Advances in Artificial Intelligence, Big Data and Algorithms
G. Grigoras and P. Lorenz (Eds.)

doi:10.3233/FAIA230805

Research on Path Planning of Improved RRT Algorithm Under Narrow Channel

Yue LIU[a], Zhiwei ZHAO[a, b], Hong WANG[a], Chao GE[a1]
[a] *College of Electrical Engineering, North China University of Science and Technology, Tangshan Hebei 063210, China;*
[b] *Tangshan University, Tangshan Hebei 063000, China*

Abstract: In complex environments where narrow passageways exist, there are problems such as low success rate of path finding in the path planning of mobile robots based on the traditional fast expanding random tree (RRT) algorithm. To address the above problems, an RRT algorithm for autonomous narrow channel finding is proposed, using an entrance finding algorithm to identify the entrance of the channel and a bias strategy to rationalize the sampling point selection to improve the success rate of path planning. In addition, a greedy algorithm is introduced to optimize the initial path and improve the quality of the planned path. Through experiments, it is shown that the proposed algorithm improves 59.6%, 56%, 9.3%, and 40% in four aspects compared with the RRT algorithm with bias in narrow channel environment in terms of iteration time, number of iterations, path length, and path planning success rate, respectively.

Key words: path planning; rapidly expanding random tree; entry detection algorithm; greedy algorithm

1. Introduction

With the development of science and technology, the research and application of mobile robots has become more and more in-depth and extensive, in which the path planning algorithm is the basis for the implementation of mobile robots[1].After the development in recent years, scholars at home and abroad have proposed numerous intelligent algorithms，These include Artificial potential field algorithm[2], Neural network algorithm[3], Ant colony algorithm[4]etc. However, these algorithms require pre-modeling of the space，As the complexity of the environment increases and the degree of freedom of the robot increases, its computation will increase significantly, and the efficiency and real-time of planning will be greatly reduced, so these algorithms are not suitable for complex environments.

To address the above issues, SM La Valle Proposed fast extended randomized tree (RRT) path planning algorithm[5]. The algorithm does not need to pre-model the space, by selecting randomly sampled points in the space to keep iterating, and then explore the free space until the target point is explored. Compared with previous traditional algorithms, the RRT algorithm is efficient and fast, especially in high-dimensional space performance has been greatly improved[6].The RRT algorithm has attracted the

[1] Corresponding Author: Chao GE, *North China University of Science and Technology;*
Emil: gechao365@126.com

research interest of a wide range of scholars, and many scholars have proposed improved algorithms of RRT [7-11].

Although the RRT algorithm has many advantages in path planning, it will face great challenges in spaces with narrow aisles. The presence of narrow channels will make the effective sampling points in the channel drop sharply. For this reason, Dong Cheng Mo et.al.[12] proposed the combination of RRT-Connect algorithm and bridge testing to solve the problem of difficult sampling in narrow channels, although this approach can obtain the sampling points in the channel by bridge testing, the problem of low success rate of resampling in narrow channels will occur when the sampling space becomes large. On this basis, JuPen Fu et.al. [13] proposed to improve the bridge test sampling efficiency problem by adding obstacle edge detection, although this approach can increase the efficiency of sampling in narrow channels, it requires pre-processing of the sampling space, which increases the complexity of the algorithm and its algorithmic efficiency is reduced. Furthermore, in the paper [14], the authors propose a Q-Learning based partitioned heuristic RRT planning algorithm (Q-PRRT) to solve the problem of sampling difficulties in complex environments.

In view of the above problems, in this paper, inspired by the literature [15], we propose a hybrid RRT algorithm (Mix-RRT) based on the Bias-RRT algorithm for maps with narrow channels. This algorithm aims to identify the entrance of narrow passages quickly and autonomously when the random tree is expanded to facilitate the smooth passage of the random tree through the narrow passages, mainly by range detection of new nodes in the expanded tree to identify the entrance of narrow passages and improve the ability of the algorithm to identify narrow passages. In addition, concave spaces are easy to appear in narrow passages, and the expansion tree of the basic RRT algorithm is easy to fall into concave spaces when expanding, so this algorithm adds a forward detection algorithm to avoid concave traps and make the expansion tree pass through narrow passages smoothly. Finally, this algorithm optimizes the sampling efficiency of sampling points and improves the quality of planned paths by incorporating strategies such as sampling range restriction and greedy algorithm. By comparing with Bias-RRT algorithm, the iteration time, number of iterations, path length, and path planning success rate are improved by 59.6%, 56%, 9.3%, and 40%, respectively.

