

Article

# Research on 3D Path Planning of Quadrotor Based on Improved A\* Algorithm

Wei Zheng <sup>1,2</sup> , Kaipeng Huang <sup>3</sup>, Chenyang Wang <sup>1,2</sup>, Yang Liu <sup>3,\*</sup>, Zhiwu Ke <sup>1,2</sup>, Qianyu Shen <sup>3</sup> and Zhiqiang Qiu <sup>1,2</sup>

<sup>1</sup> Science and Technology on Thermal Energy and Power Laboratory, Wuhan 430064, China

<sup>2</sup> Wuhan Second Ship Design and Research Institute, Wuhan 430064, China

<sup>3</sup> School of Mechanical Engineering & Electrical Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

\* Correspondence: ly2015@uestc.edu.cn

**Abstract:** Considering the complexity of the three-dimensional environment and the flexibility of the quadrotor aircraft, using the traditional A\* algorithm for global path planning has the disadvantages of less search direction, more expanded nodes, and a longer planning path. Therefore, an improved A\* algorithm is proposed, which is improved from two aspects. Firstly, a two-layer extended neighborhood strategy is proposed, which can increase the search direction and make better use of the flexibility of the aircraft. Secondly, the heuristic function is improved to make the heuristic function value closer to the actual planning path distance, which can reduce the expansion nodes and optimize the planning path. Finally, the path planning simulation of the improved A\* algorithm is carried out and the results show that the path planned by the improved algorithm is shorter and the expanded nodes are fewer, which can guide the quadrotor to reach the destination better.

**Keywords:** quadrotor; A\* algorithm; heuristic algorithm; neighborhood strategy



**Citation:** Zheng, W.; Huang, K.; Wang, C.; Liu, Y.; Ke, Z.; Shen, Q.; Qiu, Z. Research on 3D Path Planning of Quadrotor Based on Improved A\* Algorithm. *Processes* **2023**, *11*, 334. <https://doi.org/10.3390/pr11020334>

Academic Editors: Qiuye Sun, Rui Wang and Zhengmao Li

Received: 22 December 2022

Revised: 2 January 2023

Accepted: 5 January 2023

Published: 19 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have been widely used in many fields [1]. They have significant advantages in the field of public safety, disaster management and rescue operations under particular circumstances. UAVs can transport medical supplies and daily necessities to inaccessible disaster areas. In case of gas infiltration, avalanche and other disasters, UAVs can assist ground personnel in searching for missing persons and related operations, which keeps relevant personnel safe.

The quadrotor aircraft is a type of UAV and its benefits include its small size, low price, and ease of carrying for portable devices such as sensors and cameras [2,3]. At the same time, it has easy maintenance characteristics, hovering capability and vertical take-off and landing [4].

As the load of a quadrotor aircraft is limited, it can only carry a limited battery, which results in a generally short battery life [5]. If a quadrotor aircraft wants to complete its mission efficiently in a limited time, it is necessary to plan the flight path of the quadrotor. Finding a safe flight path with low energy consumption and a short flight time with the assumption of completing the mission is the primary goal of path planning [6]. In the existing literature [7–9], most of the algorithm analysis of quadrotor path planning is applied in a two-dimensional plane, but the three-dimensional environment is usually unstructured and uncertain, which is much more complicated than the two-dimensional environment. In this paper, 3D path planning will be carried out according to the energy consumption factors and the feasibility of the flight path of the quadrotor aircraft.

Path planning is to find a continuous curve that meets the constraint conditions from the starting node to the target node in the configuration space [10]. This curve has nothing to do with time variables and can be composed of multiple trajectory segments. Unlike

two-dimensional path planning, path planning in a three-dimensional environment will produce more uncertainties. Three-dimensional path planning usually needs to establish a map information model and then the actual map information is reflected by the grid map and topology map [11,12]. The path planning algorithm with the lowest time and cost is used to find the best path and complete the path planning. In the flight path planning algorithm, the configuration space is dispersed into a set of flight path nodes, and a set of node sets that minimize the flight path cost are found. At present, the commonly used algorithms are the artificial potential field method [13], A\* algorithm, genetic algorithm, ant colony algorithm and particle swarm optimization algorithm [14–17]. Among them, the A\* algorithm is easy to implement and has less time complexity because of its fast convergence speed.

In this paper, the concrete method of building a 3D grid map according to the actual map is introduced, how the traditional A\* algorithm works is explained and its shortcomings are analyzed. On this basis, the A\* algorithm is improved. Considering the flexibility and randomness of quadrotor aircraft steering, a two-layer extended neighborhood strategy is proposed. In addition, the heuristic function is improved to make its value closer to the actual planned path distance.

