

Article

# Non-Standard Map Robot Path Planning Approach Based on Ant Colony Algorithms

Feng Li <sup>1,2</sup> , Young-Chul Kim <sup>1,\*</sup> and Boyin Xu <sup>1</sup>

<sup>1</sup> Department of Mechanical Engineering, Kunsan National University, Gunsan 54150, Jeollabuk, Republic of Korea; fengli001@126.com (F.L.); xuboyin49@gmail.com (B.X.)

<sup>2</sup> College of Smart Manufacturing, Zhengzhou University of Economics and Business, Zhengzhou 450007, China

\* Correspondence: kimyc@kunsan.ac.kr

**Abstract:** Robot path planning is an important component of ensuring the robots complete work tasks effectively. Nowadays, most maps used for robot path planning obtain relevant coordinate information through sensor measurement, establish a map model based on coordinate information, and then carry out path planning for the robot, which is time-consuming and labor-intensive. To solve this problem, a method of robot path planning based on ant colony algorithms after the standardized design of non-standard map grids such as photos was studied. This method combines the robot grid map modeling with image processing, bringing in calibration objects. By converting non-standard actual environment maps into standard grid maps, this method was made suitable for robot motion path planning on non-standard maps of different types and sizes. After obtaining the planned path and pose, the robot motion path planning map under the non-standard map was obtained by combining the planned path and pose with the non-standard real environment map. The experimental results showed that this method has a high adaptability to robot non-standard map motion planning, can realize robot path planning under non-standard real environment maps, and can make the obtained robot motion path display more intuitive and convenient.

**Keywords:** robot path planning; non-standard map; calibration; grid map; ant colony algorithm



**Citation:** Li, F.; Kim, Y.-C.; Xu, B. Non-Standard Map Robot Path Planning Approach Based on Ant Colony Algorithms. *Sensors* **2023**, *23*, 7502. <https://doi.org/10.3390/s23177502>

Academic Editors: Nebojsa Bacanin, Robertas Damaševičius, Justyna Patalas-Maliszewska and Lalit Garg

Received: 11 July 2023

Revised: 17 August 2023

Accepted: 17 August 2023

Published: 29 August 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

With the development of the times, robots have become indispensable assistants in various industries, among which mobile robots are more widely used due to their unique structure. At bus stops, they can replace humans in automatic response. In warehouses, they can replace humans for material handling. On the battlefield, they can even replace humans in charge, assault, etc. [1–4]. The unique functions of robots make them increasingly at the forefront of technological development and an indispensable tool for humanity.

The determination of the robot's movement path is a crucial step in completing work tasks. The planning of robot movement path mainly refers to the route that the robot needs to pass through during the process of reaching another position from one location. This route not only ensures that the robot can reach the endpoint from the starting point but also ensures that the robot can avoid obstacles and dangerous objects on the path. With the development of technology, the intelligence level of robot movement path planning has increasingly become a focus of research for researchers. How robots can quickly and intelligently determine their movement paths can greatly optimize the autonomy and speed of mobile robot navigation. Therefore, the study of robot path planning method is of great significance [5].

In order to determine the robot's motion path, it is first necessary to obtain the robot's motion environment information, which is the robot's motion environment map. Currently, robot mobile maps mainly include Occupancy Grid Map, Octo Map, and Point cloud Map. Due to the discreteness of data in computer, we need to perform a discreteness design on

the x and y axes. In this way, we can obtain a large number of grids through discretization design on the x and y axes. Each grid can be represented by 0 or 1. This can generate a structured Occupancy Grid Map. Occupancy Grid Maps are typically generated through laser measurement of relevant data. It has a simple structure and is easy to implement various intelligent path planning algorithms for robots [6,7].

In addition, we can continuously split the 3D grid map into smaller cubes, label obstacles, and hazards. This forms an Octo map. An Octo map is a tree with eight subordinate nodes. It is often used for the expression of three-dimensional data [8,9]. We can divide the cube into eight small cubes and continue dividing the cubes containing obstacles or hazards until the requirements are met. When all the subordinate nodes of a node are occupied or idle or unknown, we can remove it and all its subordinate nodes. In addition, there is also a point cloud map that uses LiDAR to obtain the three-dimensional coordinates of different points and generate maps based on laser reflection intensity. It is precisely with the existence of these maps that the path planning for robots becomes possible.

After the robot's movement map is generated, various algorithms are also used to determine the robot's movement path and pose. In 1956, Edsger Wybe Dijkstra proposed the Dijkstra algorithm. This algorithm centers around the initial point and gradually expands outward until the goal is achieved. The A\* algorithm appeared in 1968 and optimized the Dijkstra algorithm through heuristic functions to find the optimal path faster [10]. In 1985, Khatib proposed an artificial potential field method, which assumes that obstacles have a repulsive force on the robot and the target object has gravitational force on the robot. The repulsive force and gravitational force are combined to form the motion path of the robot [11,12]. The random road map (PRM) algorithm is an algorithm that converts the continuous space state to the discrete space state and then uses search algorithms such as A\* to find the path [13,14]. The rapidly exploring random tree (RRT) algorithm detects the collision of the acquisition points to avoid spatial modeling [15–17]. It effectively solves path planning problems in high-dimensional spaces and complex constraints. In addition, there is the genetic algorithm (GA) proposed by John Holland, which simulates the search for optimal solutions in natural evolutionary processes [18,19]. It is a computational model based on natural selection and genetics [20,21].

By combining various path planning algorithms with environmental maps, we can achieve path planning for mobile robots. Since ant colony algorithms have highly efficient searchability [22–27], it has also become a commonly used path planning method. This paper proposed a method to transform the non-standard real environment map into the standard robot mobile environment map-occupancy grid map, and to then use ant colony optimization algorithms to realize the robot moving path, position, and pose planning.

In this paper, a new method of ant colony algorithm planning robot motion path using a real environment map was proposed. This method processes non-standard real environment maps and converts them into standard occupancy grid maps. Through ant colony algorithms, the planning path and pose of the robot are given. Then, the robot's planned path and pose are combined with the original non-standard map and generate a mobile path planning map for the robot in a non standard environment. The planning results of the virtual environment and the results of the real experimental environment have once again verified the application potential of this method.

The second part of the article discusses the origin of the research and introduces the practical research issues and related preparations. Part three introduces the overall solution approach, explains the method of converting detailed non-standard maps into standard maps, and the path planning was implemented in steps. The fourth part validated the method of path planning for robots under non-standard maps in real environments through experiments and conducted comparative analysis. The fifth part summarizes the article [28].

