A Comparison Between A* and RRT Algorithm in Path Planning for Mobile Robot

Qixuan Cai

High School Affiliated to Nanjing Normal University, Nanjing, China

mengfen@ldy.edu.rs

Abstract. The ability of mobile devices to navigate environments is vital. Navigation plays a critical role in averting collisions and hazardous situations.The design of paths is an essential component of robot navigation. comprising the ability to determine the optimal route from the robot's current location to a given destination. And in all state-of-the-art algorithms for robots to design the optimal route, A* and RRT algorithm stand out as the two most widely used path planning techniques, one is graph-based approach while the other is sample-based. In this paper, some variants or improvement of the two methods will be described. And a comparison about A* and RRT will be presented with appropriate criteria, which are path length and computational time. The main finding is that the A* method outperforms the Rapidly Exploring Random Tree (RRT) approach in terms of computational efficiency and path distance optimization. But it requires to consider more thoughts in applications. The comparison of the A* and RRT algorithms helps comprehend their applicability in various application circumstances. In this work, Evaluating these aspects altogether comprehensively provides a deeper understanding of the suitability of each algorithm for specific robotic navigation tasks.

Keywords: path optimization, A*, RRT algorithm, mobile robot, path planning.

1. Introduction

The ability of mobile devices to navigate environments is vital. Navigation plays a critical role in averting collisions and hazardous situations. Path planning constitutes a vital aspect of robot navigation. entailing the skill to ascertain the best path from the robot's present position to a specified destination. During the process, the optimal path can be determined with the minimum path length and better smoothness using various algorithm (e.g. A*, RRT, Dijkstra algorithm). However, a mobile robot is a system designed to function within an environment that is both unpredictable and partially unfamiliar, which means more uncertainty. Researcher provides different types of challenges in path planning and navigation and majorly categorized them into two ways: local and global navigation problem [1]. A global path planner typically produces high-level path with a lower resolution from the starting point to the destination, circumventing major obstacles and addressing path planning within the environment configuration. While local path planning entails creating a detailed path only within a close vicinity of the global path's start and end points, aiding the robot in evading smaller obstacles and the navigation method that used to define direction may cause the accuracy to be reduced due to sensor-generated noise [2]. So, achieving complete autonomous navigation in a strange environment will be really complex. For various applications and robots, optimal criteria might depend on one or more conditions, including shortest physical distance, smoothness, scalability, low risk adaptability to different environments, maximum coverage area, and minimizing energy consumption [3]. So firstly, the comparison needs to clarify the environment that robot will be in and the requirement for the robot to accomplish tasks because the comparative standard is relating to these. In this thesis, it is assumed that the robot simply demands to find the minimum path distance, namely, less fuel requirement and the best smoothness of the path. In other words, the length of path and the time needed to compute are the primary behavior metrics under consideration in this research.

A comprehensive review of cutting-edge algorithm for robot to plan optimal path outlined in concluded that A* and RRT stand out as the two most commonly employed path planning methods[4], one being graph-based and the other sampling-based. These algorithms represent two distinct path planning approaches, based on heuristic search and random exploration, respectively. The theoretical significance of this thesis can be explained as follows: 1) By comparing these algorithms, researchers can validate their effectiveness and applicability in different problem in mobile robot, enriching the theory of path planning. 2) Path planning for mobile robots is a complex real-world problem, involving various environments, constraints, and requirements. Comparing A* and RRT algorithms helps understand their suitability in different application scenarios, offering theoretical support and guidance for practical applications.

The structure of the subsequent sections of this paper is as outlined below: A* and RRT Algorithm fundamental is shown in section 2. then a literature review of the considered methods will be provided. In section 3, improvement research on the A* and RRT algorithms is described, along with related variations. The results discussion is shown in section 4 and conclusion will be presented in section 5.

2. Algorithm Fundamental

2.1. Rapidly-Exploring Random Tree (RRT)

The RRT algorithm utilizing random node selection within the configuration space. Subsequently, it extends tree branches to construct a viable path [5]. The algorithm constructs a search tree, as shown in Figure 21.