## 2. Map Information

### 2.1. Establish a Grid Map

The grid method works by dividing the complex environment into many small squares, and these small squares are grids.

Before rasterizing the map, the grid size should be determined first. If the grid is large, it cannot accurately describe the environment. A small grid is closer to the real environment but it will lead to a high resolution of the environment and it will also increase the search difficulty and time of the algorithm. Therefore, the grid size should not only satisfy the free movement of objects in the grid but also reduce the search difficulty and time as much as possible. After rasterizing the map, “0” and “1” are used to represent free grids and non-free grids. If there are obstacles in the grids, the intersection of the row and column representing the current position in the matrix expression is set to “1”, otherwise, it is set to “0”. By rasterizing the map, the problem of path planning on the map can be transformed into finding an optimal path in a network cell, which is composed of the grid where the starting node is located, the grid where the target node is located and a series of free grids in the middle [18].

The obstacles in that three-dimensional environment are respectively projected to the  $xy$  plane,  $xz$  plane,  $xy$  plane, and calculate the projected area of obstacles in three projection planes, respectively. The ratio of the estimated area of obstacles to the entire plane determines the size of the grid. The calculation method is:

- (1) All irregular obstacles are projected and filled into polygons in the plane and the polygons are divided into several disjointed triangles;
- (2) The area of the  $i$ th triangle is calculated as follows:

$$S_i = 0.5ab \sin \theta \quad (1)$$

where  $a$  and  $b$  are the lengths of any two sides of the triangle, and  $\theta$  is the angle between the two sides;

- (3) Figure out the sum of the areas of all triangles, which is  $S_{total}$ ;
- (4) Calculate the grid size as follows:

$$L = \begin{cases} L_{max} & L > L_{max} \\ k_l \frac{S_{total}}{S_p} L_{max} & L_{min} < L < L_{max} \\ L_{min} & L < L_{min} \end{cases} \quad (2)$$

where  $k_l$  is the grid adjustment factor;  $S_p$  is the area of the whole plane;  $L_{max}$  is the maximum side length of the grid; and  $L_{min}$  is the minimum length of the grid. Ac-

According to Equation (1), the lower the proportion of obstacles in the plane, the smaller the grid size and the smaller the complexity of path planning. On the contrary, the higher the proportion of obstacles, the larger the grid size and the more complicated the path planning.

## 2.2. Grid Identification

### (1) Serial number method.

In a grid map, each grid corresponds to a specific location information of the map, therefore each grid has a unique location attribute. Therefore, all grids can be distinguished by numbering them. According to the uniqueness of grid information, the numbering is also unique. In a 2D grid map of size  $n \times n$ , the length of each grid is defined as 1 and the grid in the lower left corner is the origin and numbered as 1, the unit number is increased from left to right until the rightmost grid that is numbered as  $n$ , and the number in the next row starts at  $n + 1$  and increases to the right to  $2n$  and so on until the whole 2D grid map is numbered. Similarly, in a 3D grid map of size  $n \times n \times n$ , the 3D grid map is split into  $n$  2D grid map of size  $n \times n$ . The numbering method of the lowest 2D grid map is the same as that of the 2D grid map mentioned above, and the number in the lower-left corner of the next layer 2D grid map starts at  $n^2 + 1$ . After that, the numbering method is the same as that of a 2D grid map and so on until the whole 3D grid map is numbered.

### (2) Coordinate method.

Establish a rectangular coordinate system in the grid map, set the lower left corner boundary of the grid map as the coordinate origin, and define the  $x, y, z$  axes according to the right-hand rule. The length of each grid is defined as 1, the boundary of each grid is marked as a coordinate value, and the lengths of the three coordinate axes are  $n$ . The calculation formula for converting the coordinate expression into the serial number expression is as follows:

$$S = x + (y - 1)n + (z - 1)n^2 \quad (3)$$

where  $S$  is the number corresponding to the current grid;  $x, y, z$ , respectively, are the  $x, y, z$  coordinates of the current grid. In the same way, the calculation formula for converting serial number expressions into coordinate expressions can be deduced.

## 2.3. Grid Expansion Treatment of Obstacles

Because the shape of obstacles is irregular in the map environment, the shapes projected on the  $xy$  plane,  $xz$  plane,  $xy$  plane are also irregular. After gridding, in order to reduce the computational complexity of the subsequent route planning algorithm and keep a certain safe distance between quadrotor aircraft and obstacles as far as possible, it is necessary to expand the obstacle grid. The processing method is that if the obstacles fill the whole grid, there is no need to process and the current grid is used as the obstacle grid. However, if there is an obstacle in a grid and the obstacle does not occupy the whole grid, the whole current grid is regarded as an obstacle. In short, all grids with obstacles are treated as obstacle grids, as shown in Figure 1.

