Mobile Robot

Assignment 01

MUHAMMAD ABDULLAH YOUSAF
450767

Mobile Robot Path planning Improved A* algorithm

Authors:

Chengyang Han, Baoying Li

(School of Information science and Engineering, Dalian Polytechnic University, Liaoning, China)

Abstract:

This research paper addresses the problem of inefficiencies in the A* algorithm for mobile robot path planning, such as excessive redundant nodes, slow search times, and sharp path turns.

The authors propose an improved A* algorithm that incorporates a five-neighborhood search with dynamic weighting in the heuristic function and employs a second-order Bezier curve for path smoothing12.

Simulation results in PyCharm demonstrate that the improved algorithm reduces redundant nodes by 86.57% and search time by 82.07% without compromising cost precision3.

The processed paths exhibit no large angle turning points, ensuring continuity of speed and acceleration, which enhances driving fluency for mobile robots.

Traditional A* Algorithm:

The core idea of the traditional A* algorithm is to find the minimum-cost path from the starting point to the target point within a constrained map.

F(n) = G(n)+H(n)

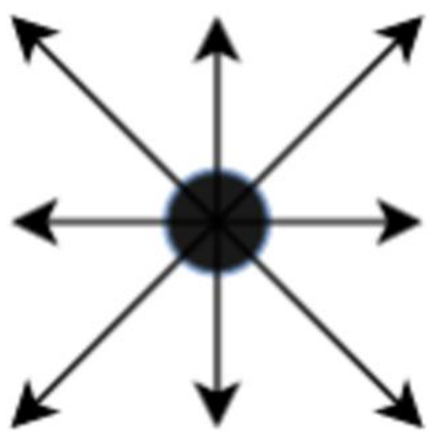
The heuristic function for the Euclidean distance is:

$$H(n) = \sqrt{(x_n - x_{goal})^2 + (y_n - y_{goal})^2}$$

The heuristic function for the Manhattan distance is:

$$H(n) = |x_n - x_{goal}| + |y_n - y_{goal}|$$

Traditional A* directions of the node will be:



Improved A* Algorithm:

The paper presents an enhanced A* algorithm for mobile robot path planning, addressing issues like redundant nodes, slow search times, and sharp path turns.

$$F(n) = G(n) + W \times H(n)$$

Dynamic Weighting & Five-Neighborhood Search:

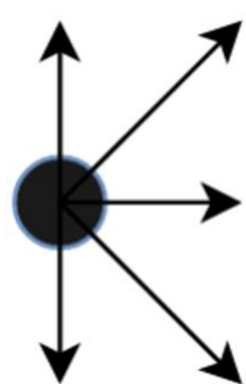
It introduces dynamic weighting in the heuristic function and a five-neighborhood search method, reducing redundant nodes by 86.57% and search time by 82.07%. This is done by limiting the quadrants of the vector with a direction from current state to goal state.

By altering the value of W, the search efficiency is affected. Simulation comparisons reveal that different values of W lead to distinct search conditions, as shown in Table I.

TABLE I
COMPARISON TABLE OF WEIGHT COEFFICIENTS

$H(n)$	w value	Weight allocation ratio	Number of search points	search speed	route	cost
Euclidean distance	$w = 1$	$G(n) = H(n)$	medium	medium	medium	medium
	$w > 1$	$G(n) < H(n)$	few	fast	Non optimal	tall
	$0 < w < 1$	$G(n) > H(n)$	many	slow	optimal	low

Five-neighborhood Diagram;



Bezier Curve Path Smoothing:

The paper utilizes a second-order Bezier curve for path smoothing, ensuring the continuity of speed and acceleration, thus improving driving fluency.

The formula used is given as;

$$B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2, \quad t \in [0,1]$$

Discrete Point selection diagrams in both cases are as follow;