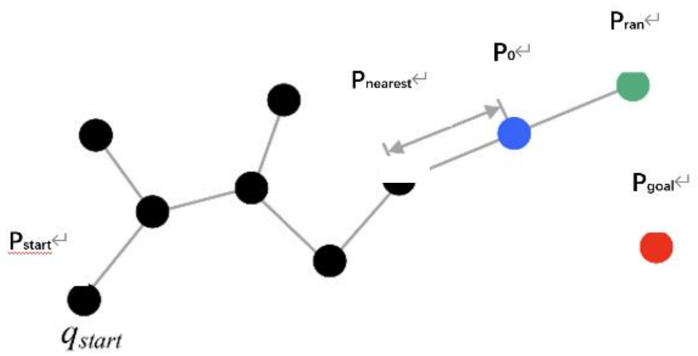

Fig. 1 RRT algorithm node extension [6]

The start from the beginning position Pinit and expands to search for a path leading to the goal position. say Pgoal. As the iterations progress, it will continue to expand. During each iteration, a random position (Pran) is selected from the space. If Pran lies in an obstacle free area, then the Pnearest can be determine in the tree. Then new node P0 is generated by connecting Pnearest and Pran through adding a growth step distance to the line toward two points. And if the distance between Pnearest and Pran is accessible according to the predefined length of step size, the tree will expand by connecting two points. Otherwise, continue the preceding steps until the predefined iteration number is exceeded certain value or the new node P0 is less than or equal to the threshold. If an initial path is found, the process continues until either a predefined time period elapses or an improved number of iterations is completed.

2.2. A* Algorithm

This algorithm was initially proposed by P. Hart, N. Nilsson, and B. Raphael in 1968. This algorithm adopts a heuristic approach to enable robots to function effectively in real-world environments, meaning that it solves problems by finding the path with the lowest costs among all possible paths. It traverses nodes within a graph from the starting node to the goal node. Utilizing heuristic information tailored to the problem's characteristics guides its performance, rendering it more effective than other blind search algorithms. Here's the formula of how to determine the shortest path:

$$f(x) = g(x) + h(x) \tag{1}$$

f(x) is the total cost function, The function f(x) provides an estimate of the importance of the candidate node along the path from the initial node to the goal node, facilitating the determination of the search order among candidate nodes for subsequent iterations. Its calculation is based on a specific formula. g(x) represents distance function, and h(x) is the heuristic function. g(x) illustrates the length between the starting node and the current node. The heuristic function describes the distance of the straight line connecting the current node to the goal node, since h(x) is an unknown, so it assumes a significant importance in the total cost function [7].

3. Related Works

The primary objective of this section is to recognize the A* and RRT algorithm for path optimalization. Since these two most researched path planning algorithms are from different categories, namely, graph-based and sample-based approach. Therefore, this section will concentrate in the traits of the two methods, and implementation, utilization, pros and cons of A* and RRT path planning methods and their variants.

3.1. A* Algorithm for Path Planning

The A* algorithm is the best general searching optimal path algorithm [8], it is also the most well-known and researched most common algorithm from the graph-based method, this part will illustrate what main characteristics this graph-based approach bring. And the applications and its associated variants of the A* algorithm as well.

3.1.1 Graph-based approach

Graph-based method have found extensive application in path planning across diverse domains owing to their simplicity. It defines the state space by partitioning it into a space-free grid. And the obstacles or objects are depicted as well. The algorithm traverses the grid points to create a path from the starting point to the goal, considering a planning of points perspective [9].

3.1.2 A* and its improvement

The conventional A* algorithm may experience slow search speeds and is prone to entering a failed search state when encountering trap obstacles in unfamiliar environments under complex scenarios. Therefore, there are a number of A* variants were developed. One of the developed A* is the path planning for virtual human motion [10], this method introduces a manual search marker to avoid the algorithm being trapped by narrow and arc-type trap barrier. The approach defines unbelievable points in the grid-based environment. As soon as the virtual human reaches it, this unbelievable point will be included in the set of manual search markers [10]. Another approach is a geometric A* algorithm, which effectively shortens the global search path by utilizing the functions P(x, y) and W(x, y), consequently removing unnecessary nodes from consideration, this can reduce the computational time [11]. A modified A star algorithm uses a preprocessing step call Rectangular Symmetry Reduction. New cells are added to the grid map when the initial, current, or goal state falls within the rectangle in this method. And this temporary cell connects all peripheral cells between adjacent rectangle to reduces symmetries and determine the path from every angle [12].

