ISSN: 2472-3703 DOI: 10.6911/WSRJ.202404_10(4).0002

Improved A* Path Planning Algorithm for Fire Robot

Wentao Qi, Xianjie Liu, Kaikai Yuan, Wanqi Chen

School of Information Engineering, Henan University of Science and Technology, Luoyang, China

Abstract

In the problem of point-to-point path planning for intelligent robots in simple scenarios, we propose an improved A* algorithm. This algorithm is an enhancement over the original A* algorithm. The original A* algorithm can find the shortest path plan by avoiding obstacles in the scene given a coordinate point, and it has the characteristics of simplicity and efficiency. However, for complex scenes, the paths generated by the basic A* algorithm appear too sharp at corners, which is not conducive to the actual movement of intelligent robots. Therefore, we introduce a polynomial fitting method to reduce the degree of path winding and achieve smoothing of the path, allowing the robot to move more smoothly. Through this improvement, we expect that the optimized path will better adapt to real scenes, improving the comfort and feasibility of robot movement. This improved algorithm maintains the simplicity and efficiency of the original A* algorithm while better adapting to practical application needs, providing a more optimized solution for the problem of path planning for intelligent robots in simple scenarios.

Keywords

Intelligent Robots, Path Planning, Improved A* Algorithm, Polynomial Fitting, Path Smoothing.

1. INTRODUCTION

In the rapidly developing electronics industry in China and the rapid development of neural networks, a large number of intelligent electronic products with various functions have emerged in the market. In the field of intelligent robots, path planning is a fundamental basis for achieving precise motion control in various robots.

Traditional path planning algorithms include Dijkstra algorithm [1-2], A* algorithm [3-4], D* algorithm [5-6], artificial potential field algorithm [7-8], etc. The specific situations of each algorithm are as follows:

(1) Dijkstra algorithm: The Dijkstra algorithm was proposed by Edsger Wybe Dijkstra in 1956, which is used to find the shortest path between nodes in a graph. The basic idea of this algorithm is greedy strategy, mainly characterized by starting from the starting point and expanding outward layer by layer until the target point is reached. During the expansion process, the Dijkstra algorithm selects the nearest unvisited node to the current point and uses that node to update the distance values of other nodes.

(2) A* algorithm: The A* algorithm is a method that combines the Dijkstra algorithm and breadth-first search algorithm. It uses a heuristic function to find the shortest path more quickly, making it the most effective direct search method for solving the shortest path in static road networks.

(3) D* algorithm: The D* algorithm was proposed by Professor Anthony Stentz from Carnegie Mellon Robotics Center in 1994, mainly used for robot exploration. The D* algorithm stores the

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202404_10(4).0002

shortest path information from each point to the destination, which is efficient. In addition, the D* algorithm performs backward search, starting from the target point. During the initial traversal, it is similar to the Dijkstra algorithm and saves information about each node.

(4) Artificial Potential Field Algorithm: The Artificial Potential Field (APF) method is a local path planning algorithm based on a virtual force field. In this algorithm, the environment is designed as an abstract artificial gravitational field, where the target point generates "gravity" for the robot and obstacles generate "repulsion". The robot's movement is controlled by the resultant force.

In practical operations, considering factors such as the complexity of task execution scenarios, the expected results of individual products, and the actual computing capabilities of the platform, to reduce the burden of robot turning and changing speed,we chose to use the improve A* algorithm for path planning in software design and development.

The contributions of this paper can be summarized as follows:

(1) The polynomial fitting algorithm is introduced to reduce the sharpness at path turning points and improve the smoothness of the path.

(2) The idea of iterative optimization is introduced to iteratively optimize the fitted curve, ensuring a good balance between path feasibility and smoothness.

(3) By combining the global optimality of the A* algorithm with the smoothing effect of polynomial fitting, the efficiency and stability of robot movement are improved.

2. OVERVIEW OF THE TRADITIONAL A* ALGORITHM

The A* algorithm is a path planning algorithm based on heuristic search [9], which is suitable for situations where the global environmental information is known. It guides the search process through an evaluation function, starting from the starting point, continuously expanding nodes and calculating their evaluation function values, and then selecting the node with the smallest evaluation function value for further expansion. This search method is similar to the optimization process in biology, by continuously evaluating and selecting the optimal node, the optimal path can be eventually found. This heuristic search method has wide applications in fields such as communication network planning and intelligent system design, and can effectively solve path planning problems in complex environments. The evaluation function is:

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

Where n represents the current node where the robot is located, $f(n)$ is the comprehensive evaluation value of node n, $g(n)$ is the actual cost from the starting node to node n, and $h(n)$ is the estimated cost from node n to the target node, also known as the heuristic function [10]. The choice of $g(n)$ and $h(n)$ values will directly affect the performance and search results of the A* algorithm.