## 2. Problem Description

### 2.1. Realistic Issues

The planning of robot motion trajectory needs to be based on a map. Robot motion maps are mostly based on sensors such as infrared to explore surrounding objects, collect relevant information, and create maps based on the collected information, as shown in Table 1 [11–37]. The robot avoids different obstacles based on the formed map and gradually approaches the target through the planned motion path and pose requirements. However, information collection requires a lot of time, and the data processing and map generation after collection also require a lot of time. If the environmental information that needs to be collected is relatively complex, the time for establishing this map is quite long. Efficiently generating robot motion environment maps has become an urgent problem to be solved [38,39].

**Table 1.** Comparison of different path planning methods using maps now.

Algorithm	Main Usage Maps	Algorithm Type	Real Environmental Map Used
ACO	Grid map, Octo Map	Search algorithm	No
A*	Grid map	Search algorithm	No
PRM	Grid map	Search algorithm	No
PRT	Grid map	Probabilistic path planning algorithm	No
RRT	Grid map, Octo Map	Probabilistic path planning algorithm	No
GA	Grid map,	Search algorithm	No
PSO	Grid map, Octo Map	Search algorithm	No

The development of artificial intelligence has made rapid progress. If we can obtain a real environment map by taking photos, process the real environment map, transform non-standard environment maps into robot path planning maps suitable for robot path planning, and then carry out path planning and pose planning for robots, this can greatly improve the efficiency of robot path planning. Grid maps are commonly used nowadays, and if non-standard maps can be converted into more common grid maps, subsequent processing will be more convenient. The purpose of this article is to process the non-standard real environment map to form a standard grid map, and then realize the robot's motion path and pose planning through ant colony algorithms.

### 2.2. Processing of Non-Standard Maps

#### 2.2.1. Image Processing

The image-processing technology is relatively mature. It can be used for disease prediction [40], crop recognition [41], and industrial defective product selection [42]. We can use image-processing technology to segment and crop non-standard maps in real environments, locate, detect, and frame obstacles in the images, and identify obstacles. Subsequently, a new image is generated using the previously obtained image parameters, which is the standard robot motion path planning map that needs to be obtained. Because the image itself expresses the relative relationships between objects in the environment, using non-standard maps for image processing to obtain standard grid maps is more accurate and convenient.

#### 2.2.2. Obtaining Maps Adapted to Algorithms

The grid method is a commonly used map modeling method in robot path planning, which is simple and convenient to use. This article obtains relevant modeling parameters through non-standard real environment maps and uses the grid method to establish a map of the mobile robot's mobile environment. In the map, 0 represents the area of the free area, and 1 represents the area of obstacles or dangerous objects in the map. The grid with a 0 value is represented in white, while the grid with a 1 value is represented in black (see Figure 1). This is similar to the principle of image binarization in image processing, but due

to the fact that image processing is mostly based on pixels and the data are huge, we need to perform secondary processing after image processing. Alternatively, the non-standard map itself can be processed to obtain a standard map suitable for robot path planning.

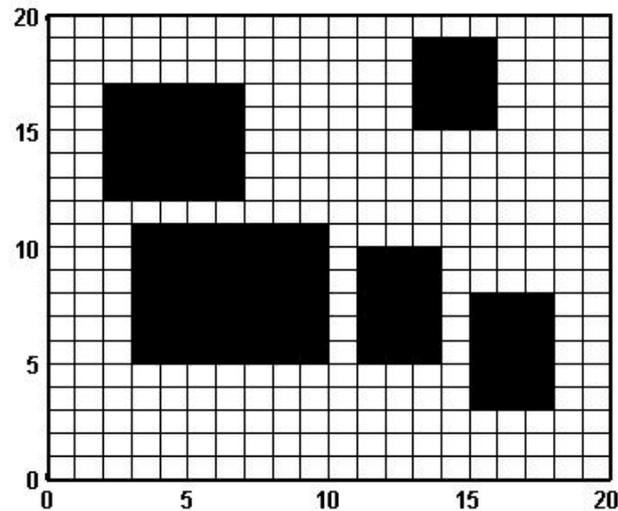


Figure 1. Common ant colony algorithm grid map.

### 2.2.3. Method of Image Processing

After the image is read, the image is transformed into an array  $I$ . If the image is a real color array, the array  $I$  is an array of  $m \times n \times 3$ . Because the grid graph is a two-dimensional array of small squares, it is necessary to convert the image into a two-dimensional array. Therefore, it is necessary to perform binary processing on the image. We can convert the true color image into a grayscale image, and convert  $m \times n \times 3$  arrays to  $m \times n$  arrays. In this way, the non-standard map can be transformed into a standard grid map that can implement ant colony algorithms. The binarized image also needs to be subjected to obstacle recognition, positioning, and subsequent data processing, as detailed in Section 3.

## 2.3. Introduction to Ant Algorithm

### 2.3.1. Basic Principles

Among intelligent algorithms, the ant colony algorithm is a common algorithm. This method mainly uses the method that each ant in the ant colony selects the moving route under the guidance of a pheromone to calculate. Because the accumulation degree of pheromone released by ants is different on different routes, ants will choose the route with more pheromone accumulation when moving. We can assume a directed graph  $G(N, A)$  with  $N$  locations, where  $N = \{1, 2, 3, 4, \dots, n-1, n\}$ ,  $A = \{(i, j) \mid i, j \in N\}$ .

The scheduled path planning task is represented by the objective function:

$$\min f(\omega) = \sum_{i=1}^n l_{i-1,i}, \quad (1)$$

and  $\omega = (i_1, i_2, i_3, i_4 \dots i_{n-1}, i_n)$ .

$l_{ij}$  is the distance between two points,

$$l_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad (i, j \in N) \quad (2)$$

When ants move from one position to the next, they can choose the next position based on the probability of reaching the next position.

This probability needs to be calculated using the following equation:

$$P_{ij}^k(t) = \frac{[\tau_{ij}^{(t)}]^\alpha \times [\eta_{ij}^{(t)}]^\beta}{\sum_{k \in \text{allowed } k} [\tau_{ij}^{(t)}]^\alpha \times [\eta_{ij}^{(t)}]^\beta}, \quad (3)$$

Among them,  $k \in \text{allowed } k$ ;  $i, j$  is the starting point and target point of the ants,

$$\eta_{ij} = \frac{1}{l_{ij}}, \quad (4)$$

$\eta_{ij}$  is visibility, which is the reciprocal of the distance between  $i$  and  $j$ ;  $\tau_{ij}^{(t)}$  is the pheromone strength from  $i$  to  $j$  at time  $t$ ; allowed  $k$  is a collection of locations that have not been visited yet;  $\alpha, \beta$  are two constants. They are the weighted values of pheromone and visibility.

