ISBN: 978-981-18-1791-5
2021 the 11th International Workshop on Computer Science and Engineering (WCSE 2021)
doi: 10.18178/wcse.2021.06.020

Improved RRT Algorithm Path Planning Combined with Artificial Potential Field Algorithm

Xiufen Wang[1+], Shengyi Yang [2]

[1] School of Mechanical Electronic Engineering, Guizhou Minzu University, Guiyang, Guizhou 550025, China

[2] Key Laboratory of Pattern Recognition and Intelligent Systems of Guizhou Province, Guizhou Minzu University, Guiyang, Guizhou 550025, China

Abstract. In order to solve the problem of low path planning efficiency in complex obstacle environment, an improved Rapidly-exploring Random Trees (RRT) algorithm is proposed. It utilizes artificial potential field to guide the fast expanding random tree to growing towards the target and away from obstacles. Considering the gravitational field of the target node and the repulsion field of the obstacle in the artificial potential field, a certain number of random nodes generated in one iteration are evaluated and selected. The simulation results show that the improved RRT algorithm has strong advantages over the basic RRT algorithm in search ability and computation time.

Keywords: RRT, path planning, artificial potential field, randomness.

1. Introduction

The goal of path planning is to plan an optimal or suboptimal obstacle avoidance path that connects the starting point and the target point and satisfies the constraint conditions of the robot itself in the working environment with obstacles [1]. At present, common path planning methods mainly include A* algorithm, artificial potential field method, neural network, and rapidly expanding random tree algorithm RRT[1-2]. Among them, RRT algorithm has flexible and powerful search ability and can be used for path planning in A variety of complex environments [3]. However, in the search process, the comprehensive cost of path is not considered. Due to the randomness of nodes in expansion, the planned path has strong randomness and the obtained path is not optimal. In fact, the optimal path planning algorithm is particularly important for the practical application of robots [4].

In view of the deficiencies of RRT algorithm, two-way search is adopted to speed up the solution process [5]. In Literature [6], K nodes closest to the target point were selected from the existing search tree during node expansion, and the evaluation function was used to guide node expansion. In reference [7], on the basis of reference [6], multiple random sampling strategy was adopted to reduce the influence of randomness of RRT method. Literature [8] improved the random sampling strategy by dynamically adjusting the sampling area of random points, and improved the node expansion efficiency of the RRT method.

In order to overcome the above problems, combined with the idea of artificial potential field, this paper makes use of the gravitational effect of the target point and the repulsive effect of the obstacle to make the random tree closer to the target and away from the obstacle. The simulation experiment shows that, compared with the basic RRT algorithm, the improved algorithm has enhanced the adaptability to the environment and improved the search efficiency.

2. Basic RRT Algorithm

2.1. The principle of Basic RRT Algorithm

+ Corresponding author. Tel.: 18786023176.
E-mail address:xiufen_wang@yeah.net.

The main idea of the basic RRT algorithm is to generate a random extension tree gradually through random sampling. As the tree grows until the target points are included, then the path formed by connecting the nodes between the root node of the random tree and the target points is the planned path. The growth process of the random tree is shown in Figure 1. Set the starting point to the root of the random tree. From this root node, the new leaves are extended outward. The way of extension includes random sampling, step size restriction, and collision detection. In each iteration, RRT will generate a random sampling Node. If the Node is located in the free region, then we can traverse all the existing nodes in the random tree and find the ClosestNode nearest to the Node. The distance between the two nodes is obtained by using the distance function. If the step size limit is satisfied, we will then carry out collision detection on the two nodes. If the step size limit is not satisfied, we need to find an intermediate point along the direction of the connection between the two points that meets the step size limit to replace the Node. Finally, if Node and ClosestNode pass collision detection, Node can be added to the random tree. The final planned path is shown in the green line in the Fig. 1.

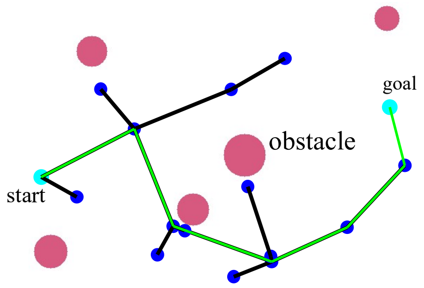


Fig. 1: RRT algorithm basic principle diagram.

2.2. Shortcomings of the Basic RRT Algorithm

(1) Strong randomness and lack of target directivity in the search process. RRT algorithm is based on random sampling algorithm, the growth of random tree nodes is random, which leads to the random tree growth has no direction, and the calculation is large.

(2) Planning path may not be optimal. There is no comprehensive consideration in the search process, and the path obtained is not always optimal

3. Improved RRT Algorithm

On the one hand, due to the disadvantages of the basic RRT algorithm, such as strong randomness and no target directionality in search, the idea of target point generating gravity in artificial potential field algorithm is introduced into the RRT algorithm. On the other hand, in order to better avoid the interference of obstacles, improve the search efficiency, reduce the amount of computation and reduce the time cost, the repulsive effect of obstacles in the artificial potential field is introduced into the RRT algorithm.

