* algorithm to connect points using RS curves instead of straight lines, the planned paths can satisfy the vehicle's kinematic constraints. Moreover, the most time-consuming aspect of path planning is the calculation of collision detection. In limited, narrow parking scenarios, and with known basic road information, random sampling will generate a large number of invalid collision sample points. The occurrence of these situations will reduce the real-time performance of parking planning and increase the time consumption.

A. Collision detection with Binary hierarchical order

To solve the issue that RRT cannot generate paths with kinematic constraints, many studies have proposed different methods for generating connecting paths between RRT sampling points. Some methods use cubic Bézier curves [4], others use genetic algorithms to optimize the path [5], and some use RS curves to connect sampling points [1]. Regardless of the method, they will all generate an unpredictable collision-free path between two known collision-free sampling points. For connections with straight lines or fixed-rule curves, bounding boxes can be used for rough collision detection. However, after passing the rough collision detection, it is still necessary to perform fine collision detection for each individual point.

Path collision detection with the normal sequence will check collision detection from the starting point until the last point is detected or a collision is found. However, due to the complexity of the scene, the step size of the generated path is often set small in order to ensure that will not appear two points is collision free but path directly cross through the obstacles. However, a smaller step size will result in more collision detection points, which in turn will lead to a more time-consuming collision detection for the entire path.

To solve this problem, this paper proposes a path collision detection method based on binary hierarchical sequence. The idea of this method comes from binary tree hierarchy sorting. Suppose that the subscript [0...N] of the waypoint is used to construct a balanced binary tree, in which the left subtree of each node is less than the value of the parent node, and the right subtree is greater than the value of the parent node. Meanwhile, the absolute value of distance between the left and right subtrees of each node and the value of the node is minimized, that is, the average value, which means that this node is the node closest to the left and right subtrees under this distance restriction. For example, use [0...N] builds a binary tree, the root node is N/2, the left subtree is [0...N/2-1], the right subtree is [N/2+1...N], and then the binary tree can be built by analogy. At this time, the binary tree can be sorted and then accessed through hierarchical traversal, so that most nodes in the tree can be accessed at the fastest speed, and the nodes that are not accessed also have their closest parent nodes. The binary hierarchical sequence realized by algorithm 1 can be obtained directly without the construction of binary tree on the basis of the specified number of segmentation.

Algorithm 1 calculate Binary Hierarchical Sequence

```
N ← pathlength
orders ← []
orderList ← [[0, 1]]
pointheader ← 0
pointtail ← 1
while pointheader < pointtail do
    head ← orderList[pointheader][0]
    tail ← orderList[pointheader][1]
    half ← (head+tail)/2
    orders.append(half)
  if len(orders) > N then
    return orders * N
  end if
    orderList.append([head, half])
    orderList.append([half, tail])
    pointtail ← pointtail + 2
    pointheader ← pointheader + 1
end while
return orders * N
```

B. Road limit collision detection

The sampling points of the original Bi-RRT* algorithm are generated based on uniform sampling of search space, which is easy to generate many useless sampling points in a relatively fixed parking scene. For fixed parking scenes, when road information is known, different sampling methods can be modified to guide the growth of Bi-RRT* trees. Yao's approach uses an obstacle-avoidance sampling method for starting search trees growing on the road, placing the sampling points more in the middle of the lane. Radial sampling is used in the search tree for the parking end point in a narrow space, so that the sampling points are distributed near the end point in line with the position of vehicle kinematic constraint movement.

Although obstacle avoidance sampling utilizes road information, most of the sampling points are distributed in the middle of the road and have a small Angle with the road direction, which cannot make full use of road space. By using road information and vehicle model information for sampling, the road edge sampling rule can generate good collision-free sampling points, and make full use of lane space to generate sampling points with a large difference between the Angle and the road direction.

The road edge sampling method is based on lane information. For lanes with a long length (l) and a wide width (w), the sampling point often does not collide in the direction of lane extension. Therefore, the x-value of the sampling point is uniformly sampled from [0, l],$x \sim \mathcal{U}(0, l)$. During sampling, it is expected that the sampling point can fully cover all angles, so the yaw value of the sampling point is uniformly sampled from [0, 360],$yaw \sim \mathcal{U}(0, 360)$. When vehicles approach one side of the road, they tend not to collide on the other side. At this time, the main factor of collision is the vehicle angle relative to the lane when the vehicle is near one lane. This angle affects the minimum distance of the vehicle from the curb.