2. RRT Algorithm

The RRT algorithm is a fast and efficient algorithm, similar to the way a tree grows and spreads into free space. The RRT algorithm is extended from the initial node, which is the root node of the random tree. By connecting random points to expand outward, the random tree also keeps expanding toward the target region as the number of iterations increases until it reaches the target region, thus producing a feasible planning path from the starting point to the target point. The specific expansion process of the random tree is shown in Fig.1.

As can be seen in Figure 1, the starting point Q_{init} of the random tree is used as the initial root node. Generating a random point Q_{rand} in the first iteration. , at this time Q_{near} is the nearest node to the random point in the tree, Connect point Q_{near} as well as point Q_{rand} ,Intercept a point of length l in the direction from point Q_{near} to point Q_{rand}

and denote it as point Q_{new}.By determining whether there is an obstacle on the line connecting point Q_{near} and point Q_{new}.If the connection crosses the barrier area, this extension fails and the next iteration is opened; Instead, this extension succeeds and adds the point Q_{new} to the random tree as a node in the tree. By continuously iterating until the target area is extended.

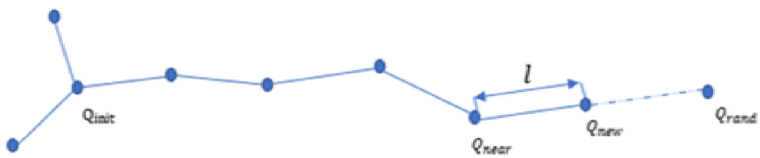


Figure 1 Specific RRT algorithm node extensions.

3. Algorithm of this paper

3.1 Narrow Passage Optimization Treatment

In complex environments with narrow passageways, the probability of a random tree getting "lost" in the complex environment is greatly increased, and it is difficult for the basic RRT algorithm to identify narrow passageways and reach the target point through them. This is because the basic RRT algorithm is globally uniformly sampled, and the probability of a sample point falling at the entrance of a narrow channel is low, making it difficult to extend the random tree to narrow channels. To solve the above problems, this paper proposes a narrow passage entrance recognition algorithm. In the process of random tree expansion, if a new expansion point is added to the random tree, an entrance detection is performed at this point, and if an entrance is found, the random tree is guided to expand in favor of the entrance, and vice versa, the normal iteration is performed. This allows the random tree to "consciously" identify the entrance to the narrow channel, so as to smoothly expand into the narrow channel and avoid falling into a dead loop, as shown in Fig.2.

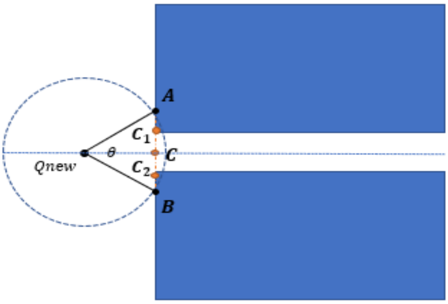


Figure 2 Schematic diagram of improved entrance identification.