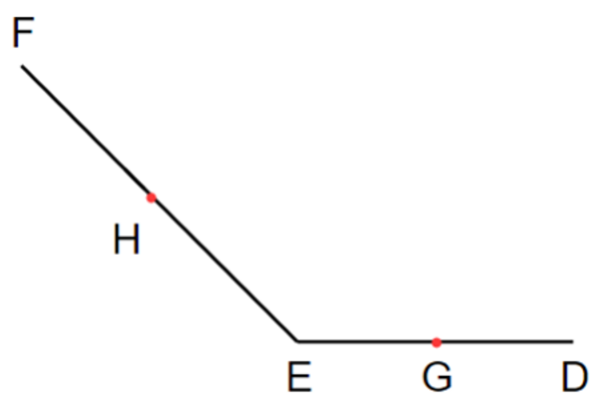


Fig. 5. Intermediate Point Selection Diagram

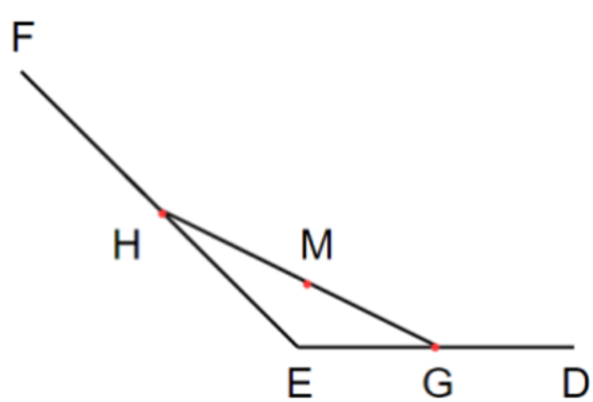


Fig. 6. Path points after optimization

Simulation Results:

Simulations in PyCharm demonstrate the improved algorithm's effectiveness, with precise pathfinding, fewer irrelevant nodes, faster search times, and optimal cost1. The paper concludes that the proposed improvements are reasonable and effective for mobile robot path planning.

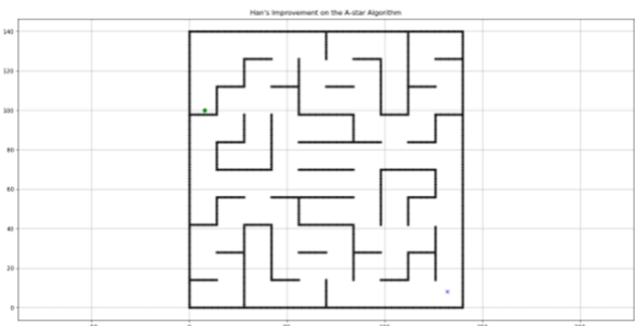


Fig. 7. 140*140 Simulation Map

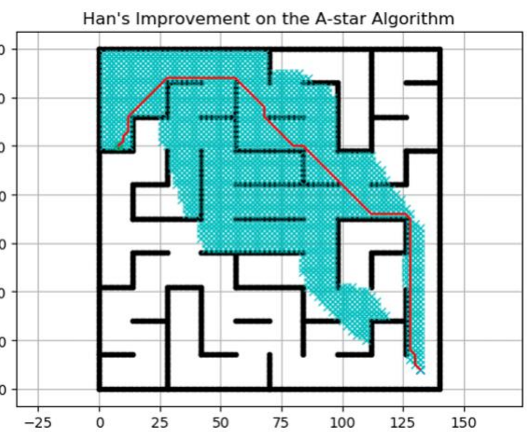


Fig. 8. Simulation Diagram of Traditional Eight Neighborhood Path Planning

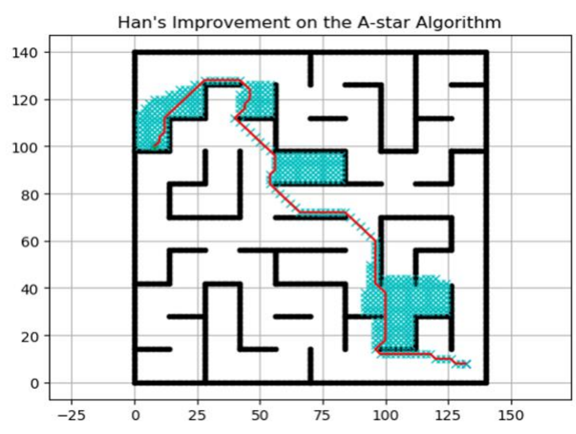


Fig. 9. Simulation Diagram of Improved Eight Neighborhood Dynamic Weighted Path Planning

