Autonomous land vehicle path planning algorithm based on improved heuristic function of A-Star

Jing Zhang*, Jun Wu, Xiao Shen and Yunsong Li

Abstract
The path planning of autonomous land vehicle has become a research hotspot in recent years. In this article, we present a novel path planning algorithm for an autonomous land vehicle. According to the characteristics of autonomous movement towards the autonomous land vehicle, an improved A-Star path planning algorithm is designed. The disadvantages of using the A-Star algorithm for path planning are that the path planned by the A-Star algorithm contains many unnecessary turning points and is not smooth enough. Autonomous land vehicle needs to adjust its posture at each turning point, which will greatly waste time and also will not be conducive to the motion control of autonomous land vehicle. In view of these shortcomings, this article proposes a new heuristic function combined with the artificial potential field method, which contains both distance information and obstacle information. Our proposed algorithm shows excellent performance in improving the execution efficiency and reducing the number of turning points. The simulation results show that the proposed algorithm, compared with the traditional A-Star algorithm, makes the path smoother and makes the autonomous land vehicle easier to control.

Keywords
Autonomous land vehicle, path planning algorithm, A-Star algorithm, artificial potential field, heuristic function

Date received: 28 March 2021; accepted: 11 August 2021

Introduction
In recent years, autonomous land vehicle (ALV) has been widely used in emergency rescue, port freight, logistics distribution, and other fields. The development of ALVs is based on path planning, vehicle control, environment perception, and so on.1 Path planning occupies a core position in the fields of ALV, the efficiency of path planning algorithms directly affects the efficiency of ALV’s path finding, and implementation of planning capabilities determines whether the ALV can accomplish the tasks efficiently and safely.2,3 On the basis of path planning, there are many extended studies. For example, fully sharing the environmental information between different vehicles in autonomous driving is not only conductive to the realization of more adequate path planning but also can ensure higher security level.4 In addition, the trajectory prediction of dynamic obstacles is also a hot topic of extended research.5 All these studies have expanded the application and development direction of path planning. In the last decades, lots of path planning algorithms are

*State Key Laboratory of Integrated Service Networks, Xidian University, Xi’an, China
Corresponding author:
Jing Zhang, State Key Laboratory of Integrated Service Networks, Xidian University, No. 2, South Taibai Road, Hi-Tech Development Zone, Xi’an, Shaanxi Province 710071, China.
Email: jingzhang@xidian.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
proposed, such as Floyd algorithm,6 Dijkstra algorithm,7 A-Star algorithm,8 colony algorithm,9 particle swarm algorithm,10 and artificial potential field.11,12

Due to the characteristics of fast search speed, the A-Star algorithm has been widely applied in solving the shortest path problem of the static road network.13 Derived from Dijkstra algorithm, A-Star introduced heuristic function to enable the fast node search speed. However, the heuristic function of the A-Star algorithm only considered distance information, which leads to redundant expansion nodes in the pathfinding process. In view of the shortcomings of the A-Star algorithm, a large number of scholars have proposed corresponding improved algorithms. Hongbin et al.14 proposed a hybrid algorithm combining an improved A-Star algorithm and a dynamic window method, which has good performance in complex dynamic environment. Chen et al.15 defined the valuation function as an exponential decay method to reduce redundant expansion. Wang et al.16 proposed an efficient full area coverage path planning algorithm based on grid mapping. Zhang et al.17 considered the actual application scenarios of unmanned boats and improved the evaluation function by adding the corner loss term, so that the final path has the smallest corner and reduces the loss on the path. Wu et al.18 improved the evaluation function of the A-Star by establishing a taboo table to quickly and effectively implement path planning. Lin et al.19 improved the real-time performance of the A-Star by introducing the influence of the parent node but did not consider the influence of the value of weight when weighting the evaluation function.

However, the above algorithms only consider the position of the obstacle in the global map but not consider the number of obstacles and distance around the ALV in the path planning process, which is not sufficient for the use of map information. Aiming at the problem of unsmooth path of the A-Star algorithm, we propose a new heuristic function, which is combined with the artificial potential field to optimize the path and improve search efficiency. When calculating heuristic items, our method considers not only the distance from the current position to the target position but also the obstacle information around the current position. The simulation experiment shows that our algorithm performances better than A-Star. Comparing the A-Star algorithm and proposed algorithm in the grid map environment, the result verifies that our method is more conducive to motion control and the planned path is smoother.