The y-value of the sampling point is sampled between the upper and lower boundaries of the road edge, so both sides of the road are selected with the same probability. When the sampling point is at the upper curb, the y-value is equal to the upper curb y-value minus the minimum distance from the curb minus $\mathcal{N}(0, 1)$*w/2. Similarly, when the sampling point is at the lower curb, the y-value of the sampling point is equal to the y-value at the lower curb plus the minimum distance from the curb plus $\mathcal{N}(0, 1)$*w/2. Here, the normal distribution of random variables is used to sample vehicles from the road edge to the middle of the road, so that sampling points are distributed on both sides of the road to make full use of the road space.

The values y of the sampling point are shown in formula (1). Assume the vehicle's rear axle center coordinates are (x, y, θ), with the vehicle length cl, width cw, and wheelbase wl. The vehicle has equal front and rear overhangs. The distance from the vehicle's rear axle center to the front corner point is l_h, and the distance from the rear axle center to the vehicle's rear corner point is l_t.

$$y = \begin{cases} y_{upper} - y_{limit} - \frac{w}{2} * \mathcal{N}(0,1) & \text{if } chooseuppercurb \\ y_{lower} + y_{limit} + \frac{w}{2} * \mathcal{N}(0,1) & \text{if } chooselowercurb \end{cases} \tag{1}$$

$$l_h = \sqrt{(\frac{l}{2} + \frac{wl}{2})^2 + \frac{w^2}{4}} \tag{2}$$

$$l_t = \sqrt{\left(\frac{l}{2} - \frac{wl}{2}\right)^2 + \frac{w^2}{4}} \tag{3}$$

$$y_{limit} = \begin{cases} l_t \cdot \sin(\theta + \pi + \operatorname{atan}(\frac{w}{l+w_l})) & \text{if } \theta \in (-\pi, -\frac{\pi}{2}) \\ l_t \cdot \sin(\theta + \pi - \operatorname{atan}(\frac{w}{l+w_l})) & \text{if } \theta \in (-\frac{\pi}{2}, 0) \\ l_h \cdot \sin(\theta + \operatorname{atan}(\frac{w}{l+w_l})) & \text{if } \theta \in (0, \frac{\pi}{2}) \\ l_h \cdot \sin(\theta - \operatorname{atan}(\frac{w}{l+w_l})) & \text{if } \theta \in (\frac{\pi}{2}, \pi) \end{cases} \tag{4}$$

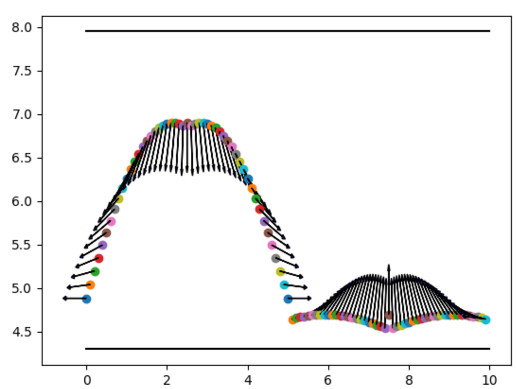

图 1. When the curb y=4.3 is selected as the lower curb, the minimum distance between the vehicle and the lower curb at different angles,from left to right is π to $-\pi$

IV. Result

The improved algorithm of Bi-RRT* is use and simulated on python3. In order to verify the effectiveness of the method presented in this paper, other conditions were tested one by one while other conditions were controlled to be consistent. To ensure the accuracy of planning results, the average of 200 planning results was taken as the final experimental result.

A. Collision detection with Binary hierarchical order

In the parking scene, the vehicle length is 3m, the vehicle width is 1.2m, the vehicle wheelbase is 2.2m, the vehicle turning radius is 1m, the initial point coordinate is (2.5, 7,0), the end point coordinate is (3, 2, pi/2), the map resolution is 0.1m, the search step is 0.1m, and the map length and width are 10m. The search method of the starting search tree is random sampling, and the search method of the end search tree is radial sampling. The height of the lower lane is 4.3m, the width of the lane is 3.65m and the width of the parking space is 1.9m. Stop searching when the number of samples is greater than 5000 or when a viable path is found. Test data were averaged from 200 experiments.

In the easy map,ordinary map and complex map, the vehicle length is 3m, the vehicle width is 1.2m, the vehicle wheelbase is 2.2m, the vehicle turning radius is 1m, the initial point coordinate is (5, 5,0), the end point coordinate is (23, 25, 0) which is center of the depression region, the map resolution is 0.1m, the search step is 0.1m, and the map length and width are 50m. The search method of the starting search tree is random sampling, and the search method of the end search tree is radial sampling. Stop searching when the number of samples is greater than 5000 or when a viable path is found. Test data were averaged from 200 experiments.