3.2. RRT Algorithm for Path Planning

RRT algorithm falls under the category of sampling-based algorithms suitable for path planning. It also has the ability to consider non-holonomic constraints [13]. However, the final path from RRT is not continuous and needs smoothing method, and it tend to be distant from the path with shortest length.

3.2.1 The feature of sample-based method

The sample-based approach tries to solve timing limitation which deterministic methods cannot cope with[13]. This method selects samples randomly from the state space or configuration state, then looking the connection between them, which can provide an advantage of time-saving.

3.2.2 Research of RRT and its variants

There are two main methods involve RRT to be considered, one is RRT* in anytime planning and the other is the fixed RRT approach based on different bias [13]. In the RRT* algorithm, instead of selecting the nearest parent node parent, RRT* evaluates all nodes within a neighborhood, then the cost of choosing each parent nodes will be evaluated, and be added in the tree. Finally, the iteration will proceed if reaching the near point via the new point would result in lower cost compared to its current parent and decreasing the total costs [14]. This method reduces the path length effectively. Another method is design to apply in a complex, dynamic environment it uses a sampling bias strategy to generate many types of function and values, which could be better implement in the specific condition planning the path and avoiding collision [15].

4. Discussion

In this section, a comparison between A* and RRT algorithm will be presented based on the fact about algorithm or researches mentioned.

4.1. Computational Time

The time algorithm needed to construct a path is significant for robot to work in a dynamic environment since the path is continuously changing, which means that the algorithm must process as fast as possible to deal with. Furthermore, the actuator also needs a time period to complete tasks, so if there is a long-time gap between the computational time and actuator's operation, the mobile robot may not have the ability to perform correctly. For A* algorithm, the searching is based on heuristic information, which will greatly reduce the time. However, according to [16], computing time can be slightly affected by different heuristic rules. And as to RRT algorithm, because the randomness of the algorithm, RRT will almost explore all the environment fairly, and that will increase the time needed. Similarly, the time can be influenced by the step length. If the step length increase, the time will decrease. But generally, the computational demand for A* is less than RRT.

4.2. Path Length and quality

Another crucial factor to consider when making comparisons is the length of the path. If the distance of the robot walk can always keep minimum, more energy or fuels can be saved, thereby allowing the robot to complete more tasks or operate over longer distances while maintaining its functionality. It is difficult to compare the path length between the two algorithms directly since the distance generated by RRT is associated with step size. And in general, the path length for A* is shorter than that for RRT. For this situation, RRT can be improved by using RRT* algorithm. However, although the quality of the path can be improved by shortening it, this enhancement comes at the cost of computational efficiency. This issue becomes even more pronounced as the dimensionality increases or when there are additional degrees of freedom [17]. Furthermore, as mentioned early, the final path built by RRT can be not continuous or even jerky, so further and better smoothing method is required to make walking more feasible for robots as opposed to A*.

4.3. Suggestion

First and foremost, the particular requirement is required in the first place. If completing the assignment requires traveling the least amount of physical distance, then A* should be selected. RRT algorithm can also be a viable option if the environment is unknown and very unpredictable because it is sample-based and selects nodes at random. Second, the A* algorithm's heuristic rule is diverse and might be difficult to construct, therefore it needs to be handled with sufficient care.

5. Conclusion

A* and Rapidly-Exploring Random Tree (RRT) stand as two prominent algorithms in the domain of robot path planning for navigation tasks. A* operates on a graph-based approach, meticulously exploring potential paths, while RRT adopts a sample-based strategy, creating a randomized tree structure to probe the search space.