Establishment of environment model for ALV path planning

For path planning, environment modeling is a crucial step. According to establish environment model, the real environmental scene is converted into an abstract scene that can be processed by the robot, which is convenient for the robot to carry out path planning. Common map representation methods include topology modeling method, outline map method, and grid map method.

We use the grid map method to establish a grid map to record the obstacle grids and path information of ALV. We define the attributes of the grid, store information such as the parent node of the track and whether the grid is an obstacle and use a matrix to map the information of each node in the grid map. Then binarize the grid map, define 0 as the passable grids and 1 as the obstacle grids to obtain the binarized map. Figure 1 shows an example of the grid map. The black grid represents obstacles, whereas the white grid is the passable area. In the environment map of rasterization model, the map is divided into several closely connected small grids, and each grid represents a node of the map. Each node in the map only constitutes a path with eight adjacent nodes. Therefore, in the grid map, the movement direction of the ALV is generally only 8, which are the directions of the 8 neighboring nodes of the grid map.

Background

A-Star algorithm

A-Star algorithm was introduced by Hart et al.21 and now is widely used in the fields of the shortest path problem. As a typical heuristic search algorithm, the A-Star algorithm evaluates nodes through heuristic search, which improves the efficiency of node search and has better performance and accuracy.21 The heuristic search using the valuation function, and the valuation function is expressed as follows

\[ f(n) = g(n) + h(n) \] (1)
where \( f(n) \) denotes the total estimated cost from the start position through the current position \( n \) to the target position, \( g(n) \) denotes the actual cost of the path from the starting position to the current position, and \( h(n) \) denotes the estimated cost from position \( n \) to the target position. The pathfinding process of the A-Star algorithm is shown in Figure 2.

### Artificial potential field

Khatib introduced the artificial potential field method in 1986. The artificial potential field method constructs a gravitational potential field with the target position as the center. The value of the attractive potential field increases with the increase of the distance between the ALV and the target, and the direction is from the autonomous vehicle to the target. The mathematical expression of the attractive potential field function is as follows

\[
F_a(x, y) = \varepsilon \cdot \left[ (\text{goal}_x - x)^2 + (\text{goal}_y - y)^2 \right]^{3/2}
\]

(2)

\( f_a(x, y) \) is the attractive potential field function and \( \varepsilon \) is a scaling factor.

With the current position of the ALV as the center, a repulsive force field is constructed, which decreases as the distance between the obstacle and the ALV increases, and the direction is from the obstacle to the ALV. The mathematical expression of the repulsive force field function is as follows

\[
F_r(x, y) = \begin{cases} 
\eta \cdot \left( \frac{1}{\rho(x, y)} - \frac{1}{d_0} \right)^2, & \text{if } \rho(x, y) \leq d_0 \\
0, & \text{if } \rho(x, y) > d_0 
\end{cases}
\]

(3)

where \( \rho(x, y) \) is the distance from the current point \( (x, y) \) to the obstacle and \( d_0 \) is a distance threshold. When the distance from the current point to the nearest obstacle is greater than \( d_0 \), no repulsive force is generated. \( \eta \) is also a simple scaling factor.

These two potential fields together form an abstract artificial potential field. The calculation of the potential field force of the artificial potential field method is shown in Figure 3. In the artificial potential field, for a certain point in space, the ALV will be subjected to a force with a certain magnitude and direction. The artificial potential field method abstracts the motion of the ALV as the motion dragged by this virtual force.