In the process of ant colony algorithms, the pheromone in the algorithm is constantly updated.

$$\tau_{ij}^{(t)} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k, \quad (5)$$

In the above formula,  $m$  is the number of ants in the ant colony.  $\Delta\tau_{ij}^k$  is the Pheromone left by the  $k$ -th ant when passing on the path from  $i$  to  $j$ .

$$\Delta\tau_{ij}^k = (C_k)^{-1}, \quad (6)$$

Among them,  $i, j$  are the points on the path passed by the ant.  $C_k$  is the total length of the path obtained by the  $k$ -th ant walking along the entire path.

In this way, the ant completes the path once, iterates the operation once, and implements the algorithm through continuous iteration [43].

### 2.3.2. Implementation

Usually, after generating a standard grid map, we can set the starting point and the target point to be reached in the planned path according to the requirements and set the basic parameters of the algorithm. The ant colony algorithm is used to search the path. The ant updates the pheromone on the path according to the algorithm after reaching the destination, outputs information, and saves the path information.

### 2.3.3. Post-Processing

After the robot path planning is realized according to ant colony algorithms, it is also necessary to obtain the position and pose information of the robot movement according to the path information. After obtaining the path and pose information, it is necessary to combine the path and pose information with the non-standard real environment map to complete path planning and pose planning under non-standard maps. The entire process is a unified one. In this way, combined with image processing, ant colony algorithm, and pose planning algorithm, it is very convenient to realize robot path planning under non-standard maps [2,44,45].

### 2.3.4. Related Work

There are many methods for robot path planning now. Jinghua Wang et al. combined Dijkstra's algorithm and used a fuzzy logic system to consider the environmental conditions of path planning to study the path planning of mobile robots in complex two-dimensional terrain [46]. Oscar Loyola, John Kern, and Claudio Urrea proposed to generate the robot's path by adding a boredom element without a reward [47]. Yigao Ning et al. proposed to add a safe distance to the existing repulsive potential energy field, and an obstacle avoidance controller was designed based on IAPF to study vehicle path planning under a multi-

obstacle environment [48]. Novak Zagradjanin et al. proposed a cloud-based autonomous multi-robot system to perform highly repetitive tasks. The study further improved path planning in complex and crowded environments [49]. W Wei et al. studied the long-distance autonomous path planning of intelligent vehicles by using the Dijkstra algorithm and studied the autonomous obstacle avoidance of intelligent vehicles by using the artificial potential field method combined with the direction sensor and infrared sensor [50]. Hao Ge, Xin Li, Leiyan Yu et al. also studied the path planning method through different methods [51–55]. These methods implement and optimize the path planning from different aspects, but the establishment of the environment map is based on free modeling, and the path planning is not implemented in the real environment. Haichuan Zhang et al. proposed a search path design algorithm to improve the efficiency of maritime search and rescue. Based on the basic ant colony algorithm, they designed the search and rescue path of the ship at sea [56]. In this study, the designed maritime search and rescue path was finally represented on the navigation map. However, this study was still based on the existing parameters for modeling and generating maps for path planning and was not carried out in the real environment map.

It can be seen that the research of planning path algorithms on real environment maps is blank. However, applying the planning path algorithm to the real environment map can not only greatly improve the intelligence and practicability of the algorithm, but also greatly reduce the control difficulty of robots and vehicles. At present, car automatic driving is generally based on the combination of satellite navigation and sensors, if the path can be automatically planned under the real environment map, it will greatly improve the intelligence of automatic driving [57]. The path planning method under the real environment non-standard map of this article will be introduced in detail below.

This article first created a non-standard simulation map of mobile robot motion and used the non-standard simulation map to provide an overall introduction to the method proposed in the article. We also conducted experiments using the path planning method proposed in this article by displaying non-standard maps of the environment. The proposed method was experimentally validated. The path planning method for mobile robots under non-standard maps proposed in the article mainly included three steps.

Step 1: Process non-standard real environment maps to form a standard grid map for robot motion path planning;

Step 2: Plan the robot's motion planning path according to the grid map;

Step 3: Integrate the path and pose planned by the grid map with non-standard real environment maps. Create a robot motion path and pose planning map under non-standard real environment maps.

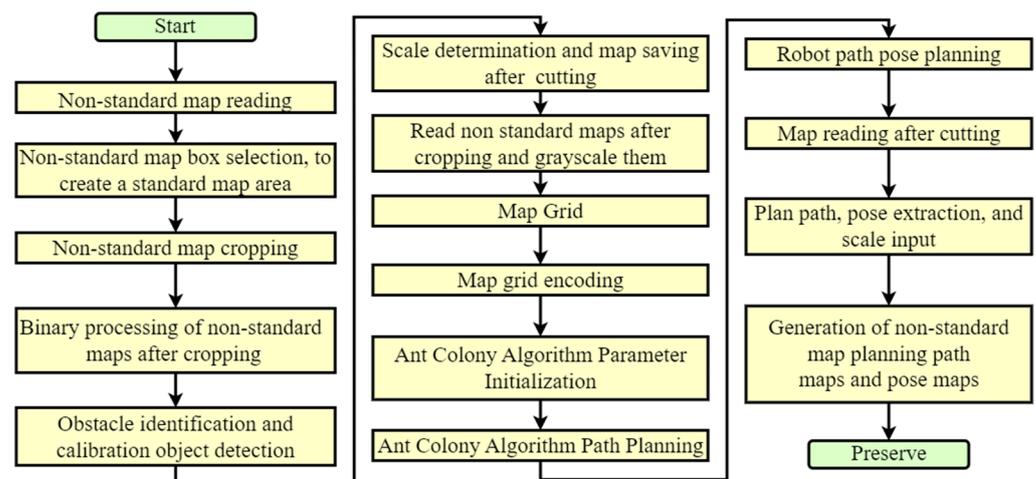
### 3. Methods

Performing ACO on a real environment map required first normalizing the non-standard real environment map to generate a standard 2D grid map. Then, a 2D grid map and ant colony algorithm were used to plan the path from the initial location to the target location. At the same time, there were some obstacles in the non-standard map, and the planned path should not collide with these obstacles.