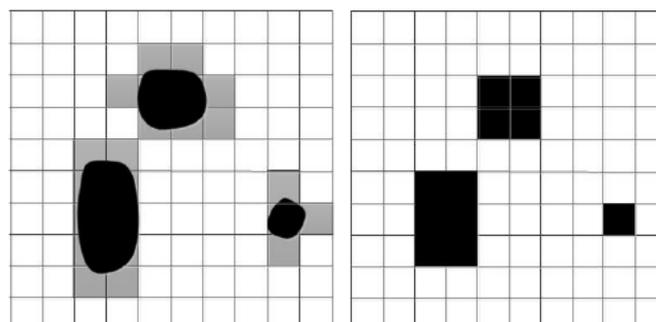


Figure 1. Grid expansion treatment.

### 3. Improved Path Planning Based on the A\* Algorithm

#### 3.1. Traditional A\* Algorithm

In 1968, the A\* algorithm was proposed by Peter Hart et al. The A\* algorithm is a heuristic algorithm, therefore it has high search efficiency [19,20]. The principle of the heuristic algorithm is to search according to preset information in the pathfinding process. This search is directional so that the pathfinding direction is concentrated in the preset range and a large number of unnecessary search paths are reduced. When planning the path, all grid nodes in the map have an expected distance. In the process of expanding nodes, the node with the smallest expected distance to expand is selected.

In the A\* algorithm, the estimated value of each node can be calculated as follows:

$$F(n) = G(n) + H(n) \quad (4)$$

where  $F(n)$  is the total cost of starting at the beginning node and passing through the  $n$  node to the target node;  $G(n)$  is the distance traveled from the beginning node to the  $n$  node; and  $H(n)$  is the expected shortest distance from the  $n$  node to the destination. There are many kinds of methods to calculate  $H(n)$  such as the Manhattan distance and the Euclidean distance. The Euclidean distance between two nodes is performed by actually connecting the two nodes with a line segment and calculating the length of this line segment. In a 3D environment, the calculation method is as follows:

$$Dist_{euc}(x, y, z) = \sqrt{(x_n - x_1)^2 + (y_n - y_1)^2 + (z_n - z_1)^2} \quad (5)$$

According to the Euclidean distance calculation method, the object can move in any direction in a 2D or 3D environment.

These are the steps of the A\* algorithm:

- (1) Record the starting position and initialize the open list and close list. In the A\* algorithm, the open list is to record the path nodes to be visited, that is, the nodes that were recorded but not expanded. The close list is used to record the nodes that were traversed;
- (2) Search open list. If there are no nodes in it, this means that there is no viable path. At this time, an error is returned to the console and the main process of the algorithm ends. If there are nodes that have not been traversed and there is a target node in these nodes, it means that the target node has been found. At this time, the search results are returned to the console and the main process of the algorithm is finished. If there are nodes that have not been traversed and the target node does not exist in these nodes, the process moves on to the next step;
- (3) The node to be explored is the one in the open list with the minimum  $F$  value and the node is moved from the open list to the close list;
- (4) Calculate all nodes within the current node extension range. For an adjacent node, if it does not exist in the above two lists, it will be added to the open list. If it exists in the open list and its  $F$  value is smaller when passing through the current node, the current node is its parent node;
- (5) Return to step (2).

#### 3.2. Improved Neighborhood Expansion Search Strategy

As the traditional A\* algorithm only has eight search directions at each node in the pathfinding process, it does not consider the flexibility of quadrotor aircraft steering and the planned path is not the shortest path. Therefore, the neighborhood expansion search strategy is improved to solve the above problems.

In the grid map, the node searched by the A\* algorithm in the pathfinding process is the center of the grid. As shown in Figure 2, the current node that needs to be expanded is located at the center node of the example diagram and the eight adjacent red nodes constitute the traditional A\* algorithm neighborhood of the current node, which is defined as the first-layer extended neighborhood. At this node, there are eight expansion directions

and the angle between the expansion directions is  $0.25\pi$ . To increase the expansion direction of the current node, the second-layer extended neighborhood is defined, and the second-layer extended neighborhood adds eight blue nodes based on the first-layer extended neighborhood, meaning that there are 16 expansion directions. After adding the second-layer extended neighborhood, there is no need to search for redundant grey nodes in the same direction when expanding nodes.