### Method

In application scenarios, the ALV first plans a path in the environment map and then proceeds along with the path. However, when using the traditional A-Star algorithm in path planning, the planned path contains too many turning points and the path is not smooth enough. As a result, the ALV needs to make many unnecessary posture adjustments during the path tracking process to adjust the moving...
direction of the ALV. When the vehicle moves along the winding path, it will slow the speed of the ALV, and the excessive posture adjustment can easily disrupt the balance of the ALV. To solve those problems mentioned above, this article combines the idea of artificial potential field method to improve the A-Star algorithm and proposes a new heuristic function construction method. In our method, the influence of surrounding obstacles and target points on the ALV is also included in the estimated cost. The new valuation function is expressed as follows

\[
f(n) = g(n) + h'(n) + v(n)
\]

where \( f(n) \) denotes the total estimated cost from the start position through the current position to the end position, \( g(n) \) denotes the movement cost of the path from the starting position to the current position[22], we use \( h(n) = h'(n) + v(n) \) as improved heuristic function, \( h'(n) \) denotes the distance cost in traditional A-Star, and \( v(n) \) denotes the potential field cost and contains information about surrounding obstacles.

**Improved distance heuristics \( h'(n) \)**

The heuristic term \( h'(n) \) of the heuristic function has an important impact on the computational efficiency of the algorithm. Its weight in the total estimated cost is positively related to the search speed of the A-Star. The heuristic function is informative. The so-called informativeness of the heuristic function is the constraint condition when evaluating a node. The more information (that is, the more constraint conditions), the more nodes will be excluded.

For the selection of heuristic term \( h'(n) \), there are the following conclusions, where \( H(n) \) is the actual path length from node \( n \) to the target. If \( h'(n) < H(n) \), the algorithm can find the optimal path but there are many search nodes, and the search efficiency is very low; if \( h'(n) = H(n) \), this is the most ideal situation, and the algorithm can search for the optimal path with the highest efficiency; and if \( h'(n) > H(n) \), at this time, although the algorithm searches faster, it is difficult to obtain the optimal path.[24,25,22,26]

In addition, there are two extreme cases:

1) If \( h'(n) \gg H(n) \), the A-Star evolves into a breadth-first algorithm.
2) If \( h'(n) \ll H(n) \), the A-Star can be approximately regarded as the Dijkstra algorithm.

If a heuristic function that is closer to the actual loss can be found, this will improve the search efficiency. The function to estimate the path cost of the A-Star algorithm generally takes the Manhattan distance; in addition, there are Euclidean distance and diagonal distance.

Suppose there are two points \( P \) and \( Q \) in rectangular coordinates, \( P \) point coordinates are \((x_P, y_P)\), and \( Q \) point coordinates are \((x_Q, y_Q)\), the Manhattan distance between \( P \) and \( Q \) can be expressed as follows

\[
d_{m_{(P,Q)}} = |x_Q - x_P| + |y_Q - y_P|
\]

Euclidean distance can be expressed as follows

\[
d_{e_{(P,Q)}} = \sqrt{(x_Q - x_P)^2 + (y_Q - y_P)^2}
\]

The diagonal distance can be expressed as follows

\[
d_{d_{(P,Q)}} = d_s + d_v + (\sqrt{2} - 2)\min\{d_s, d_v\} \\
d_s = |x_Q - x_P|; d_v = |y_Q - y_P|
\]

Considering that the square grid map has eight moving directions, there is a diagonal movement, and the diagonal distance is closer to the actual path cost. To simply verify the influence of these three distance selection methods on the A-Star algorithm, we compared the search speed and the number of search grid nodes of the A-Star algorithm under different distance selection methods through experimental simulations. A verification experiment was carried out on a grid map of size 40 × 40 and the results are shown in Figure 4 and Table 1.
As the experimental results shown, because the estimated cost calculated by using the Manhattan distance method is greater than the cost in the actual path, the search time is the fastest and the number of searched grids is the least. However, the search range is the smallest and failed to search to the shortest path, the length of the path planned in this way is the longest. The estimated cost that is calculated using the Euclidean distance method is smaller than the cost in the actual path, the search range is the smallest and failed to search to the shortest path, the length of the path planned in this way is the longest. The estimated cost that is calculated using the diagonal distance method is moderate, compared to Manhattan distance and Euclidean distance.