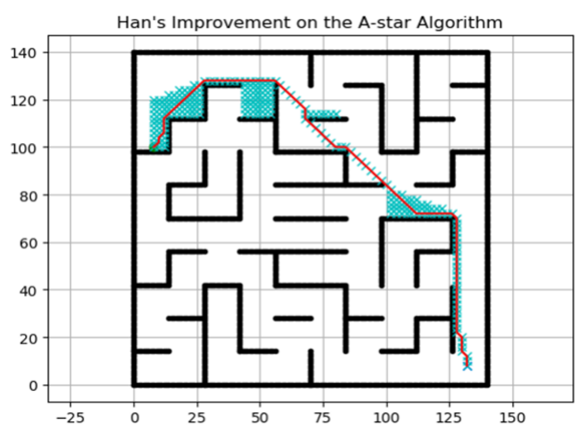


Fig. 10. Simulation Diagram of Improved Five Neighborhood Dynamic Weighted Path Planning

TABLE II
ALGORITHM PERFORMANCE COMPARISON TABLE

Environment map size	Algorithm	Number of redundant nodes	Search time	cost
120×120	Traditional Eight Neighborhood A* Algorithm	1757	18.07134	113.15432
	Improved A* Algorithm with Eight Neighborhood Dynamic Weighting	483	4.48411	122.81118
	Improved A* Algorithm with Dynamic Weighting of Five Neighborhood	236	3.05042	113.15432

Path smoothing (Bezier):

The path obtained by the five-neighborhood dynamic weighted A* algorithm is not optimal and has the problem of sharp turn at large Angle, so the second-order Bessel curve is used for processing. Divide the minimum node set on the path into 7 groups for grouping optimization and fusion processing.

The optimized path is shown as;

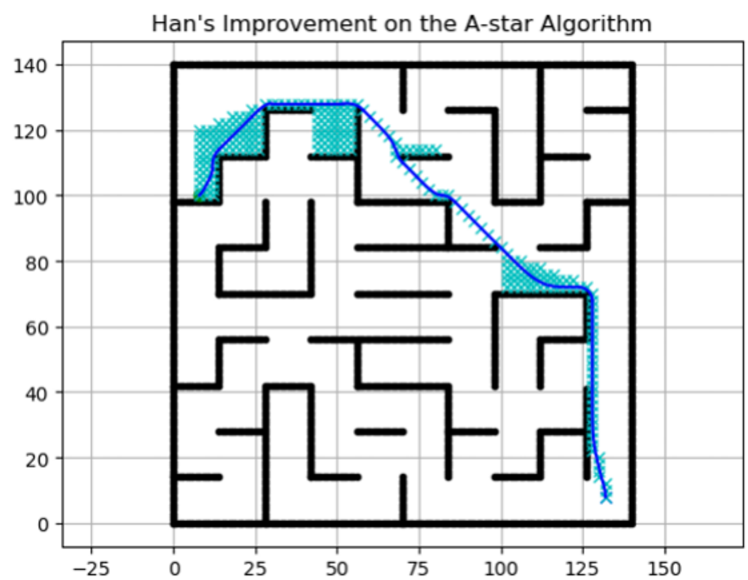


Fig. 11. Simulation Diagram of Whole Path Smoothing Processing

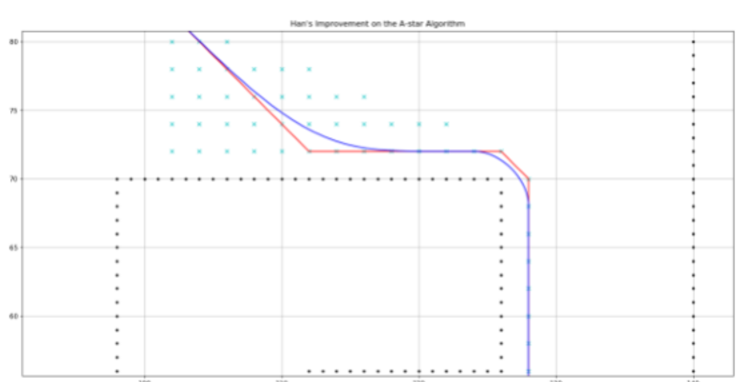


Fig. 12. Enlarged view of local corner areas from 102 to 127

Conclusion:

Through the comparison between the simulation results and the data in the data table, it can be seen that A* algorithm after dynamic weighting of five neighborhoods, combined with second-order Bezier curve smoothing optimization, is reasonable and effective in the path planning direction of mobile robots.

Research Idea:

We can use Machine learning techniques to adaptively learn and update the heuristic function used by A* in real-time, enabling the robot to navigate efficiently while exploring unfamiliar terrains.