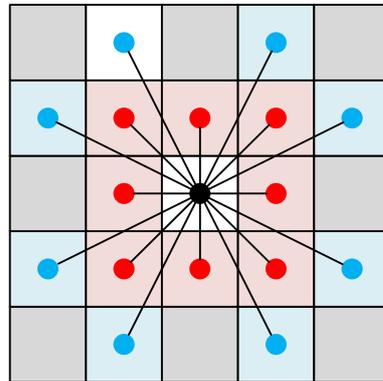


Figure 2. Exploration nodes of a two-layer extended neighborhood.

When the traditional A\* algorithm expands nodes, it only needs to judge whether the eight search nodes adjacent to the current node are accessible, after that, it calculates the estimated value of the accessible nodes. When the A\* algorithm based on the two-layer extended neighborhood expands the current node, it is necessary to determine whether the 16 search nodes are accessible and whether the grid passing between the current node and the search node is an obstacle grid or not. If there is an obstacle grid, it is judged that the search node is not accessible. Otherwise, it is considered accessible.

First, judge the neighborhood nodes of the second layer. As shown in Figure 3, the current node is the black node. The nodes that need to be searched are the purple nodes. Let the current node coordinate be  $(x_n, y_n)$ , the distance between grid nodes is 1, and then the coordinate of the node to be searched is  $(x_n + 1, y_n + 2)$ . It can be seen from Figure 3 that the current node passes through two grid nodes to the node to be searched, and the two grid nodes' coordinates are  $(x_n, y_n + 1)$  and  $(x_n + 1, y_n + 1)$  to judge whether the node to be searched and the passing nodes are located in the obstacle grid. If there is an obstacle grid, the search node is not considered. Otherwise, its estimated value is calculated.

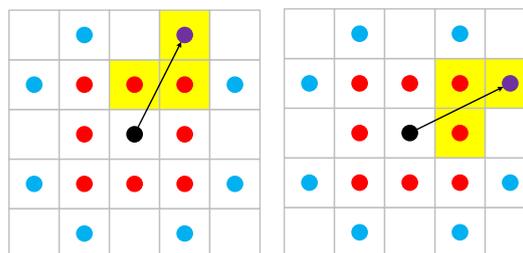


Figure 3. Expansion directions of two-layer extended neighborhood.

When there is no specific search node, let the coordinate of the node to be searched be  $(x_i, y_i)$  and the slope between the node to be searched and the current node can be calculated as follows:

$$k = \left| \frac{y_i - y_n}{x_i - x_n} \right| \tag{6}$$

where  $k = 2$  or  $k = 0.5$ . The coordinates of the two grid nodes passing through from the current node to the node to be searched can be obtained in Table 1.

**Table 1.** Slope and coordinates of passing through grid nodes.

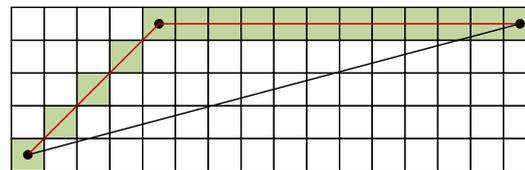
Slope	Coordinate of First Node	Coordinate of Second Node
0.5	$\frac{x_n+x_i}{2}, \frac{y_n+y_i}{2} + 0.5$	$\frac{x_n+x_i}{2}, \frac{y_n+y_i}{2} - 0.5$
2	$\frac{x_n+x_i}{2} - 0.5, \frac{y_n+y_i}{2}$	$\frac{x_n+x_i}{2} + 0.5, \frac{y_n+y_i}{2}$

**3.3. Improved Heuristic Function**

The cost function of the A\* algorithm consists of the shortest path that was traveled and the predictive distance, which are  $G(n)$  and  $H(n)$ . The value of  $G(n)$  is related to the explored parent node and the heuristic function  $H(n)$  only calculates the distance from the current node to the target node. It is the crucial information of the A\* search algorithm that guides the direction in the pathfinding process. Its guidance makes the algorithm not need to traverse all the extended nodes, which reduces the algorithm’s computation. Therefore, the performance of the A\* algorithm is significantly impacted by the choice of heuristic function.

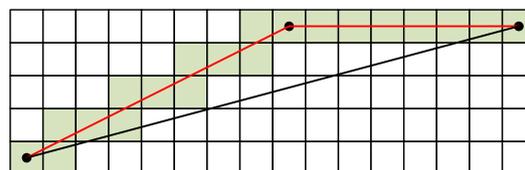
In the calculation process, due to the heuristic function of the traditional A\* algorithm being calculated by the Euclidean distance formula, there is a big difference between the calculated distance and the actual distance. If  $H(n)$  is always less than or equal to the actual distance, the A\* algorithm can definitely find the best path. If  $H(n)$  is greater than the actual distance, the path found by the A\* algorithm may not be the best.