The diagonal distance can find the shortest path while the running time is moderate. Therefore, we use the diagonal distance as the distance heuristic \( h'(n) \). The formula of \( h'(n) \) is as follows

\[
h'(n) = d_x + d_y + (\sqrt{2} - 2) \min\{d_x, d_y\}
\]

where \( d_x = |x_n - x_e|; d_y = |y_n - y_e| \), and \( x_n, y_n \) denote the abscissa and ordinate of the current position, \( x_e, y_e \) denote the abscissa and ordinate of the target. The diagonal distance is closer to the actual length. Therefore, as the diagonal distance as the distance heuristic \( h'(n) \).

### Obstacle heuristics \( v(n) \)

According to the principle of artificial potential field method, obstacles will produce repulsive force, which is decreases as the distance between obstacles and nodes increases, and the target position generates attraction, which is proportional to the distance from the node to the target. In this article, the environment map is constructed as a grid map, and the position of each grid is represented by the coordinates. Therefore, the calculation method of the potential field force is different from the artificial potential field method.

In the rasterized map environment, the calculation formula of repulsion is as follows, and the direction is from the obstacle \( i \) to the node \( k \)

\[
F_r(k, i) = \begin{cases} 
\left( \frac{1}{\rho(x_i, y_i)} - \frac{1}{d_0} \right)^2 \cdot \bar{v}_r, & \text{if } \rho(x_i, y_i) \leq d_0 \\
0, & \text{if } \rho(x_i, y_i) > d_0 
\end{cases}
\]

where \( \rho(x_i, y_i) = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \)

\[
\bar{v}_r = \frac{|x_k - x_i| + |y_k - y_i|}{\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}}
\]

### Numbers of searched grids

| Distance method | Running time (ms) | Planned path length (cm) | Numbers of searched grids |
|-----------------|-------------------|--------------------------|--------------------------|
| Manhattan       | 154               | 50.63                    | 162                      |
| Euclidean       | 197               | 48.77                    | 342                      |
| Diagonal        | 173               | 48.77                    | 231                      |

where \( k \) denotes the current position and the coordinates are \((x_k, y_k)\); \( i \) represents the \( i \)th grid containing obstacles and the coordinates are \((x_i, y_i)\), \( \bar{v}_r \) denotes the vector from the obstacle \( i \) to the node \( k \), and \( \bar{e}_x, \bar{e}_y \), respectively, represent the unit vector in the \( X \)-axis direction and the \( Y \)-axis direction. Considering that only those obstacles close to the ALV will affect the ALV’s working, and to reduce the amount of calculation, this article limits the scope of obstacles. Where \( \rho(x_i, y_i) \) is the distance from the current node \( k \) to the \( i \)th obstacle and \( d_0 \) is a distance threshold, in this article, \( d_0 = 10 \).

The formula for calculating the magnitude of attraction is shown in Eq. (11), and the direction is from the node \( k \) to the target point \( d \)

\[
F_a(k) = \left[ (x_k - x_d)^2 + (y_k - y_d)^2 \right] \cdot \bar{v}_r,
\]

\[
\bar{v}_r = \frac{|x_d - x_k| + |y_d - y_k|}{\sqrt{(x_k - x_d)^2 + (y_k - y_d)^2}}
\]

where \( k \) denotes the current position and the coordinates are \((x_k, y_k)\), \( d \) represents the target point and the coordinates are \((x_d, y_d)\), and \( \bar{v}_r \) denotes the vector from the node \( k \) to the target point \( d \).

After the attraction and repulsion are obtained, the combined force of the attraction and repulsion can be calculated to obtain the potential field force experienced by the current grid node \( k \). The calculation formula is as follows

\[
\tilde{F}(k) = \alpha F_a(k) + \beta \sum_i F_r(k, i)
\]

here \( \alpha \) and \( \beta \), respectively, represent the weight of attraction and repulsion. In this article, \( \alpha = 5 \) and \( \beta = 4 \).