For n nodes randomly generated in each iteration, the selection of new nodes is determined by comprehensive analysis of their gravitational and repulsive forces, as shown in Equation (1).

$$f(P_i) = \frac{dis(P_i, goal)}{(\min dis(P_i, obstale))^{\alpha}} \tag{1}$$

Where, P_i is the generated random node, $dis(P_i, goal)$ is the distance between P_i and the goal point, $\min dis(P_i, obstale)$ is the minimum distance between P_i and obstacles, and α is the corresponding impact factor. Of course, according to the actual needs, we can change the relative influence degree of the target point and the obstacle by adjusting the α value. It can be seen that the smaller the $f(P_i)$ value, the better the superiority of the corresponding point.

As shown in Fig. 2, the current node is P, and three nodes P_1, P_2 and P_3 are randomly generated. Therefore, the applicability of the three subsequent nodes needs to be evaluated. After calculation, the corresponding values of the three nodes are $f(P_1)=92.90$, $f(P_2)=84.29$, $f(P_3)=99.84$ respectively, where α=0.3 .Therefore, P_2 is taken as the optimal selection point.

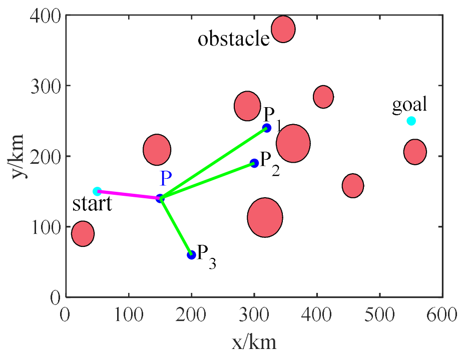


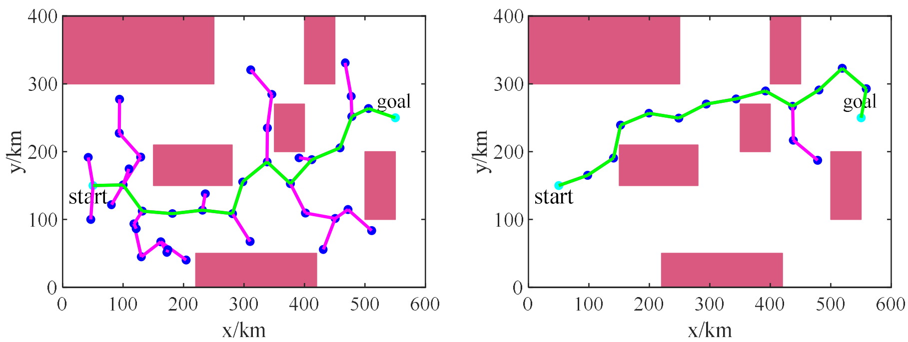
Fig. 2: Selection of node.

4. Experiment and Analysis

Based on the algorithm designed above, the algorithm simulation experiment is carried out on the Intel(R) Core(TM)i5-10210U CPU@1.6GHZ 2.11GHz CPU and 12.0GB memory laptop computer using MATLAB R2018b.

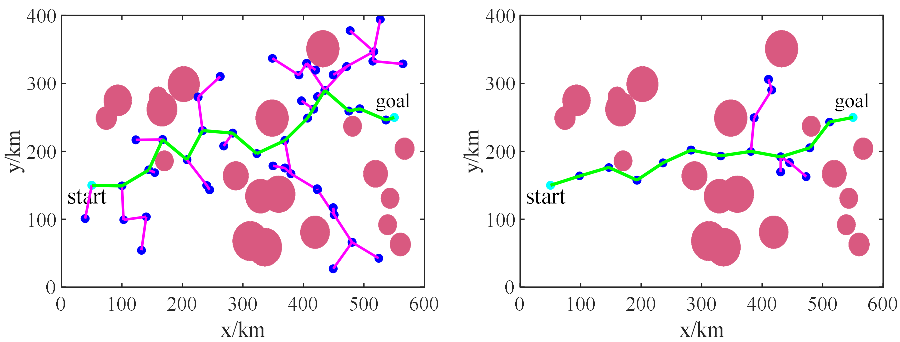
4.1. Simulation Experiment

In order to verify the performance of the improved RRT algorithm, the simulation comparison was conducted in different environments. The map scale of all environments is [600,400], the starting point of path planning is set as [50,150], and the target point is set as [550,250]. Due to the randomness of the RRT algorithm, 50 repeated tests were performed for the basic RRT algorithm and the improved RRT algorithm in each environment. Finally, the planned path was given and the average value of each indicator was counted. The planning results under the three environments are shown in Fig. 3-5, where the blue nodes are not on the red obstacles, and which can be seen clearly after magnification.