As shown in Figure 4, the red line segment represents the path planned by the A\* algorithm, and the black line segment represents the distance calculated by the heuristic function using the Euclid formula. The traditional A\* algorithm has eight extended neighborhood directions, therefore the path distance actually planned by the A\* algorithm is usually greater than that calculated by the Euclid formula. They will be equal when the starting node and the target node are horizontal, vertical, or at an angle of 45 degrees with the horizontal.



**Figure 4.** A\* algorithm path planning based on single-layer extended neighborhood.

In this paper, the two-layer extended neighborhood method is adopted and the path planned by this method is shown in Figure 5. There is also the case that the actual planned path is greater than the calculated Euclidean distance. Because the calculated value  $H(n)$  is less than the actual distance, the algorithm’s pathfinding approach will involve a significant number of extended nodes.



**Figure 5.** A\* algorithm path planning based on two-layer extended neighborhood.

Therefore, aiming at the above problem, the heuristic function is improved, and the improved heuristic function is calculated by subsection. The current and target nodes’ coordinates are, respectively,  $(x_n, y_n, z_n)$  and  $(x_t, y_t, z_t)$ . The calculation method is as follows:

(1) The difference between the current node and the target node on the three axes is calculated and the absolute value of the difference is taken. The axis where the minimum absolute value is located is called the minimum axis and the axis where the maximum absolute value is located is called the maximum axis. The first plane passes through the

current node and is parallel to the minimum and maximum axes. The second plane passes through the target node and is perpendicular to the minimum axis;

(2) In the first plane, it can just reach the second plane by moving a  $d_1$  distance from the current node close to the target node. The position where the current node moves to the second plane is the intermediate node. The first distance can be calculated as follows:

$$d_1 = \sqrt{5} \min(|x_t - x_n|, |y_t - y_n|, |z_t - z_n|) \quad (7)$$

(3) In the second plane, the path planning of the intermediate node and the target node degenerates into a two-dimensional environment. The difference between the intermediate node and the target node on the two coordinate axes in the two-dimensional plane is calculated, and the absolute value of the difference is taken, respectively. The maximum absolute value is  $V_{max}$  and the minimum absolute value is  $V_{min}$ . The second distance and third distance can be calculated as follows:

$$\begin{cases} d_2 = \sqrt{5}V_{min} \\ d_3 = V_{max} - 2V_{min} \end{cases} \quad (8)$$

Therefore, the intermediate node can reach the target node by moving  $d_2 + d_3$  distance;

(4) To sum up, the current node can move these three distances to reach the target node. The value of the improved heuristic function  $H(n)$  is the sum of these three distances.

#### 4. Simulation and Analysis

The simulation platform is MATLAB, in which a three-dimensional grid map is built. The path planning of the A\* algorithm before and after the improvement is simulated and compared.

This paper's improvement of the A\* algorithm is divided into two parts. The first part is to increase the neighborhood expansion direction and the second part is to modify the calculation method of the heuristic function. The increase in the extended neighborhood makes the quadrotor have more flight directions. At the same time, it explores a shorter path, so modifying the heuristic function according to the flight direction is necessary. Therefore, the algorithm with the extended neighborhood is simulated and analyzed first, and then the algorithm with a modified heuristic function is simulated and analyzed on this basis. In all simulation diagrams, the red line segment is the path planned by the improved algorithm, while the black line segment is the path planned by the traditional A\* algorithm. The height of each grid is 1, where the yellow grid is an obstacle.

The path planning simulation diagram before and after increasing the extended neighborhood is shown in Figure 6. It can be seen that the planning path from the starting node (1,1,1) to the node (9,3,1) is shorter after increasing the extended neighborhood. After increasing the extended neighborhood directions, the moving direction is not limited to eight directions, which makes some turns more flexible, the planned path smoother, and the path distance shorter.

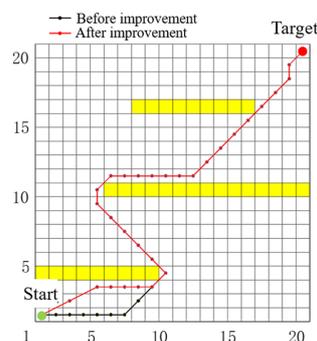


Figure 6. Path planning before and after increasing the extended neighborhood.

The path planning simulation diagram before and after the improved heuristic function is shown in Figure 7. As the estimated value of the path calculated by the improved heuristic function is more accurate, it can be seen from Figure 7 that the improved A\* algorithm has a relatively obvious reduction in the number of extended nodes. Moreover, the path length planned by the algorithm was further reduced compared with the path before the improvement.