We read non-standard real environment maps using Matlab software. To standardize map processing, first, we selected the specified box selection area in the non-standard map, and we cut non-standard maps to obtain standard areas for establishing standard maps. Then, we performed binary processing on the cropped generated area. After the binarization of non-standard maps, obstacle recognition was performed on the map. In order to clearly reflect the actual dimensions on the map, there must be a scale calibration object inside the non-standard map. The dimensions of other objects and the planned path parameters should be referenced by the calibration object. This lay a good foundation for generating maps in non-standard environments that can reflect the real environment. Because the data after obstacle recognition were extracted by MATLAB, the binarized map was then subjected to two-dimensional grid processing. In order to facilitate the reading

of positions on the map, it was necessary to encode the grid after grid processing. This made it very convenient to know the positions of various points on the map. After grid processing, non-standard maps were converted into standard two-dimensional grid maps.

Next, path planning was carried out through the ant colony algorithm. After obtaining the planned path, we can proceed with pose planning based on the path parameters. After obtaining the path and pose, we read the non-standard map. Due to the use of size calibration objects in the early stage, the obtained path parameters and pose parameters can be converted into real environmental parameters. Based on these parameters, we can display the planned path and pose in the non-standard map, so as to obtain the planned path and pose of the non-standard map. The main process is shown in Figure 2.



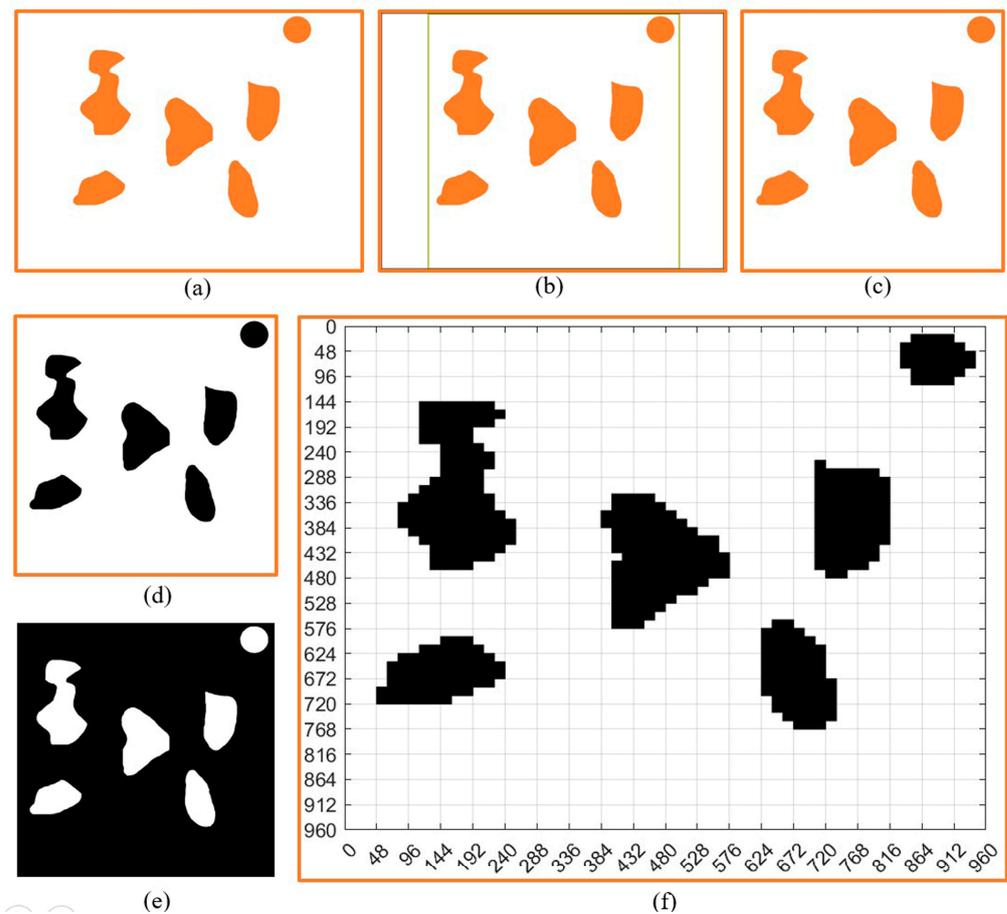
**Figure 2.** Non-standard map path planning approach based on ACO research method process.

### 3.1. Non-Standard Map Processing

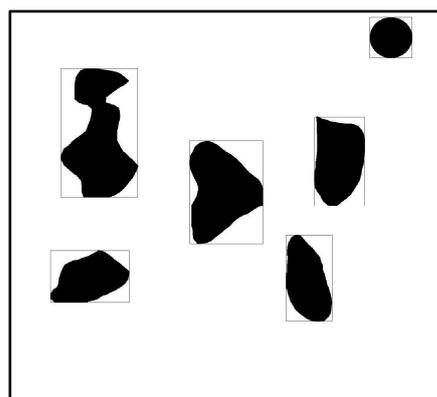
For ease of explanation, we first established a virtual non-standard map, as shown in Figure 3. From the figure, it can be seen that each obstacle on the map had different sizes and irregular shapes. The circle in the upper right corner of Figure 3a represents the designated calibration object, and we assumed its diameter to be  $L$ . The size of other objects was based on it as a reference. In order to streamline, generalize, and standardize the entire process, we box-selected the non-standard map in Figure 3a with a size of  $a \times a$ . Then, we obtained Figure 3b. After box selection, we cropped it to obtain Figure 3c. Next, we binarized Figure 3c to obtain Figure 3d,e, which are two different forms of image binarization. In order to facilitate subsequent processing and facilitate the identification of obstacles and calibration objects, we selected Figure 3d for subsequent processing.

### 3.2. Determination of Obstacles in Non-Standard Environmental Maps

After obtaining Figure 3d, we used the region props function provided by MATLAB to obtain the area parameters of the obstacle. We used the rectangle function for box selection, as shown in Figure 4. After the box selection, we not only obtained information about the obstacles but also obtained the size  $L_1$  of the calibration object in the figure. Though the size of the entire map was  $a \times a$ , because the actual diameter of the calibration object was  $L$ , the actual size reflected in the entire diagram should be  $(a \times L/L_1) \times (a \times L/L_1)$ . From the figure, it can be seen that the boundary between the identified obstacle and the calibration object in the upper right corner was clearly visible. We placed the calibration object in the upper right corner to accurately size the map without affecting the robot's final planning path.



**Figure 3.** A diagram of the process of generating a grid map with a non-standard map. (a) Non-standard environment map. (b) Non-standard environment map box selection. (c) Non-standard environment map cutting. (d) Non-standard environmental map binarization result 1. (e) Non-standard environmental map binarization result 2. (f) Grid design for non-standard maps.