As shown in Figure 2, the detection circle of radius r is first generated with Q_{new} as the centre, and two rays of angle θ and length r are extended with the centre point. Calculate the radius of the detection circle according to the formula(1):

$$r = k_1 \times X + k_2 \times Y \tag{1}$$

Where X is the length of the map, Y is the width of the point map, k_1 and k_2 are the scale factors of the two lengths, which can be set according to the size of the map, which can be set according to the size of the map, but generally ensure that the radius r of the detection circle is larger than the maximum width of the narrow passage. When there is a narrow passage entrance, the ray will produce intersection $A(x_a, y_a)$ and intersection $B(x_b, y_b)$ with the obstacle respectively. the ray will intersect with the obstacle respectively $A(x_a, y_a)$ and the intersection point $B(x_b, y_b)$ .Connection Points A with point B ,We divide the line segment $\overline{AB}$ into four equal parts,This produces three equal points $C(x_c, y_c)$ $C_1(x_1, y_1)$ $C_2(x_2, y_2)$ on the line segment $\overline{AB}$ as shown in Figure 2,As can be seen from Figure 2, the collision detection of the three points, the three points C_1 and C_2 points in the obstacle, C points in the free space, then we can initially determine the existence of free space between the line segment AB, However, in a complex environment obstacles can exist concave traps, and to solve this problem, a forward detection approach is introduced, as shown in Fig. 3.

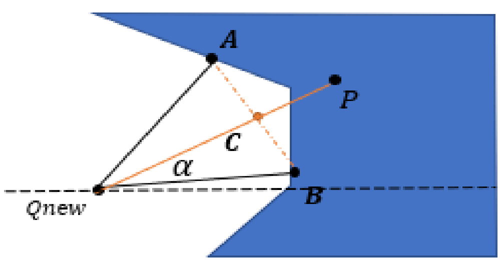


Figure 3 Schematic diagram of forward detection.

In Fig. 3, we extend the length of $r/2$ in the direction of $\overline{Q_{new}C}$ to get a point P Collision detection is performed at this point, and if an obstacle appears, it means that the extension here is caught in a concave trap. So stop this direction and get out of the trap here.

The forward detection point $P(x_p, y_p)$ can be expressed as:

$$x_p = (x_c + \frac{r}{2}) \tag{2}$$

$$y_p = (x_c + \frac{r}{2}) \tan\alpha \tag{3}$$

where α is the angle between point C and the horizontal axis, and is the length of the radius of the detection circle.Therefore, the presence of a narrow passage entrance can be indicated even if the point C is in a free position and the forward detection is also valid.After that, point P is set as the temporary target point, and the random tree expands to the temporary target point at this time, so that the random tree finds the entrance of the narrow passage and expands.

3.2 Improved Probabilistic Target Point Selection Strategy

When a random point Q_{rand} is being obtained, the basic RRT algorithm is selected randomly in the free space, which will consume a lot of computation time on random points of little significance. The heuristic Bias-RRT algorithm based on the literature [16] introduces a random threshold ρ_1 as well as a random number $\rho < (0,1)$ .If $\rho_1 > \rho$, the point Q_{rand} will be chosen randomly; conversely if $\rho_1 < \rho$, the target point Q_{goal} will be set as random point Q_{rand}, that is $Q_{\text{rand}}=Q_{goal}$ .Introducing range restrictions on top of this will better guide the random tree towards the target point, reducing the number of iterations and the generation of useless points.

In this paper, we introduce a maximum distance limit by setting a new threshold $S_{\max}$, where the maximum length from the root start point Q_{init} to the target point Q_{goal} is $L_{\max}$ .The average value of the convergence speed was obtained by several tests at a threshold value of 1.3 times the length of the maximum value, which was greater. Therefore, the threshold $S_{\max}$ is set to 1.3 times the maximum length, $S_{\max} = 1.3L_{\max}$ .In addition, Q_{rand} is the sum of the Euclidean distances from the random point to the target point and the initial point. If S_t is greater than the $S_{\max}$ threshold, it is decided that the random point is outside the limited range and the point is excluded. The opposite can be done for random tree expansion. This eliminates useless points that increase the path length, streamlines the structure of the random tree and speeds up the planning. The sum of Euclidean distances S_t from the random point Q_{rand} to the target point and the initial point is calculated according to equation(4):

$$S_t = \sqrt{(x_t - x_{init})^2 + (y_t - y_{init})^2} + \sqrt{(x_t - x_{goal})^2 + (y_t - y_{goal})^2} \le S_{\max} \tag{4}$$

$$Q_{rand} = (x_t - y_t), Q_{init} = (x_{init}, y_{init}), Q_{goal} = (x_{goal}, y_{goal}) \tag{5}$$

3.3 Greedy Algorithm to Optimize Route Processing

A path can be found after the optimization of the above algorithm, but because the basic RRT algorithm selects sampling points for expansion, so that the path will be a zigzag route, which increases the length of the path. Therefore, this paper uses the greedy algorithm [17] to remove the redundant points in the path and reduce the path length to make its path quality higher. The process is shown in Figure 4.