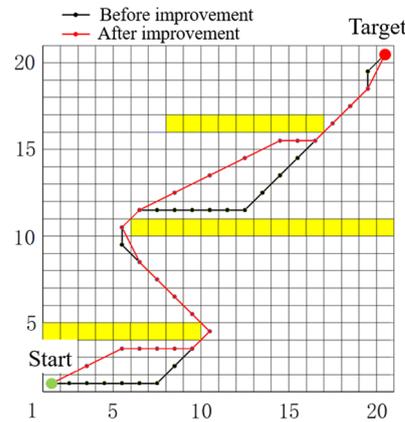


Figure 7. Path planning before and after improving the heuristic function.

Considering that the actual three-dimensional environment of quadrotor aircraft is relatively complex, a complex three-dimensional environment is established to simulate the real environment, and the performance of the improved algorithm is analyzed. Combined with the above advantages of increasing an extended neighborhood and improving the heuristic function, path planning is carried out in a 3D environment before and after the improved A\* algorithm. The simulation result is shown in Figure 8.

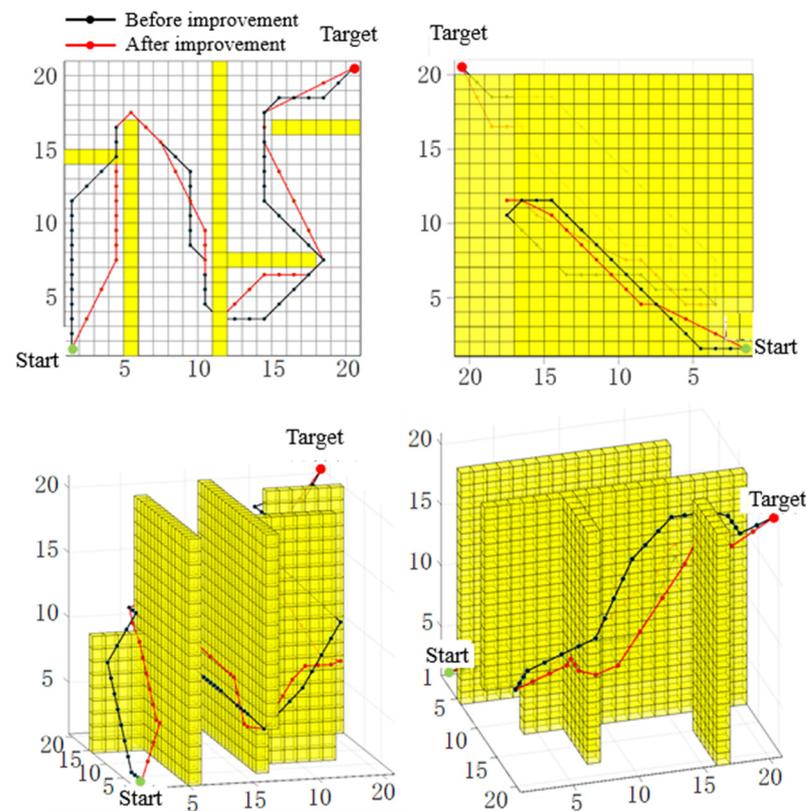
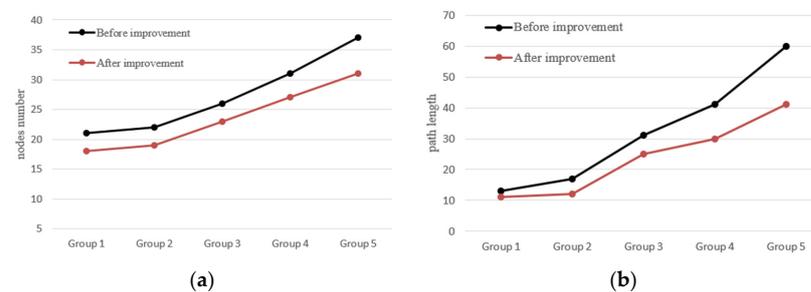


Figure 8. Path planning in a 3D environment before and after the improved A\* algorithm.

In Figure 8, it is found that in the complex three-dimensional environment, the traditional A\* algorithm, due to the limitation of its search direction and the inaccurate description of the estimated distance by heuristic function, leads to more planned path breaks and more path cost consumption. It is intuitive to see that the improved algorithm has improved the original problem, further shortened the length of path planning, effectively reducing the breakpoints, and obviously reduced the cost of quadrotor flight.