**Figure 4.** Obstacle identification and box selection.

### 3.3. Grid Map Design

We needed to segment the binary map into  $n \times n$  grids to convert it into a grid map. This means that the map was divided into  $n^2$  grids. Because the size of the entire diagram was  $a \times a$ , the side length of each grid was  $u = a/n$ . Next, we used the floor function to convert the matrix where the obstacle was located to 0, and the remaining areas to 1. Afterward, the fill function was used to fill the area with a median value of 0 in black, forming a standard grid map as shown in Figure 3f. In order to facilitate the subsequent

ant colony algorithm path planning, we also needed to code each grid. Because the edge length of each grid was  $u$ , we divided the map into  $n \times n$  parts, so the position of each encoding can be obtained in the following way. The encoding principle process is shown in Algorithm 1, and the encoded map is shown in Figure 5.

Algorithm 1 Principle of location coding for non-standard environmental maps

Coding Principle

Input: Image edge length  $a$ , number of grids  $n \times n$

Output: Grid edge length  $u = a/n$

1. Loop  $n$  times starting from the first line
2. Loop  $n$  times starting from the first column
3. The encoding position  $y$  for each loop row is  $u \times j$ , and  $j$  is the current number of columns
4. Each loop has a column encoding position  $x$  of  $u \times i$ , where  $i$  represents the current number of rows
5. The encoded value for each position is  $\text{num2str}((i - 1) \times a/u + j)$
6. End column loop
7. End row loop
8. End

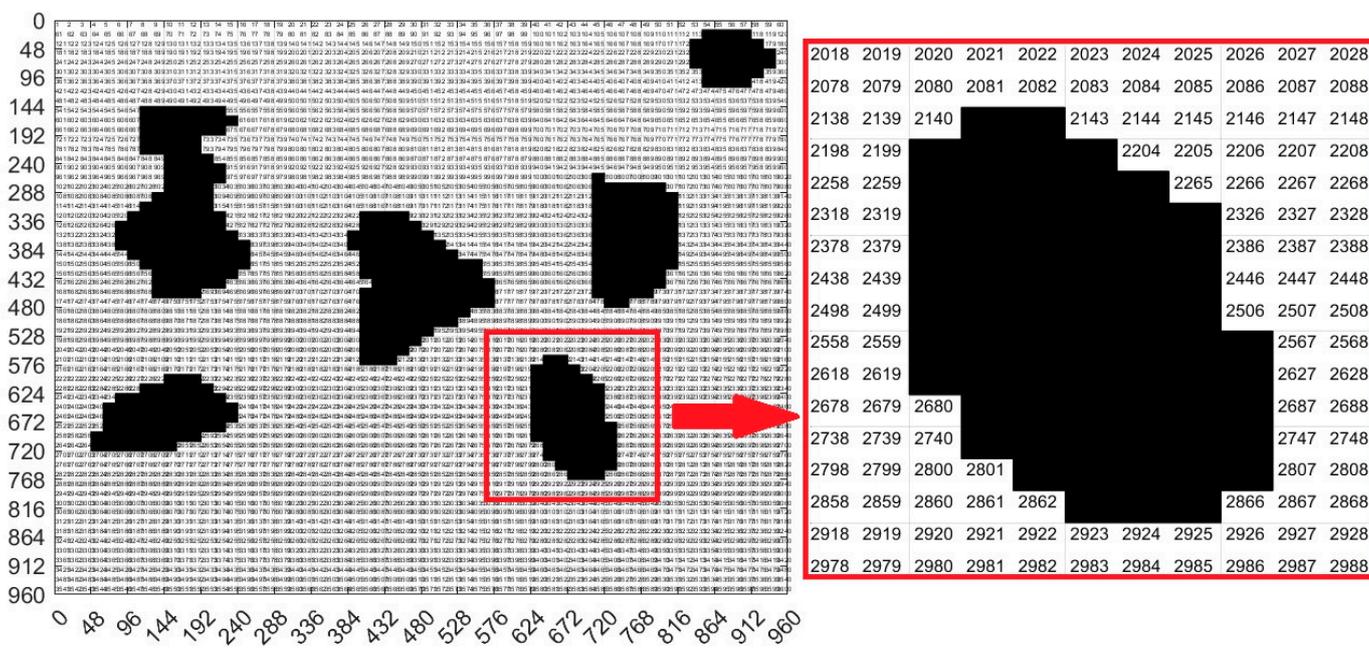
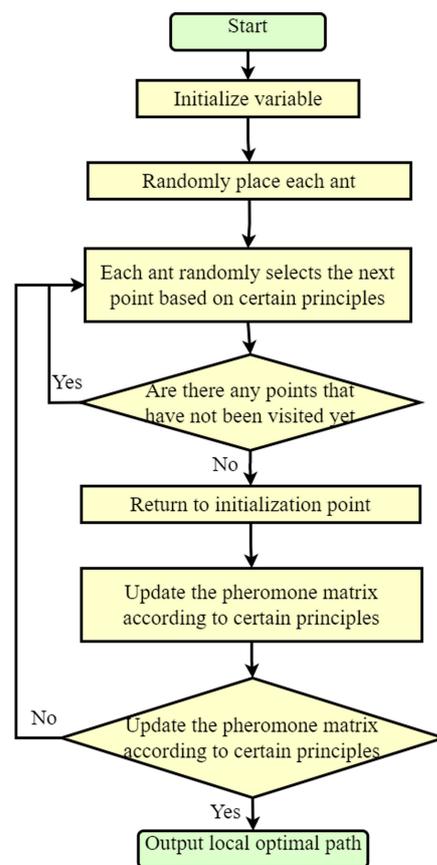


Figure 5. Display of standard map coding results.

3.4. Algorithm Design

After obtaining the standard map, we used ant colony algorithms to plan the path. The specific process is shown in Figure 6. Due to the encoding of the map in the early stage, path planning for any two locations can be implemented by inputting the encoding. From Figure 5, it can be seen that we divided the image into  $60 \times 60$  grids, which means there were a total of 3600 grids and 3600 positions in the entire image. Except for the areas marked as obstacles in the figure, other areas can be selected for path planning, which was convenient and simple.



**Figure 6.** Principle of ant colony algorithms.

In this virtual non-standard map, we set the starting position as point 130 and the target position as point 3513. The number of iterations here was 100. The number of ants was 100. These parameters can be increased or reduced, and the larger the value, the better the path planning effect. But the calculation time also increased. According to the previous research experience, we set the evaporation coefficient of the pheromone as 0.3, the increasing intensity coefficient of pheromone as 1, and the importance coefficient of the pheromone as 1 [28]. See the Table 2 for the other main parameters.