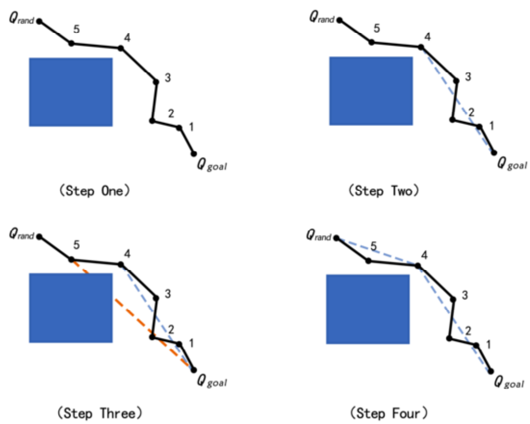


Figure 4 Diagram of greedy algorithm.

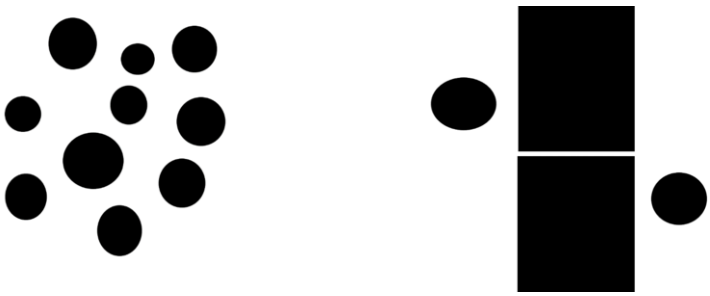
From Fig. 4, it can be seen that the (a) diagram is the path found by the basic RRT algorithm, which has too many twists and turns and is not the optimal path. In the figure (b), let the target point Q_{goal} connects point 1, point 2, point 3, and point 4, respectively, and perform collision detection on these four connected segments. Obviously, until the connection to point 4 is free. In the figure (c), the target point Q_{goal} connects to point 5 when the connected line segment passes through the obstacle. Therefore, the target point A can be directly connected to point 4, replacing the original path as the optimized path while points 1, 2, and 3 can be discarded. Then the above iteration is carried out with point 4 as the next temporary target point until it connects to the starting point Q_{init} .The dashed line as shown in (d) is the optimized path. It can be seen that the optimized path is significantly better than the original path.

4. Simulation Verification

In order to verify the effectiveness of the algorithm proposed in this paper, simulation experiments are conducted to validate the basic RRT algorithm, the biased RRT algorithm and the optimized RRT algorithm proposed in this paper for comparison experiments.

4.1 Map Building

As shown in Fig. 5(a) and (b), the map for this comparison test, the black part of which is the area where the obstacle is located, the white area is the free area that can be expanded, and the map size is 800×800.The specific parameters are set as (1,1) for the initial point, (700,700) for the target point, 15 for the step size, and 5000 for the maximum number of iterations. The original RRT algorithm, the RRT algorithm with bias, and the improved algorithm in this paper are run for 100 experimental simulations, and compared by counting the average values of four factors: running time, number of iterations, path length, and path planning success rate in the three algorithms.