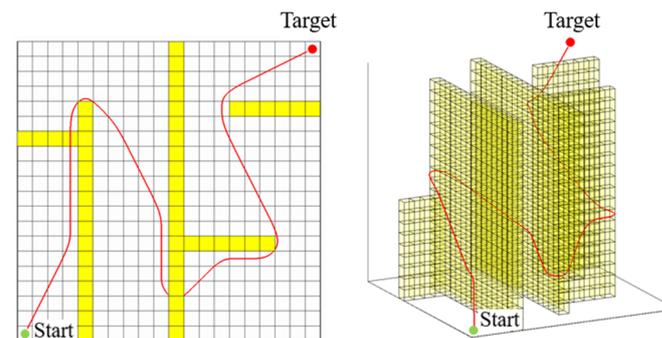
In this paper, the improved A\* algorithm is simulated in several groups of three-dimensional environmental maps and compared with the traditional A\* algorithm according to the results. The analysis results are as follows.

From several groups of experiments, it can be seen in Figure 9 that when the improved A\* algorithm expands nodes, the search range is further refined, and the total number of expanded nodes decreases. At the same time, as the path direction increases, the total length of the planned path decreases. Therefore, the path planned by the improved A\* algorithm is shorter and the number of extended nodes is less.



**Figure 9.** Comparison of path planning effect before and after A\* algorithm improvement, where (a) is the comparison of the number of path nodes and (b) is the path length comparison.

After the improved A\* algorithm is used to plan the path of the quadrotor, a set of node sets of the path from the starting node to the target node can be obtained. Based on the node sets of the path, the B-spline interpolation function is simulated and its smoothing performance is analyzed. The path used in the simulation is the path planning in the 3D environment of Figure 8, and the simulation result is shown in Figure 10.



**Figure 10.** Smooth trajectory after B-spline function interpolation.

From the simulation results, it can be seen that the original path is effectively and obviously smoothed by using the B-spline interpolation function, which eliminates the sudden change of direction in the flight of the quadrotor, improves the flight path of the quadrotor, and facilitates the trajectory tracking control of the quadrotor.

## 5. Conclusions

Firstly, this paper introduces the concrete method of building a 3D grid map according to the actual map, which provides environmental information for the subsequent path-planning algorithm of the quadrotor. Then, the traditional A\* algorithm is introduced and the shortcomings of the traditional algorithm are analyzed. On this basis, the A\* algorithm

is improved. Considering that the steering of the quadrotor is flexible and random, a two-layer extended neighborhood strategy is proposed, which can further reduce the shortest path by increasing the search direction. Secondly, the heuristic function of the A\* algorithm adopts the Euclidean distance formula. In most cases, the estimated value of the heuristic function is larger than the actual distance, which leads to the addition of additional extended nodes. Based on this, the heuristic function is improved to make its value closer to the actual planned path distance. MATLAB is used to simulate the path planning of the aircraft. Compared with the path planning effect of the original A\* algorithm and the improved A\* algorithm, it can be clearly seen that the improved A\* algorithm plans a shorter path, has fewer extended nodes and has better results. B-spline is used to smooth the path, and finally, a smooth trajectory is obtained. Therefore, the improved algorithm can guide the quadrotor to reach the destination more economically and conveniently, which proves that the method has a certain application value.