The problem of local minima is inevitable in the artificial potential field method. When the calculated \( \tilde{F}(k) \) is 0, to avoid the algorithm return to the traditional A-Star algorithm, we will give up repulsion at that point. At this time, the calculation formula of \( \tilde{F}(k) \) is as follows

\[
\tilde{F}(k) = \alpha F_a(k)
\]
The calculation process is as follows: First calculate the unit vector from the point \( k \) to the adjacent node \( n \), as shown in the following equation

\[
\hat{e}_n = \frac{|x_n - x_k| \hat{e}_x + |y_n - y_k| \hat{e}_y}{\sqrt{(x_n - x_k)^2 + (y_n - y_k)^2}} \quad (14)
\]

Then calculate the angle between \( \vec{F}(k) \) and \( \hat{e}_n \), as shown in the following equation

\[
\theta_n = \langle \hat{e}_n, \vec{F}(k) \rangle \quad (15)
\]

The calculation formula of \( v(n) \) is as shown in the following equation

\[
v(n) = |\vec{F}(k)| \cos \theta_n \quad (16)
\]

Figure 5 shows the calculation diagram of the current position \( k \) when expanding its adjacent node \( n \). First, calculate the repulsion \( \vec{F}_r(k,S_i) \) of the obstacle \( S_1 \sim S_3 \) and the attraction \( \vec{F}_a(k) \) of the target position \( d \) and then calculate the resultant force \( \vec{F}(k) \) of all repulsive forces and attraction. When calculating the \( v(n) \) of the node \( n \), which is on the right of current position \( k \), first calculate the direction vector \( \hat{e}_n \) and then calculate the projection of the resultant force on the direction vector to obtain \( v(n) \). According to this method, calculate the \( v(n_i) \) of the eight adjacent expansion nodes of node \( k \) (excluding nodes that are obstacles). Then calculate \( h'(n), g(n), \) and \( f(n) \) to obtain the cost function of the extended node and perform a path search.

The algorithm steps are as follows:

**Step 1:** Create Open List \( O \) and Close List \( C \), the \( O \) is used to record the nodes that have been generated but have not been visited, and the \( C \) is used to record the nodes that have been visited; add the starting point to the \( O \).

**Step 2:** Add the node \( k \) with the smallest cost \( f \) in the \( O \) to the \( C \) and check whether the node is the target node, if it is the target, path planning ends; otherwise, go to the next step;

**Step 3:** Expand all adjacent nodes of node \( k \). Calculate the \( v(n), g(n), h'(n) \) and \( f(n) \) of all adjacent node \( n \). If node \( n \) is neither in the \( O \) nor in the \( C \), add the node \( n \) to the \( O \); if \( n \) is in the \( O \), but calculated \( f(n) \) is smaller than the old one in the \( O \), update the \( f(n) \) of node \( n \) in the \( O \), then reorder all nodes in the \( O \) according to the value of \( f(n) \); if \( n \) is in the \( C \), skip this point. Repeat these steps until all adjacent nodes of node \( k \) are expanded.

**Step 4:** Check whether the target point is in the \( C \) and whether the \( O \) is empty. If the target point is in the \( C \) or the \( O \) is empty, the algorithm ends; otherwise, repeat Step 2 ~ 4.

**Path smoothing**

To obtain a smooth path, commonly used methods include spline function and Bezier function. B-spline is an extension of Bezier curve. The polynomial degree of the B-spline can be independent of the number of control points, while the degree of the Bezier curve and the control points are closely related. Therefore, we use the quartic B-spline curves to smooth the planned path. Given \( n + 1 \) control points, the equation of the quartic B-spline curve is as follows

\[
C_4(u) = \sum_{i=0}^{n} N_{n,i}(u) P_i \quad (17)
\]

The mathematical expression of the basis function is as follows

\[
\begin{align*}
N_{0,4}(t) &= \frac{1}{24} (t^4 - 4 \times t^3 + 6 \times t^2 - 4 \times t + 1) \\
N_{1,4}(t) &= \frac{1}{24} (-4 \times t^4 + 12 \times t^3 - 6 \times t^2 - 12 \times t + 11) \\
N_{2,4}(t) &= \frac{1}{24} (6 \times t^4 - 12 \times t^3 - 6 \times t^2 + 12 \times t + 11) \\
N_{3,4}(t) &= \frac{1}{24} (-4 \times t^4 + 4 \times t^3 + 6 \times t^2 + 4 \times t + 1) \\
N_{4,4}(t) &= \frac{1}{24} t^4 
\end{align*}
\quad (18)
\]

**Experiment**

In this section, three sets of grid maps are designed for experimental verification, and each grid in the grid map represents the actual environment of 10 cm \( \times \) 10 cm. The computer configuration used in our simulation is i7-6700HQ with 16 GB RAM.