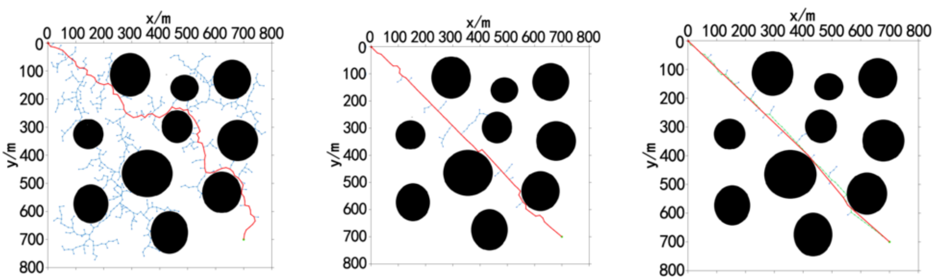


(a)Simple Map (b) Narrow passage map

Figure 5 Map of the experiment.

4.2 Simple Map Analysis

In this experiment, the results of the simple map experiment are shown in Figure 6, and the results of the comparison experiment are shown in Table 1.



（a）Original RRT algorithm （b）Biased RRT algorithm （c）Algorithm of this paper

Figure 6 Results of the simple map experiment

Table 1 Results of the simple map comparison experiment

Algorithm	Running time/s	Number of iterations/times	Path length/m	Success rate/%
RRT	45	1381	1136.8921	96%
Bias-RRT	12	835	1021.6624	98%
Mix-RRT	11	816	1006.2930	98%

（1）In terms of planning time, it can be seen from Table 1 that in this setting, the algorithm in this paper reduces 75.6% compared to the original RRT algorithm. Compared with the RRT algorithm with bias, the two are basically equal. This is reflected in the fact that the planning efficiency of this paper's algorithm in simple maps is about the same as that of the RRT algorithm with bias.

（2）In terms of the number of iterations, the number of iterations of the algorithm in this paper is reduced by 40.9% and 2.3% compared to the original RRT algorithm and the RRT with bias algorithm, respectively. In this regard, the number of iterations of this algorithm becomes less and the efficiency of the algorithm is improved accordingly.

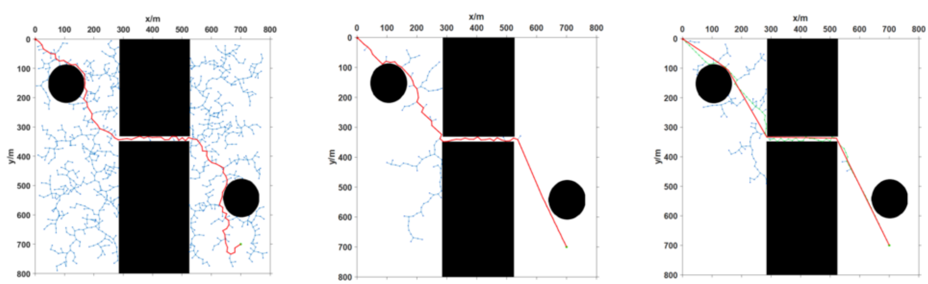
（3）In terms of path length, the path length of the algorithm in this paper is reduced by 130.5991 m and 15.3694 m compared with the original RRT algorithm and the RRT with bias algorithm, respectively. the greedy algorithm is introduced in this paper to optimize the path, which reduces the redundancy points and improves the path quality.

（4）In terms of success rate, it can be seen from Table 1 that the corresponding success rate is lower than that of the algorithm in this paper and the RRT with bias algorithm due to the poor targetability of the original RRT algorithm. However, all three algorithms maintain a success rate of more than 95%. This indicates that all three algorithms can find the corresponding planning paths with sufficient number of iterations at simple maps.

（5）As can be seen from Fig. 6, in the simple map case, the original RRT algorithm can find the planning path, but there are more redundant points compared with the RRT with bias algorithm and the algorithm in this paper. The RRT with bias algorithm reduces the redundancy points accordingly, but there is still a situation that the planned path is not ideal. The algorithm in this paper plans paths with fewer redundancy points, straighter paths, and higher path quality.

4.3 Narrow Passage Map Simulation Analysis

In this experiment, the narrow channel environment is added to verify the applicability of the algorithm of this paper in the special environment, and the experimental results are shown in Figure 7, and the comparison results of the three algorithms are shown in Table 2.