**Table 2.** Main parameters of virtual map for ant colony algorithm operation.

Parameter	$K$	$M$	$\alpha$	$\beta$	$\rho$	$Q$	$S$	$E$
Value	100	200	1.8	25	0.3	1	130	3513

### 3.5. Path and Pose Generation

Figure 7 shows the planning path obtained through 100 iterations of the ant colony optimization algorithms. In the figure, grid 130 is the initial point of the design, and grid 3513 is the end point of the design. The blue route is the planned path. Ants pass through 57 points, as shown in Table 3, through points 130, 191, 252, 313, 374–3393, 3453, 3513, bypassing obstacles and reaching the target location.



Based on the above analysis, the planned displacement and position can be easily obtained. In order to find a good planning path, we chose the convergence method with a limited number of iterations. Figure 8 is the displacement convergence curve obtained by 100 iteration based on the above analysis. Figure 9 is the pose planning diagram obtained from the robot planning route in Figure 7.

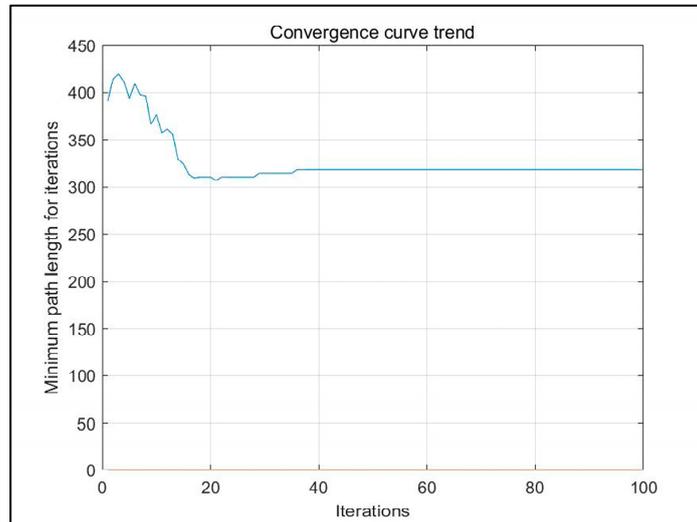


Figure 8. Displacement convergence curve chart of virtual map.

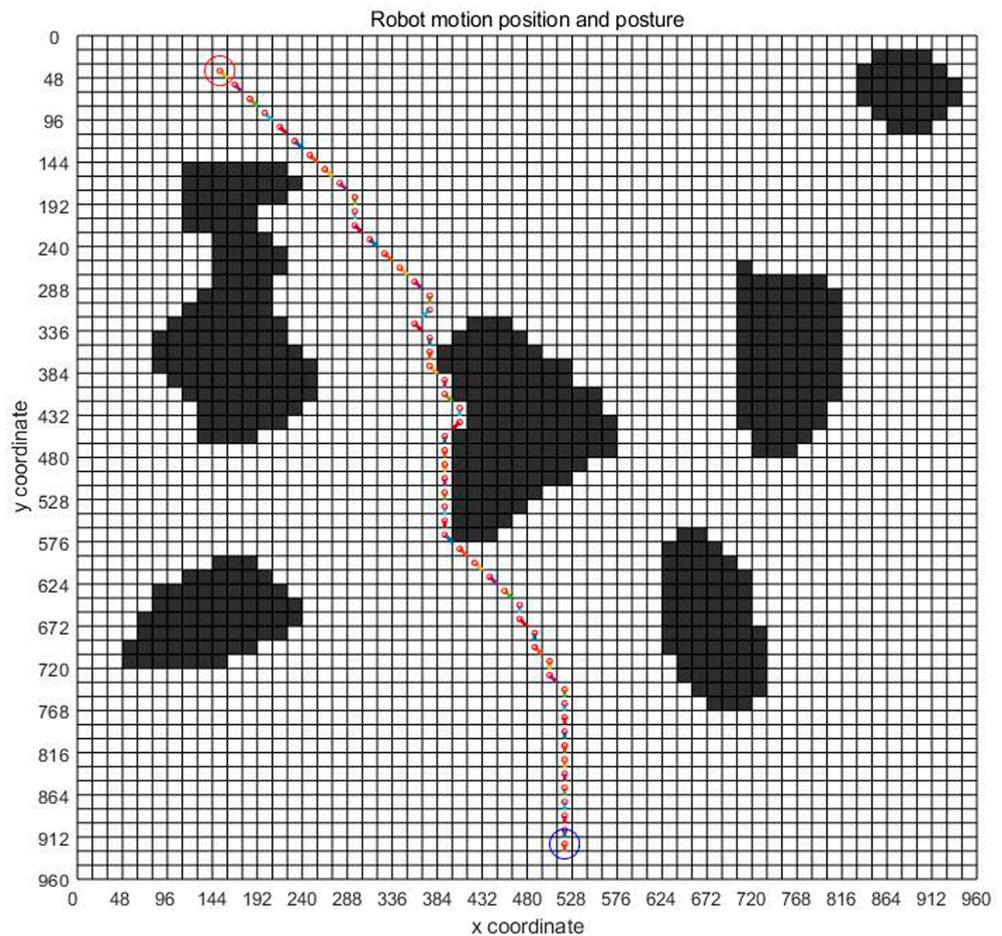


Figure 9. Pose planning of virtual map.

### 3.6. Integration of Non-Standard Virtual Environment Maps and Path Planning

Because the virtual map path in Figure 7 was obtained from the original non-standard map, the obtained planning path and pose can be regressed to the original non-standard map. According to the data in Figures 7 and 9, the path and position of ant optimization algorithms in the non-standard map can be obtained by displaying them in the original non-standard map. The thick blue line in Figure 10 is the robot motion path planned by the ant colony optimization algorithms. From the initial position to the target position, the robot's motion path was very clear. Each solid circle in Figure 11 represents the position of the robot at each step on the map, while the straight lines in the circle indicate its upcoming posture. Compared with traditional standard maps, this non-standard map that combines real environments was simpler and clearer for robot motion path planning and pose planning. From the process of robot motion path planning under the above virtual non-standard map, it can be seen that, theoretically, we can obtain the robot's planned path and pose by converting the non-standard real environment map into a standard grid map. Then, by returning the planned path and pose to a non-standard map, the robot's motion path planning route and pose in the real environment map can be obtained. Next, we needed to analyze and study the entire path planning process through real non-standard environmental maps.