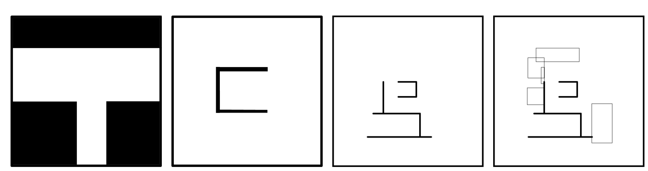

图 2. some case of test map. from left to right are parking map, easy map, ordinary map and complex map

Table 1 is experimental results of normal sequential collision detection and binary hierarchical sequential collision in different scenarios under given experimental conditions.

表 I
EXPERIMENTAL RESULTS

map	evaluate method	normal order	Binary hierarchical order
parking map	run time	8.487s	6.428s
	average collision times	5.24	3.3
easy map	run time	11.344s	4.097s
	average collision times	47.61	11.92
ordinary map	run time	14.40m	18.70ms
	average collision times	8.06	6.12
complex map	run time	51.96ms	30.92m
	average collision times	8.06	6.38

Average collision times refers to the number of collision detection times required to detect such collision when the path is a collision path. Generally, this value increases as the step size decreases. It can be seen from table1 that whether the parking map or the simple to complex map, the collision detection of paths by using binary hierarchical sequence can reduce the time spent by reducing the number of collision detection. Binary hierarchical sequential path collision detection works well on empty maps with fewer obstacles, and can even reduce the average collision times found by 75% on easy Maps. On other maps, it was nearly 37% better.

B. Road limit collision detection

In the parking scene, the vehicle length is 3m, the vehicle width is 1.2m, the vehicle wheelbase is 2.2m, the vehicle turning radius is 1m, the initial point coordinate is (2.5, 7,0), the end point coordinate is (3, 2, pi/2), the map resolution is 0.1m, the search step is 0.1m, and the map length and width are 10m. The search method of the starting search tree is choose from random sample, Avoid sample, Road limit sample, Probabilistic road limit (80% random sample + 20% Road limit sample), and the search method of the end search tree is radial sampling. The height of the lower lane is 4.3m, the width of the lane is 3.65m and the width of the parking space is 1.9m. Stop searching when the number of samples is greater than 5000 or when a viable path is found. Test data were averaged from 200 experiments.

It can be seen from the table that the effect of using the road limit sampling method is better than that of avoiding sampling, but the final result is still worse than that of limiting the sampling range to even sampling on the road. However, after combining road information, the Probabilistic safety probability of the sampling points in the road limit sampling method increased compared with the uniform sampling method, and the probabilistic safety probability was also shown to be increased in the combined Road limit sampling method.

表 II
EXPERIMENTAL RESULTS

Sample method	Run time	sample point safety rate	expand number of node	path length
Random	10.491s	18.7%	17.06	7.123
Avoid	12.810s	12.0%	26.27	7.841
Road limit	9.691s	50.2%	24.16	7.183

V. CONCLUSION

Addressing the issues of time-consuming collision detection and low sampling efficiency in the Bi-RRT* algorithm for parking problems, this paper presents a two-level path detection sequence based on path collision detection sequence optimization. The simulation test results demonstrate that the improved path collision detection method outperforms the original method and significantly reduces the number of path collision detections in parking scenarios. Moreover, this method is more effective in maps with overall empty spaces but sparse obstacles.

In this paper, a road edge sampling method based on road information is proposed to enhance the sampling efficiency. This method shortens the calculation time and reduces the probability of sampling point collisions compared to the radial sampling method. Additionally, this method provides a higher safety probability for sampling points than uniform sampling.

参考文献

[1] 姚智龙, 张小俊, 王金刚. 改进 Bi-RRT* 算法的自动泊车路径规划 [J/OL]. 机械科学与技术:1-10[2023-05-03].https://doi.org/10.13433/j.cnki.1003-8728.20220310.

[2] Klemm S, Oberlander J, Hermann A, et al. RRT*-Connect: faster, asymptotically optimal motion planning[C]//2015 IEEE International Conference on Robotics and Biomimetics (ROBIO). Zhuhai: IEEE Press, 2015: 1670-1677

[3] Ziegler J, Stiller C. Fast Collision Checking for Intelligent Vehicle Motion Planning[C]//2010 IEEE Intelligent Vehicles Symposium. La Jolla: IEEE Press, 2010: 518-522

[4] Alejo, J. A. Cobano, G. Heredia, J. R. Martínez-De Dios, and A. Ollero, "Efficient trajectory planning for wsn data collection with multiple UAVs", in Cooperative robots and sensor networks. vol. 604, ed: Springer International Publishing, 2015, pp. 53-75.

[5] Lee, H. Song, and D. H. Shim, "Optimal path planning based on splineRRT* for fixed-wing UAVs operating in three-dimensional environments", presented at the 14th International Conference on Control, Automation and Systems (ICCAS 2014), Korea, 2014.