（a）Original RRT algorithm （b）Biased RRT algorithm （c）Algorithm of this paper

Figure 7 Experimental results of the narrow passage map

Table 2 Results of the narrow channel comparison experiment

Algorithm	Running time/s	Number of iterations/times	Path length/m	Success rate/%
RRT	152	4581	1472.5328	23%
Bias-RRT	52	2321	1252.7324	65%
Mix-RRT	21	1021	1135.9580	91%

(1) In terms of planning time, the algorithm in this paper reduces 86.2% compared to the original RRT algorithm. The reduction is 59.6% compared to the RRT with bias algorithm. In terms of planning time, the planning time of the algorithm in this paper is substantially reduced.

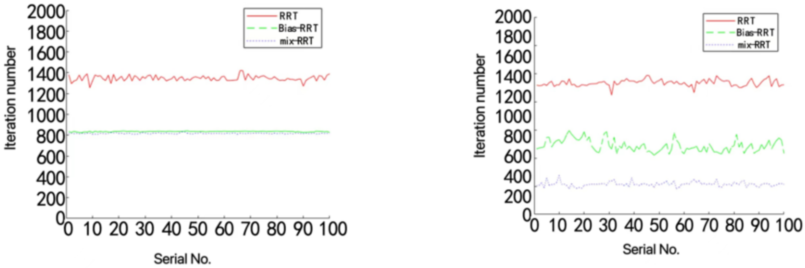
(2) In terms of the number of iterations, the iteration of the algorithm in this paper is reduced by 77.7% and 56.0% this time compared to the original RRT algorithm and the RRT with bias algorithm, respectively. In this regard, the number of iterations of this paper's algorithm is significantly reduced due to its ability to identify narrow channels.

(3) In terms of path length, the path length of the algorithm in this paper is reduced by 336.5748 m and 116.7744 m compared to the original RRT algorithm and the RRT with bias algorithm, respectively. in this environment, the original RRT algorithm and the RRT with bias algorithm have a corresponding increase in the planned path length due to the increase in complexity of the environment.In this paper, greedy algorithm is introduced in the algorithm to optimize the path, which reduces the path length and improves the path quality.

(4) In terms of success rate, the success rate of the original RRT algorithm is only 23%, the RRT with bias algorithm is 65%, and the algorithm in this paper is 91%.It is known that the algorithm of this paper can be better applied in complex environments, especially in the presence of narrow passages.

(5) As can be seen from Fig. 7, the path planning capability of the original RRT algorithm will be significantly reduced in the case of narrow channel maps, which will easily lead to a dead loop of iterations without finding the entrance to the channel, making the path planning fail. For the RRT algorithm with bias, its planning capability is on par with the algorithm in this paper in the simple map case, but its planning capability decreases significantly in the complex environment, which is affected by the target bias, making it easy to skip narrow channel entrances, resulting in planning failure and reduced success rate. Compared with the previous two algorithms, this algorithm can better identify narrow passage entrances and avoid getting into dead loops. In addition, route optimization processing is added to make the planned paths tend to be optimal.

4.4 Data Analysis



(a) (b)

Figure 8 Statistics on the number of iterations

The statistics of the number of experimental iterations are shown in Fig. 8. From Fig. 8(a), it can be seen that in the case of simple maps, the number of iterations of the RRT algorithm is higher compared to the Bias-RRT algorithm and the algorithm in this paper, and the number of iterations of the Bias-RRT and the algorithm in this paper is approximately the same.It can be seen from Fig. 8(b). The number of iterations of the RRT algorithm increases significantly in the environment where narrow channels exist, and a significant gap also appears between the number of iterations of the Bias-RRT algorithm and the algorithm in this paper.

5. Conclusion

The algorithm in this paper improvesp the planning success rate of this paper compared to the traditional RRT algorithm and the Bias-RRT algorithm in environments with narrow passages by adding three optimizations to the Bias-RRT algorithm, including narrow passage entrance identification strategy, improved probabilistic target point selection strategy, and greedy algorithm optimized route processing. Through experimental comparison, the planning success rate of the algorithm in this paper is 68 and 26 percentage points better than that of conventional RRT and Bias-RRT, respectively. To a large extent, it shows that the proposed algorithm in this paper is suitable for the environment where narrow channels exist.