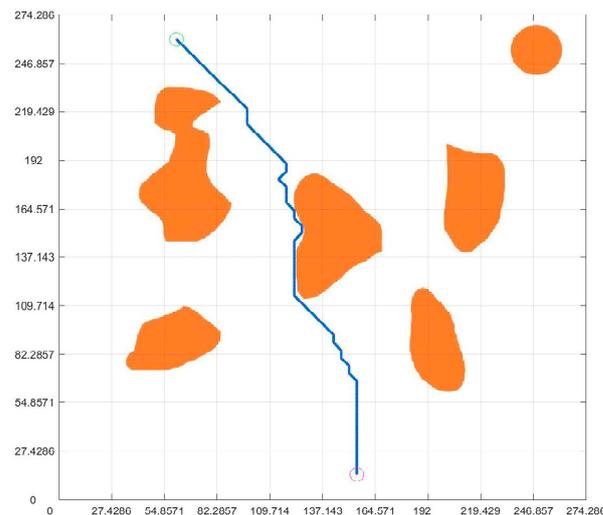


Figure 10. Virtual map and path fusion.

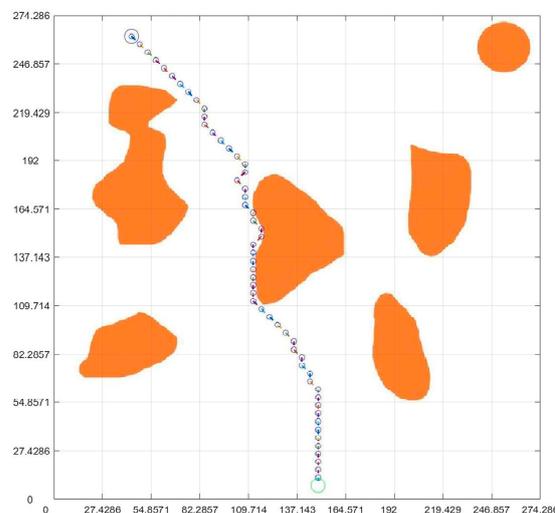
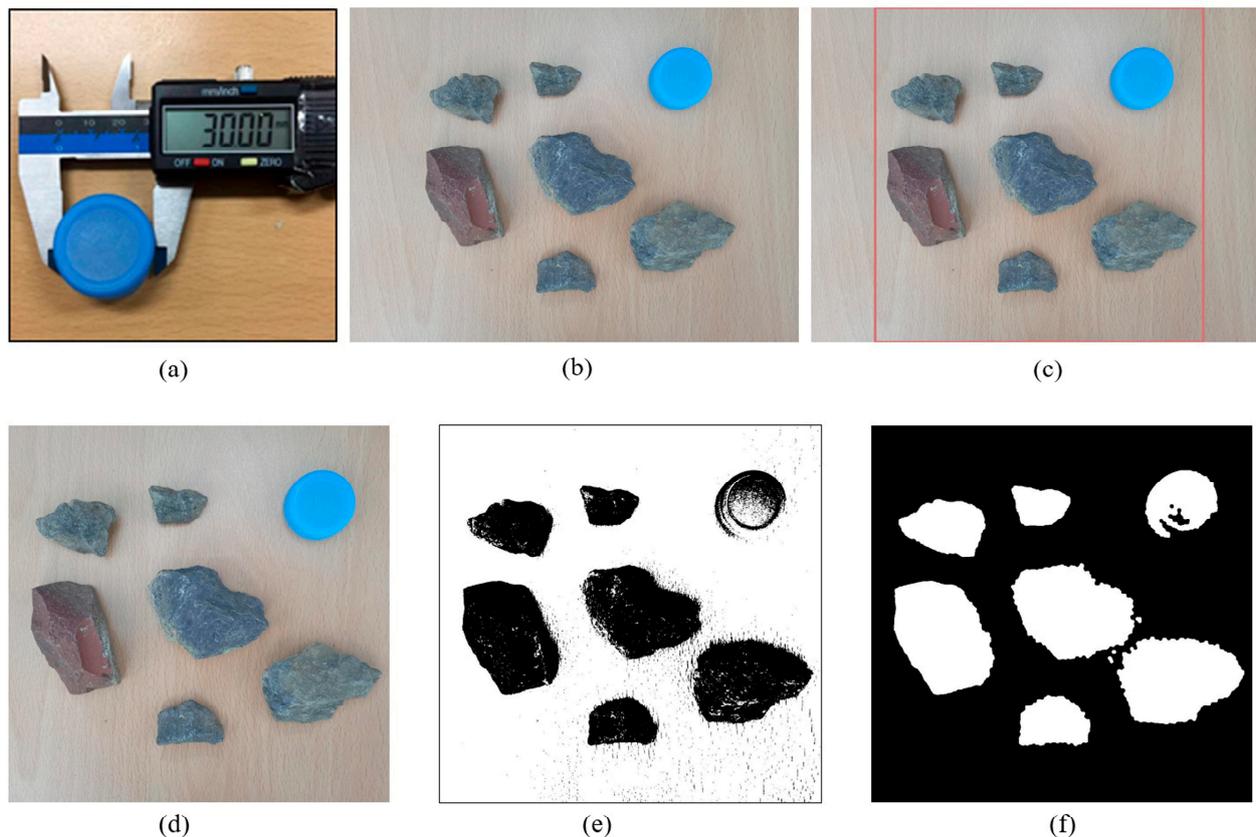


Figure 11. Virtual map and pose fusion.

## 4. Experiment

### 4.1. Processing of Non-Standard Real Environment Map

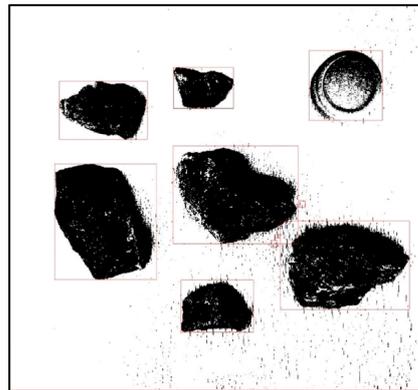
Figure 12 show the calibration and non-standard real environment map preprocessing. The calibration object in the upper right corner of Figure 12a is very important as it ensures reference for later map size determination. Similarly, we conducted it in the Matlab environment. Figure 12b shows a non-standard map of the real environment that was read in. After reading Figure 12b, we boxed and cropped it (as shown in Figure 12c) to obtain the area for later standardized map design, as shown in Figure 12d. By binarizing Figure 12d, we can obtain Figure 12e,f. For the convenience of later processing, we chose Figure 12e as the object for grid processing.



**Figure 12.** Calibration and non-standard real environment map preprocessing. (a) Size calibration object. (b) Non-standard real environment map. (c) Non-standard real environment map box selection. (d) Non-standard real environment map cutting. (e) Non-standard environmental map binarization result 1. (f) Non-standard environmental map binarization result 2.