However, in its actual application, we found that due to the physical constraints of intelligent vehicles, solely relying on the A* algorithm resulted in overly coarse planning results that could not be directly used. Therefore, we decided to make certain improvements to the original basic A* algorithm. We processed the raw and sharp polyline paths generated by the A* algorithm through a certain rounding process, aiming to reduce the sharp turns between paths in multi-point connected paths and achieve smoother transitions. This approach not only retains the

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202404_10(4).0002

existing advantages of the A* algorithm in global path planning but also integrates it with reality, achieving shortest path planning.

3. IMPROVE A* ALGORITHM BASED ON THE PATH SMOOTHING FITTING

Although the A* algorithm can obtain the optimal global path in a static environment, the planned path has too many turning points, increasing the burden on the robot's turning and speed changing. Therefore, we can use polynomial fitting to smooth the path found by the A* algorithm [11], in order to reduce the tortuosity of the path and make the robot's movement faster and smoother.

Polynomial fitting is a method with relatively low computational complexity, which can minimize the loss function values for all samples and can sometimes obtain the global optimum solution or an approximate optimum solution. As a type of curve fitting, polynomial fitting uses the method of mapping a set of data points to a polynomial function to achieve curve fitting.

The process of polynomial curve fitting is described as follows: For a set of corresponding measurement data $(x_0, y_0),(x_1, y_1),(x_2, y_2),\cdots,(x_m, y_m)$ of x and y, fit them into a polynomial of nth degree $(n<m)$:

$$y = f(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_n x^n = \sum_{i=0}^{n} \theta_i x^i \tag{2}$$

where θ_n is the weight value, and the sum of squared error distances between the corresponding $y_1 \sim y_m$ values and the curve is calculated at the given points $x_1 \sim x_m$:

$$H^2 = \sum_{i=1}^{m} \left[f(x_i) - y_i \right]^2 \tag{3}$$

To make the fitted curve reflect the trend of measurement data changes, it is required that the sum of squared errors of all points is minimized, i.e., H^2 takes the minimum value. There are various mathematical methods to solve for the coefficients of the polynomial curve using least squares, including algebraic method, matrix method, etc. Since the algebraic method involves tedious calculations and large computations, we will use the matrix method to introduce the solution steps below:

To find the minimum value, we take the first-order partial derivative of H^2 with respect to it, making its result zero, which is:

$$\begin{aligned}
\frac{\partial H^2}{\partial \theta_0} &= \sum_{i=1}^{m} 2\left[f(x_i) - y_i \right] = 0 \\
\frac{\partial H^2}{\partial \theta_1} &= \sum_{i=1}^{m} 2x_i\left[f(x_i) - y_i \right] = 0 \\
\frac{\partial H^2}{\partial \theta_2} &= \sum_{i=1}^{m} 2x_i^2\left[f(x_i) - y_i \right] = 0 \\
&\vdots \qquad \vdots \qquad \vdots \\
\frac{\partial H^2}{\partial \theta_n} &= \sum_{i=1}^{m} 2x_i^n\left[f(x_i) - y_i \right] = 0
\end{aligned} \tag{4}$$

Convert them into the form of a system of equations:

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202404_10(4).0002

$$\theta_0 m + \theta_1 \sum_{i=1}^{m} x_i + \theta_2 \sum_{i=1}^{m} x_i^2 + \cdots + \theta_n \sum_{i=1}^{m} x_i^n = \sum_{i=1}^{m} y_i \tag{5}$$

Written in matrix form is:

$$\begin{bmatrix} m & \sum_{i=1}^{m} x_i & \cdots & \sum_{i=1}^{m} x_i^n \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x_i^2 & \cdots & \sum_{i=1}^{m} x_i^{n+1} \\ \sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i^3 & \cdots & \sum_{i=1}^{m} x_i^{n+2} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{m} x_i^n & \sum_{i=1}^{m} x_i^{n+1} & \cdots & \sum_{i=1}^{m} x_i^{2n} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_i y_i \\ \sum_{i=1}^{m} x_i^2 y_i \\ \vdots \\ \sum_{i=1}^{m} x_i^n y_i \end{bmatrix} \tag{6}$$