Acknowledgements

This work was. Supported by the Hebei Provincial Natural Science Foundation (F202109006); Hebei Provincial Natural Science Foundation-Steel Joint Research Fund Grant (E2019105123); Hebei Provincial Higher Education Science and Technology Research Project Grant (ZD2019311) The authors gratefully acknowledge these grants.

References

[1] A. Gasparetto,P. Boscariol,A. Lanzutti,and R. Vidoni.Path planning and trajectory planning algori thms: A general overview[J].Motion and operation planning of robotic systems,2015.
[2] Szeto-Hua-Jie, H.B. Lei, and C.G. Zhuang. RRT path planning algorithm based on artificial pot ential field guidance in dynamic environment [J]. Computer Application Research,2021,38(03):71 4-717+724.
[3] Y.Zhang, M.L. Zhang, Z.H. Jiang,and L.V. Xiao. Research on path planning of mobile robot in dynamic environment [J]. Journal of hefei university of technology (natural science),2020,43(1 0):1297-1306.
[4] J.K.Bai and X.N.Wu. Robot Path Planning Algorithm based on Ant Colony Random Tree [J]. Machinery & Electronics,2015(07):73-76.
[5] S.M. LaValle.Rapidly-exploring random trees: A new tool for path planning[R]. Iowa State Uni versity,1998.
[6] Q.L.Chen, H.Y.Jiang,Y.J.Zheng. Review of fast extended random tree algorithm for robot path p lanning [J]. Computer engineering and applications,2019,55(16):10-17.
[7] S.M. LaValle,J.J. Kuffner, B.R. Donald. Rapidly-exploring random trees: Progress and prospects [J]. Algorithmic and computational robotics: new directions, 2001, 5: 293-308.

[8] J.J. Kuffner and S.M. LaValle. RRT-connect: An efficient approach to single-query path plannin g[C].Proceedings 2000 ICRA. Millennium Conference. IEEE International Proceedings (Cat. No. 00CH37065). IEEE, 2000, 2: 995-1001.
[9] Q.J.Zou, S.H.Liu, Y.Zhang,and Y.L.Hou.A fast path reprogramming algorithm based on Reinforc ement Learning for Exploring random Tree In Special Environment [J]. Control theory & applic ations,2020,37(08):1737-1748.
[10] H.D.Zhang, Y.Chen, Y.X.Wu. Real-time path planning for mobile robot in unknown environment [J].Computer engineering and applications,2018,54(19):140-146.
[11] E.H.Liu, W.B.Gao, R.P.Kong, B.Y.Liu, Y.Dong,and Y.Y.Chen. Improved RRT path planning algo rithm [J]. Computer engineering and design,2019,40(08):2253-2258.
[12] D.C.Mo, G.D.Liu. Improved rt-connect path planning algorithm for biped robot [J]. Journal of c omputer applications,2013,33(08):2289-2292.
[13] J.P.Fu, G.H.Zeng, B.Huang,and Z.J.Fang.Journal of computer applications,2019,39(10):2865-2869.
[14] J.D.Zhong. Research on Robot Path Planning Based on Narrow Channel Identification[D].Shang hai Jiaotong University,2012.
[15] Z. Liu, F. Lan, H. Yang. Partition Heuristic RRT Algorithm of Path Planning Based on Q-learn ing[C].In IEEE 4th Advanced Information Technology, Electronic and Automation Control Confe rence (IAEAC). IEEE, 2019, 1: 386-392.
[16] Y. Ying, Z. Li, G. Ruihong, et al. Path planning of mobile robot based on Improved RRT Alg orithm[C].Chinese Automation Congress (CAC). IEEE, 2019: 4741-4746.
[17] E. Taheri, M. H. Ferdowsi, M. Danesh. Fuzzy greedy RRT path planning algorithm in a compl ex configuration space[J]. International Journal of Control, Automation and Systems, 2018, 16 (6): 3026-3035.