In the first map, the given start position coordinates are \((1,1)\), and the target position coordinates are \((20,17)\). The original planned path is shown in Figure 6(a), where the red is the path planned by our algorithm and the blue is the path planned by the A-STAR algorithm. Figure 6(b) shows the...
smoothed path. The comparison between A-Star and our proposed algorithm planning path is presented in Table 2. Among them, the time consumed by the algorithm is the average value obtained after 30 repeated experiments.

As shown in Figure 6, the A-Star algorithm passes between a piece of obstacles close to the endpoint. The improved A-Star algorithm avoids crossing obstacles and chooses to detour from the side. There is almost no unnecessary turning point in the planned path. From Table 2, we can see that the path lengths are both 323.1 cm, but the path planned by the A-Star requires a total turn of 540°/C14, while our proposed algorithm only needs a turn of 315°/C14. After the path smoothing, the path of A-Star is shortened to 294.4 cm, and the turning angle is reduced to 219.09°/C14. The path planned by our method is shortened to 303.3 cm, and the turning angle is reduced to 140.86°/C14. Since our algorithm is more complex, the calculation time is longer than that of A-Star. The A-Star algorithm took 3.8 ms, while our algorithm took 5.2 ms.

In the second set of experiments, we selected a more complex environment. The ALV needs to bypass obstacles many times to reach the given target point. The given start position coordinates are (24,17) and the target position coordinates are (4,16).

The original planned path is shown in Figure 7(a). As shown in Figure 7, the path planned by the improved A-Star is very straight, and there are few of unnecessary inflection points. Table 4 presents that the path lengths planned by both are 906.7 cm, but the path planned by the A-Star requires a total turn of 1080°, while our proposed algorithm only needs a turn of 540°. After the path smoothing, the path of A-Star is shortened to 851.6 cm, and the turning angle is reduced to 450.64°. The path planned by our method is shortened to 863.8 cm, and the turning angle is reduced to 270.16°. From the experimental results, we can see that our algorithm has better performance than A-Star. Since our algorithm is more complex, the calculation time is longer than that of A-Star. The A-Star algorithm took 99.6 ms, while our algorithm took 110.1 ms.

We also experimented our algorithm on reactive oxygen species. Use the Gmapping method to create a grid map as shown in Figure 8, and the map resolution is 5 cm/pixel and the puffing safety distance is set to 15 cm. We use a four-wheel drive car based on the Ackerman model and the size is 20 cm × 18 cm × 13 cm. The given start position coordinates are (3.6 m, 2.15 m) and the target position coordinates are (4 m, 15 m).

The original planned path is shown in Figure 8(a), where the red are the path planned by our algorithm and the blue are the path planned by the A-Star algorithm. Figure 8(b) shows the smoothed path. The comparison between A-Star

Table 2. Comparison results of A-Star algorithm and our method in 20 × 20 grid map.

| Method     | Path length (cm) | Total turning angle (°) | Time (ms) |
|------------|------------------|-------------------------|-----------|
| A-Star     | 323.1            | 540                     | 3.8       |
| A-Star (smoothing) | 294.4            | 219.09                  | 4.5       |
| Ours       | 323.1            | 315                     | 5.2       |
| Ours (smoothing) | 303.3            | 140.86                  | 6.3       |
and our proposed algorithm planning path is presented in Table 4.

As shown in Figure 8, the path planned by the improved A-Star is very straight and there are few of the unnecessary inflection points. Table 4 presents that the path length planned by ours is 1558.9 cm but the path planned by the A-Star requires a total turn of 1755°, while our proposed algorithm only needs a turn of 495°. After the path smoothing, the path of A-Star is shortened to 1516.5 cm, and the turning angle is reduced to 687.46°. The path planed by our method is shortened to 1534.9 cm, and the turning angle is reduced to 237.84°. From the experimental results, we can see that our algorithm has better performance than A-Star. Since our algorithm is more complex, the calculation time is longer than that of A-Star. The A-Star algorithm took 3332.7 ms, while our algorithm took 6657.4 ms. As the complexity of the map increases, the calculation time consumed also increases. As shown in Figure 8, the path planned by A-Star has many jagged turning points. This makes the path not smooth enough, which is not conducive to the motion control of ALV. Because the ALV has to adjust the angle at each inflection point, this will inevitably reduce the speed and affect the operating efficiency.