In this paper, the basic knowledge of the two most popular path planning algorithm A* and RRT algorithm is introduced and some variants and improvements that be done for algorithms as well. Finally, based on two main factors, which are path length and computational time, a comparison between A* and RRT is made. The primary conclusion drawn from the analysis is that the A* algorithm surpasses the Rapidly-exploring Random Tree (RRT) algorithm in both path distance optimization and computational efficiency. And although RRT* can better the situation, there will be a trade-off between. Besides, other considerations should take count when comparing such as path step of RRT, or heuristic rules for A* and the main criteria for the performance of the algorithm according to different demand of the robot. Evaluating these aspects altogether comprehensively provides a deeper understanding of the suitability of each algorithm for specific robotic navigation tasks.

References

[1] Alatise, M. B., & Hancke, G. P. A review on challenges of autonomous mobile robot and sensor fusion methods. IEEE Access, 2020, 8, 39830-39846.

[2] Khaksar, W., Vivekananthen, S., Saharia, K. S. M., Yousefi, M., & Ismail, F. B. A review on mobile robots motion path planning in unknown environments. In 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS) 2015, 295-300.

[3] Eraghi, N. O., Lopez-Colino, F., De Castro, A., & Garrido, J. Path length comparison in grid maps of planning algorithms: HCTNav, A* and Dijkstra. In Design of Circuits and Integrated Systems, 2014, 1-6.

[4] Zammit, C. and van Kampen, E. J., “Comparison between A* and RRT Algorithms for UAV Path Planning”, AIAA Guidance, Navigation and Control Conference, AIAA SciTech Forum, Kissimmee, FL, 2018.

[5] LaValle, S. M., & Kuffner Jr, J. J. Randomized kinodynamic planning. The international journal of robotics research, 2001, 20(5), 378-400.

[6] Guo, T., Wang, J., Chen, W., Wang, Z., Chen, G., Lin, Y., & Liu, J. ArticlesObstacle Avoidance Path Planning for Mobile Robotic Arm with Multi-Degree of Freedom Based on TBPS-RRT Algorithm.

[7] Fu, B., Chen, L., Zhou, Y., Zheng, D., Wei, Z., Dai, J., & Pan, H. An improved A* algorithm for the industrial robot path planning with high success rate and short length. Robotics and Autonomous Systems, 2018, 106, 26-37.

[8] Sathyaraj, B. M., Jain, L. C., Finn, A., & Drake, S. Multiple UAVs path planning algorithms: a comparative study. Fuzzy Optimization and Decision Making, 2008, 7: 257-267.

[9] Fu, B., Chen, L., Zhou, Y., Zheng, D., Wei, Z., Dai, J., & Pan, H. An improved A* algorithm for the industrial robot path planning with high success rate and short length. Robotics and Autonomous Systems, 2018, 106, 26-37.

[10] Yao, J., Lin, C., Xie, X., Wang, A. J., & Hung, C. C. Path planning for virtual human motion using improved A* star algorithm. In 2010 Seventh international conference on information technology: new generations. 2010, 1154-1158.

[11] Tang, G., Tang, C., Claramunt, C., Hu, X., & Zhou, P. Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment. IEEE access, 9, 59196-59210.

[12] Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., & Jurišica, L. Path planning with modified a star algorithm for a mobile robot. Procedia engineering, 2021, 96, 59-69.

[13] González, D., Pérez, J., Milanés, V., & Nashashibi, F. A review of motion planning techniques for automated vehicles. IEEE Transactions on intelligent transportation systems, 2015, 17(4), 1135-1145.

[14] Karaman, S., Walter, M. R., Perez, A., Frazzoli, E., & Teller, S. Anytime motion planning using the RRT. In 2011 IEEE international conference on robotics and automation. 2011, 1478-1483.

[15] Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E., & How, J. P. Real-time motion planning with applications to autonomous urban driving. IEEE Transactions on control systems technology, 2009, 17(5), 1105-1118.

[16] Liu, X., & Gong, D. A comparative study of A-star algorithms for search and rescue in perfect maze. In 2011 international conference on electric information and control engineering. 2011, 24-27.

[17] Noreen, I., Khan, A., & Habib, Z. A comparison of RRT, RRT* and RRT*-smart path planning algorithms. International Journal of Computer Science and Network Security (IJCSNS), 2016, 16(10), 20.