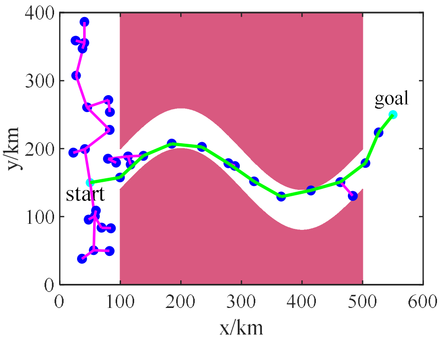
Basic RRT algorithm Improved RRT algorithm

Fig. 3: Planning results in environment 1

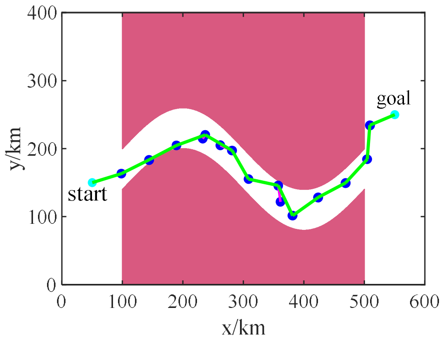


Basic RRT algorithm Improved RRT algorithm

Fig. 4: Planning results in environment 2



Basic RRT algorithm



Improved RRT algorithm

Fig. 5: Planning results in environment 3

In the figure: the green line represents the optimal path planned; The magenta lines are branches of the extension tree; start is the starting point. goal is the target point. Based on the planning results of the three environments, it can be seen that compared with the basic RRT algorithm, the path search of the improved RRT algorithm is more targeted and directional, which greatly reduces the ineffective extended search and improves the efficiency of path search.

4.2. Planning Efficiency Analysis

In general, if the iterations times of the successful path searched by the algorithm is less, the running time and path length are shorter, and the ratio of effective nodes is higher, the efficiency of path planning is higher. Therefore, the planning efficiency is measured by the average iterations times, the average running time, the average path and the ratio of effective nodes. The path planning result parameters in the three environments are shown in Table 1 respectively.

Table 1: Path planning results of two algorithms in different environments

Environment	Algorithm	Shortest path /km	Average path /km	average iteration times	Effective node ratio	Average time /s
1 (General)	Basic RRT	599.32	639.51	38	0.43	1.95
	Improved RRT	551.44	598.91	25	0.86	1.75
2 (Dense obstacle	Basic RRT	641.79	682.34	55	0.41	1.87
	Improved RRT	595.96	660.32	31	0.72	1.75
3 (Narrow channel)	Basic RRT	686.89	701.61	49	0.35	2.71
	Improved RRT	630.84	689.35	27	0.72	2.13

From Table 1, it can be seen that the average path length of the basic RRT algorithm in three environments is 674.39km, while the improved RRT algorithm is 649.53km, a decrease of 3.7%. The average iteration times of the basic RRT algorithm in the three environments is 47, while the improved RRT algorithm is 28, which is reduced by 41.5%. Meanwhile, the average running time of the basic RRT algorithm in the three environments is 2.18s, while the average running time of the improved RRT algorithm is 1.88s, reducing the average running time by 13.8%.

5. Conclusion

In this paper, an improved RRT algorithm is proposed to solve the problem that the basic RRT method has too much randomness in the path planning, and can not quickly obtain a path close to the optimal path. The proposed algorithm inherits the diversity of the basic RRT method, increases the number of random nodes in a single expansion, and selects nodes by integrating the gravity of the target point and the repulsion of obstacles. Simulation results show that the improved algorithm achieves good results in different environments.

6. Acknowledgement

This research was financially supported by the Guizhou Science and Technology Foundation ([2017]1088).

7. References

[1] F. Huo, J. Chi, Z. Huang, et al．Review of Path Planning Algorithms for Mobile Robots [J]. *Journal of Jilin University (Information Science Edition)*, 2018, 36(6): 639-647．

[2] D. Zhu, M. Yan. Survey on technology of mobile robot path planning [J]. *Control and Decision*, 2010, 25(07): 961-967.

[3] Q, Chen，H, Jiang，Y, Zheng．Summary of Rapidly-Exploring Random Tree Algorithm in Robot Path Planning [J]. *Computer Engineering and Applications*, 2019, 55(16): 10-17．

[4] Y. Y，P. J，W. W．Survey of optimal motion planning[J]．*IET Cyber-systems and Robotics*, 2019, 1(1): 13-19．

[5] KUFFNER JJ, LAVALLE S M. RRT-connect: An efficient approach to single-query path planning[A]. *Proceedings of the IEEE International Conference on Robotics and Automation*. San Francisco: IEEE, 2000. 995-1001．

[6] URMSONC, SIMMONSR. Approaches for heuristically biasing RRT growth [A]. *Proceedings of the IEEE Conference on Intelligent Robots and Systems*. Las Vegas, Nevada: IEEE, 2003. 1178-1183．

[7] PENG Hui, WANG Lin, SHEN Lin-cheng. The modified RRT-based real-time route planning for UAV area target searching[J]. *Journal of National University of Defense Technology*, 2009, 31(5): 86 -91.

[8] YERSHOVA A, JAILLET L, SIMEON T. Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain [A]. *Proceedings of the IEEE International Conference on Robotics and Automation*. Barcelona: IEEE, 2005. 3851-3861.