As the experimental simulation results shown that, compared with the traditional A-Star algorithm, our improved algorithm has lower turning loss, straighter path. The path length planned by our algorithm is similar to the A-Star algorithm. But, the total turning angle was reduced by 4060% compared with A-Star. However, our algorithm is more complex, and the calculation time will be longer than A-Star. Especially in the environment with many obstacles, the calculation time will increase more obviously, and the efficiency of the algorithm still needs to be further improved. Our algorithm reduces the inflection points, allowing the ALV to easier adjust its posture and realize the autonomous movement of the ALV. When the ALV encounters a turn during the operation, it will reduce the speed and adjust the angle to ensure safe operation. In this process, the efficiency of ALV will be reduced, the control difficulty will increase, and a large part of the time will be consumed. Although our method increases the cost of time spent in planning, it improves the efficiency of operation and reduces the running time and control difficulty of ALV.

---

**Table 3.** Comparison results of A-Star algorithm and our method in 40 x 40 grid map.

| Method       | Path length (cm) | Total turning angle (°) | Time (ms) |
|--------------|------------------|-------------------------|-----------|
| A-Star       | 906.7            | 1080                    | 99.6      |
| A-Star (smoothing) | 851.6          | 450.64                  | 103.2     |
| Ours         | 906.7            | 540                     | 106.3     |
| Ours (smoothing) | 863.8           | 270.16                  | 110.1     |

**Table 4.** Comparison results of A-Star algorithm and our method in ROS environment.

| Method       | Path length (cm) | Total turning angle (°) | Time (ms) |
|--------------|------------------|-------------------------|-----------|
| A-Star       | 1552.7           | 1755                    | 3332.7    |
| A-Star (smoothing) | 1516.5        | 687.46                  | 3384.6    |
| Ours         | 1558.9           | 495                     | 6657.4    |
| Ours (smoothing) | 1534.9          | 237.84                  | 6712.3    |

ROS: reactive oxygen species.
Conclusion

In this article, a new heuristic function design method is proposed to improve the widely used A-Star algorithm. The influence of surrounding obstacles on the operation of the trolley is considered. The artificial potential field force is added to the heuristic function to strengthen the search direction information of the algorithm, which can improve the performance of the A-Star algorithm. Compared with the traditional heuristic function that only considers distance, our method takes into account the surrounding obstacle information and is more suitable for environments with many obstacles; the improved algorithm plans a path that is straighter, effectively reducing unnecessary inflection points, and making vehicle can adjust the angle as little as possible, which reduces path loss and is more conducive to the motion control of ALV.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by the Natural Science Foundation of China under Grant 61801359 and Grant 61571345.

ORCID iD

Jing Zhang https://orcid.org/0000-0002-8495-2804

Supplemental Material

Supplemental material for this article is available online.

References

1. Erke S, Bin D, Yiming N, et al. An improved A-Star based path planning algorithm for autonomous land vehicles. Int J Adv Robot Syst 2020; 17(5): 1729881420962263.
2. QI X, Huang J, and Cao J. Path planning for unmanned vehicle based on improved A* algorithm. J Comput Appl 2020; 40(7): 2021–2027.
3. Zhao X, Wang Z, Huang C, et al. Mobile robot path planning based on an improved A* algorithm. Robot 2018; 40(6): 903–910.
4. Wu Z, Qiu K, and Gao H. Driving policies of V2X autonomous vehicles based on reinforcement learning methods. IET Intell Transp Syst 2020; 14(5): 331–337.
5. Gao H, Su H, Cai Y, et al. Trajectory prediction of cyclist based on dynamic Bayesian network and long short-term memory model at unsignalized intersections. Sci China Inf Sci 2021; 64(7): 1–13.
6. Tsitsiashvili GS and Losev AS. Application of the Floyd algorithm to the asymptotic analysis of networks with unreliable ribs. *Autom Remote Control* 2008; 69(7): 1262–1265.