**Author Contributions:** Conceptualization, Y.L. and W.Z.; methodology, Z.K.; software, K.H.; validation, C.W., Y.L. and K.H.; formal analysis, W.Z.; investigation, W.Z. and Z.Q.; resources, C.W.; data curation, Q.S.; writing—original draft preparation, W.Z.; writing—review and editing, K.H.; visualization, C.W.; supervision, W.Z.; project administration, Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by pre research project of Equipment Development Department of the Military Commission: 1126190601A.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Wang, X.; Tan, G.-Z.; Lu, F.-L.; Zhao, J.; Dai, Y.-S. A Molecular Force Field-Based Optimal Deployment Algorithm for UAV Swarm Coverage Maximization in Mobile Wireless Sensor Network. *Processes* **2020**, *8*, 369. [[CrossRef](#)]
2. Yuan, L.; Reardon, C.; Warnell, G.; Loianno, G. Human Gaze-Driven Spatial Tasking of an Autonomous MAV. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1343–1350. [[CrossRef](#)]
3. Wang, R.; Sun, Q.; Ma, D.; Liu, Z. The Small-Signal Stability Analysis of the Droop-Controlled Converter in Electromagnetic Timescale. *IEEE Trans. Sustain. Energy* **2019**, *10*, 1459–1469. [[CrossRef](#)]
4. Wang, X.; Cai, L. Aircraft navigation based on differentiation–integration observer. *Aerosp. Sci. Technol.* **2017**, *68*, 109–122. [[CrossRef](#)]
5. Wang, R.; Sun, Q.Y.; Hu, W.; Li, Y.S.; Ma, D.Z.; Wang, P. SoC-Based Droop Coefficients Stability Region Analysis of the Battery for Stand-Alone Supply Systems with Constant Power Loads. *IEEE Trans. Power Electron.* **2021**, *36*, 7866–7879. [[CrossRef](#)]
6. Kim, J. Fast Path Planning of Autonomous Vehicles in 3D Environments. *Appl. Sci.* **2022**, *12*, 4014. [[CrossRef](#)]
7. Zhou, J.; Cheng, Y.; Du, H.; Wu, D.; Zhu, M.; Lin, X. Active finite-time disturbance rejection control for attitude tracking of quad-rotor under input saturation. *J. Frankl. Inst. Eng. Appl. Math.* **2020**, *357*, 11153–11170. [[CrossRef](#)]
8. Xu, S.; Dong, H.; Qin, Z.; Han, Y.; Gong, D.-W.; Zou, S.-L.; Wei, C.; Zhao, F. Parallel processing of radiation measurements and radiation video optimization. *Opt. Express* **2022**, *30*, 46870. [[CrossRef](#)] [[PubMed](#)]
9. Chatziparaschis, D.; Lagoudakis, M.G.; Partsinevelos, P. Aerial and Ground Robot Collaboration for Autonomous Mapping in Search and Rescue Missions. *Drones* **2020**, *4*, 79. [[CrossRef](#)]
10. Majeed, A.; Hwang, S. Path Planning Method for UAVs Based on Constrained Polygonal Space and an Extremely Sparse Waypoint Graph. *Appl. Sci.* **2021**, *11*, 5340. [[CrossRef](#)]
11. Wang, R.; Sun, Q.; Tu, P.; Xiao, J.; Gui, Y.; Wang, P. Reduced-Order Aggregate Model for Large-Scale Converters with Inhomogeneous Initial Conditions in DC Microgrids. *IEEE Trans. Energy Convers.* **2021**, *36*, 2473–2484. [[CrossRef](#)]
12. Wang, R.; Sun, Q.Y.; Zhang, P.J.; Gui, Y.H.; Qin, D.H.; Wang, P. Reduced-Order Transfer Function Model of the Droop-Controlled Inverter via Jordan Continued-Fraction Expansion. *IEEE Trans. Energy Convers.* **2020**, *35*, 1585–1595.
13. Chang, Z.; Zhang, Z.; Deng, Q.; Li, Z. Route planning of intelligent bridge cranes based on an improved artificial potential field method. *J. Intell. Fuzzy Syst.* **2021**, *41*, 4369–4376. [[CrossRef](#)]
14. Zhang, Z.; Wu, J.; Dai, J.; He, C. A Novel Real-Time Penetration Path Planning Algorithm for Stealth UAV in 3D Complex Dynamic Environment. *IEEE Access* **2020**, *8*, 122757–122771. [[CrossRef](#)]

15. Roy, S.K.; De, D. Genetic Algorithm based Internet of Precision Agricultural Things (IopaT) for Agriculture 4.0. *Internet Things* **2022**, *18*, 100201. [[CrossRef](#)]
16. Chen, J.; Ling, F.; Zhang, Y.; You, T.; Liu, Y.; Du, X. Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system. *Swarm Evol. Comput.* **2022**, *69*, 101005. [[CrossRef](#)]
17. Mesquita, R.; Gaspar, P.D. A Novel Path Planning Optimization Algorithm Based on Particle Swarm Optimization for UAVs for Bird Monitoring and Repelling. *Processes* **2021**, *10*, 62. [[CrossRef](#)]
18. Zhu, D.; Zhou, B.; Yang, S.X. A Novel Algorithm of Multi-AUVs Task Assignment and Path Planning Based on Biologically Inspired Neural Network Map. *IEEE Trans. Intell. Veh.* **2021**, *6*, 333–342. [[CrossRef](#)]
19. Tseng, F.H.; Liang, T.T.; Lee, C.H.; Der Chou, L.; Chao, H.C. A Star Search Algorithm for Civil UAV Path Planning with 3G Communication. In Proceedings of the 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Kitakyushu, Japan, 27–29 August 2014; pp. 942–945.
20. Lv, Z.; Yang, L.; He, Y.; Liu, Z.; Han, Z. 3D Environment Modeling with Height Dimension Reduction and Path Planning for UAV. In Proceedings of the 9th International Conference on Modelling, Identification and Control (ICMIC), Kunming, China, 10–12 July 2017; pp. 734–739.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.