The Van der Monde matrix in (6) can be simplified as:

$$\begin{bmatrix} 1 & x_1 & \cdots & x_1^n \\ 1 & x_2 & \cdots & x_2^n \\ 1 & x_3 & \cdots & x_3^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \cdots & x_m^n \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix} \tag{7}$$

From (7), we can obtain the matrix: $X*A=Y$,then: $A=\left(X'*X\right)-1*\left(X'*Y\right)$ .The matrix coefficient A can be obtained from Matrix, and the coefficients of the polynomial fitting curve can be solved as follows: $\theta_0 \sim \theta_n$.

4. ALGORITHM IMPLEMENTATION PROCESS

The A* algorithm uses an estimated value (heuristic function) to predict the cost from the starting node to the target node during path planning. After extracting all control points, it selects the path with the minimum total cost by backtracking from the target node to the starting point. However, the A* algorithm may generate some winding paths when generating paths, especially when there are many grid points. This causes the robot to frequently change directions while driving, increasing the load on the motor. In addition, this frequent direction change also poses difficulties for the gyroscope's balance, potentially causing vehicle instability and even affecting the durability of certain parts of the robot. Therefore, smoothing the tortuous path is of necessity for the robot's driving.

Polynomial fitting can generate a polynomial through a set of data points, achieving the effect of smoothing the path. Therefore, by using polynomial fitting to improve the A* algorithm, not only can we obtain the globally optimal static path, but also improve the feasibility and comfort of the path. The flowchart of the improved algorithm is shown in Figure 1:

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202404_10(4).0002

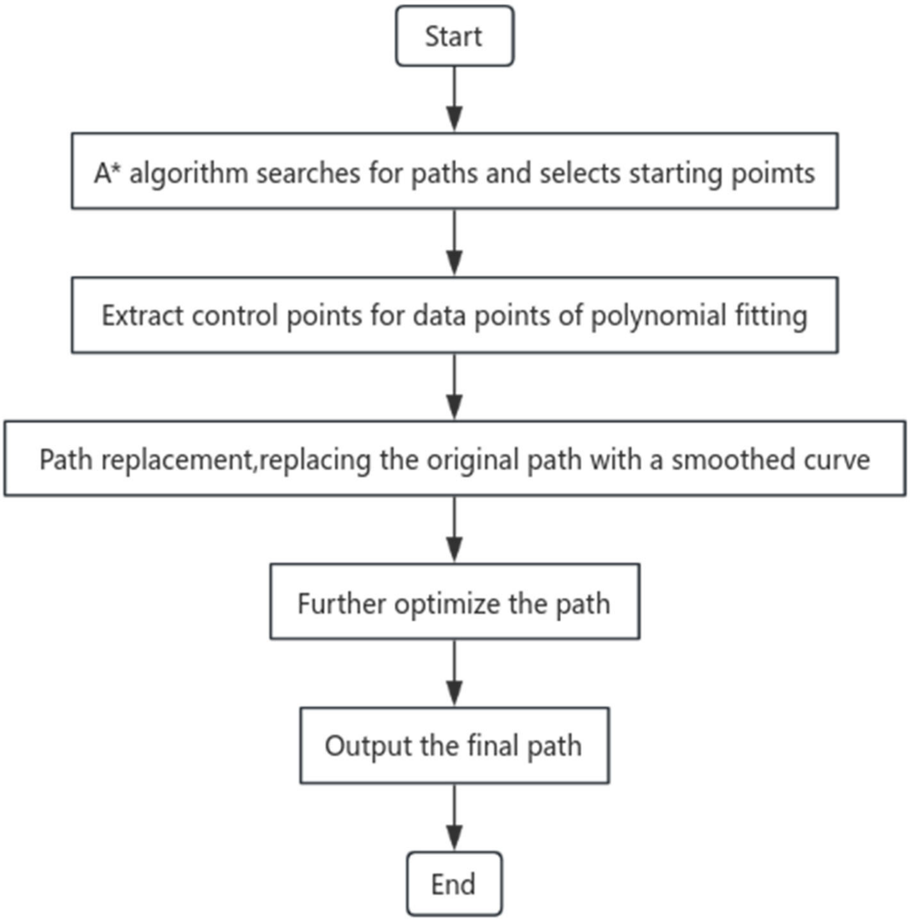

Figure 1. Improved A* algorithm implementation flowchart

As shown in Figure 1, the first step is to use the A* algorithm to find the initial path from the starting point to the target point. The control points extracted by the A* algorithm are used as data points for polynomial fitting. Secondly, a polynomial fitting method is used to fit the control points and generate a smooth curve. The initial path obtained by the A* algorithm is replaced with the smooth curve obtained by polynomial fitting. After replacing the path, the polynomial fitting method can be further used to optimize the curve again to ensure a good balance between feasibility and smoothness. Finally, the optimized path is taken as the navigation path of the robot.

5. CONCLUSION

Through the improvement of the A* algorithm and path smoothing fitting, this paper effectively solves the robot's path planning problem in complex environments. By introducing heuristic search methods to guide the search process, the robot can find the optimal path. However, the paths planned by the A* algorithm may have too many turning points, which brings a burden to the robot's movement. To solve this problem, we use polynomial fitting to smooth the path. By mapping a set of data points to a polynomial function, we can reduce the tortuosity of the path and make the robot move more quickly and smoothly.

In practical applications, we can choose appropriate polynomial degrees and weight values according to specific situations to achieve the best path smoothing effect. At the same time, we need to pay attention to the deviation that may be introduced during the path smoothing process, so validation and adjustment are necessary in practical applications. Through the improvement of the A* algorithm and path smoothing fitting, we can achieve efficient path

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202404_10(4).0002

planning for robots in complex environments. This not only improves the robot's motion efficiency but also reduces its energy consumption and lifespan. In the future, with continuous technological development, we can further optimize algorithms and fitting methods to meet more complex environmental and task requirements.

ACKNOWLEDGMENTSAS

This work was supported by the National Innovative Entrepreneurship Training Program For Undergraduates (2023102).

REFERENCES

[1] HU Li-kun, WANG Shuai-jun, LV Zhi-lin, ZHU Wen-tian: Multi-objective Search Based on D* -ant Colony Fusion Algorithm, Journal of Chinese Computer Systems, Vol. 41 (2020) No. 3, p. 471-476.

[2] SEIFEDINE K, GENNADY A, VIKTOR F: D-star algorithm modification, International Journal of Online and Biomedical Engineering, Vol. 16 (2020) No. 8, p. 108-113.

[3] Chi Xu, Li Hua, Fei Jiyou: Research on robot random obstacle avoidance method based on fusion of improved A* algorithm and dynamic window method, Chinese Journal of Scientific Instrument, Vol. 42 (2021) No. 3, p. 132-140.

[4] ZHANG Zhi, WENG Zong-nan, SU Li, GUANG Zheng-hui: Indoor Mobile Robot Collision Avoidance Path Planning, Journal of Chinese Computer Systems, Vol. 40 (2019) No. 10, p. 2077-2081.

[5] KAZUNORI U, LEONARD B: Dijkstra algorithm-based ray tracing for tunnel-like structures, International Journal of Grid and Utility Computing, Vol. 11 (2020) No. 1, p. 84-92.

[6] LUO Min, HUO Xiaorong, Yang Jing: Surface optimal path planning using an extended Dijkstra algorithm, IEEE Acess, Vol. 8 (2020), p. 147827-147838.

[7] YANG Wenlin, WU Peng, ZHOU Xiaoqi, et al: Improved artificial field and dynamic window method for amphibious robot fish path planning, Applied Sciences, Vol. 11 (2021) No. 5, p. 2114-2114.

[8] JEANFRANCOIS D, STEPHANE V, PIERRE M: Contributions on artificial potential field for effective obstacle avoidance, Fractional Calculus and Applied Analysis, Vol.24 (2021) No. 2, p. 421-446.

[9] XU Tang-jian: Parallel Design and Improvement of A* Path Finding Algorithm, Modern Computer, Vol.21 (2018) No. 8, p. 1007-1423.

[10] QI Xuanxuan, HUANG Jiajun, CAO Jian'an*: Path planning for unmanned vehicle based on improved A* algorithm, Journal of Computer Applications, Vol.10 (2022) No. 7.

[11] LI Ming, GAO Xing-wei, XU Ai-gong: A modified polynomial fitting of cycle-slip processing, Science of Surveying and Mapping, Vol.4 (2008) No. 27, p. 1009-2307.