7. Wang H, Yu Y, and Yuan Q. Application of Dijkstra algorithm in robot path-planning. In *2011 Second international conference on mechatronic automation and control engineering*, Inner Mongolia, China, 15–17 July 2011, pp. 1067–1069. IEEE.

8. Dучоň F, Babinec A, Kajan M, et al. Path planning with modified a star algorithm for a mobile robot. *Procedia Eng* 2014; 96: 59–69.

9. Mazzeo S and Loiseau I. An ant colony algorithm for the capacitated vehicle routing. *Electron Notes Discrete Math* 2004; 18: 181–186.

10. Tharwat A, Elhoseny M, Hassanien AE, et al. Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm. *Clust Comput* 2019; 22(2): 4745–4766.

11. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. In Cox IJ and Wilfong GT (eds) *Autonomous robot vehicles*. New York: Springer, 1986, pp. 396–404.

12. Mora MC and Tornero J. Path planning and trajectory generation using multi-rate predictive artificial potential fields. In 2008 IEEE/RSJ international conference on intelligent robots and systems, Nice, France, 22–26 September 2008, pp. 2990–2995. IEEE.

13. Zhang HM, Li ML, and Yang L. Safe path planning of mobile robot based on improved A* algorithm in complex terrains. *Algorithms* 2018; 11(4): 44.

14. Hongbin W, Pengheng Y, Wei Z, et al. Mobile robot path planning based on improved A* algorithm and dynamic window method. *J Robot* 2020; 42(3): 346–353.

15. Chen RM, Hsieh FR, and Wu DS. Heuristics based ant colony optimization for vehicle routing problem. In 2012 7th IEEE conference on industrial electronics and applications (ICIEA), Singapore, 18–20 July 2012, pp. 1039–1043. IEEE.

16. Wang W, Zhang P, Liang C, et al. Design, path planning improvement and test of a portable massage robot on human back. *Int J Adv Robot Syst* 2018; 15(4): 1729881418786631.

17. Zhang J, Zhang F, Liu Z, et al. Efficient path planning method of USV for intelligent target search. *J geovis spat anal* 2019; 3(2): 1–9.

18. Wu TY, Xu JH, Liu JY, et al. Research of cross-country path planning based on improved A* algorithm. *Jisuanji Yingyong Yanjiu* 2013; 30(6): 1724–1726.

19. Lin M, Yuan K, Shi C, et al. Path planning of mobile robot based on improved A* algorithm. In 2017 29th Chinese control and decision conference (CCDC), Chongqing, China, 28–30 May 2017, pp. 3570–3576. IEEE.

20. Wang D. Indoor mobile-robot path planning based on an improved A* algorithm. *J Tsinghua Uni Sci Technol* 2012; 52(8): 1085–1089.

21. Hart PE, Nilsson NJ, and Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans Syst Sci Cybern* 1968; 4(2): 100–107.

22. Li D, Liu M, Zhang J, et al. An improved A* algorithm applicable for campus navigation system. In 2015 International conference on network and information systems for computers, Wuhan, China, January 23 2015 to January 25 2015. IEEE, pp. 588–591.

23. Tang X and Chen F. Robot path planning algorithm based on Bi-RRT and potential field. In 2020 IEEE international conference on mechatronics and automation (ICMA), Beijing, China, 13 October 2020 to 16 October 2020. IEEE, pp. 1251–1256.

24. Leach AR and Lemon AP. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. *Proteins* 1998; 33(2): 227–239.

25. Seet BC, Liu G, Lee BS, et al. A-Star: a mobile ad hoc routing strategy for metropolis vehicular communications. In *International conference on research in networking*, Prague, Czech Republic, 21–25 May. Springer, pp. 989–999.

26. Zheng T, Xu Y, and Zheng D. AGV path planning based on improved A-Star algorithm. In 2019 IEEE 3rd advanced information management, communicates, electronic and automation control conference (IMCEC), Chongqing, China, October 11–13, 2019. IEEE, pp. 1534–1538.