### 4.2. Identification of Obstacles and Calibration Objects in Non-Standard Environmental Maps

After obtaining Figure 12e, we used the region props function to obtain the area parameters of obstacles and calibration objects, and we used the rectangle function for box selection, as shown in Figure 13. It is evident from Figure 13 that various obstacles were boxed, including the size calibration objects placed. We obtained a diameter of 183 for the calibration object in the system through box selection. As the box selection area parameter in the previous entire map was  $960 \times 960$ , the size of the entire image was  $960 \times 960$ . Because the actual size of the calibration object was 30 mm, based on the previous analysis, the scale obtained was  $183/30 = 6.1$ . From this, we obtained the true environmental map size through data conversion, which was  $960/6.1 = 157.377$  mm.



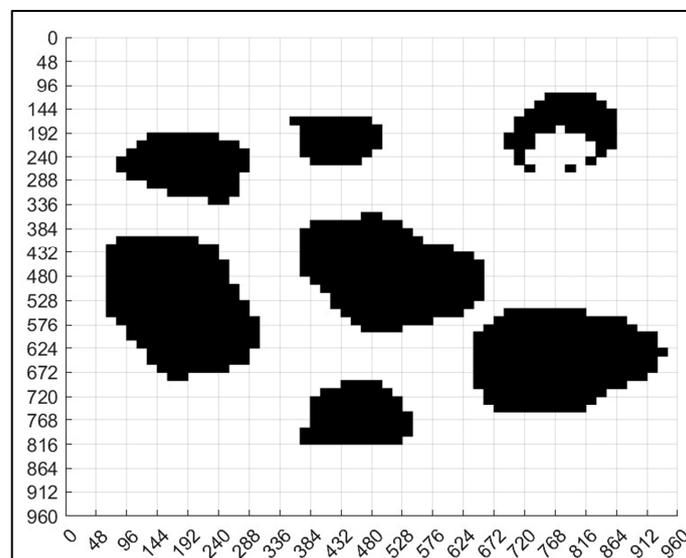
**Figure 13.** Identified obstacles in the real environment.

#### 4.3. Non-Standard Environment Map Generation

After binarization, we needed to segment the map to convert it into a grid map. We divided the map into  $60 \times 60$  grids, which means a total of 3600 grids, and the size of the entire map was  $960 \times 960$ ; therefore, the side length of each grid is

$$U = a/n = 960/60 = 16 \quad (10)$$

Next, we used the floor function to convert the matrix where the obstacle is located to 0, and the remaining areas to 1. Then, the fill function is used to fill the area with a median value of 0 in black, forming a standard grid map as shown in Figure 14. In order to facilitate the subsequent ant colony algorithms path planning, we also needed to code each grid. We divided the map into  $60 \times 60$  parts, and the encoded map is shown in Figure 15.



**Figure 14.** Non-standard real environment map grid design.

Figure 15 real environment map grid code result is also very important. Through it, we can clearly understand the specific positioning of each part in the map.

#### 4.4. Implementation of Ant Colony Optimization Algorithms

##### 4.4.1. Main Parameters

In this map, we set the starting position as point 80, the target position as point 3220, the number of iterations as 100, and the number of ants as 200. Based on the previous settings

in Table 2, we obtained the parameters of the non-standard map for the actual environment, as shown in Table 4. After the parametric setting, we can carry out ant algorithms.

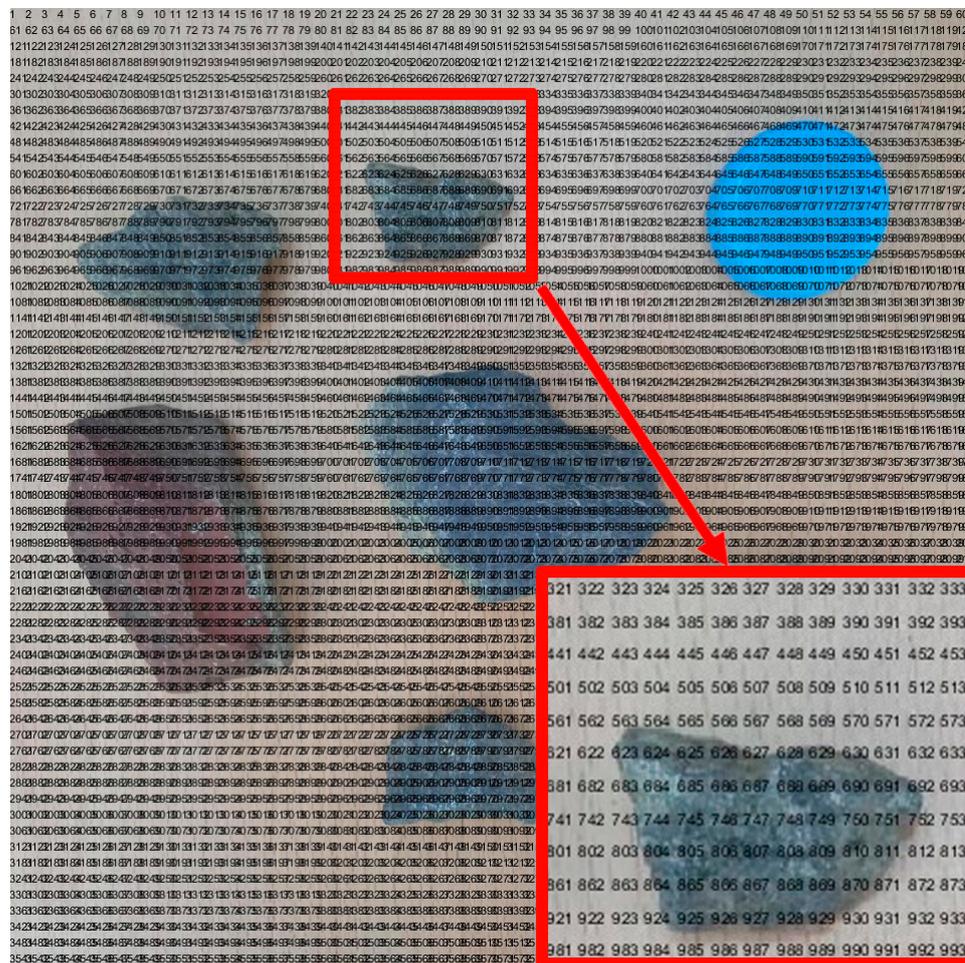


Figure 15. Real environment map grid code result.

Table 4. Parameters of ant colony algorithms.

Parameter	<i>K</i>	<i>M</i>	<i>Alpha</i>	<i>Beta</i>	<i>Rho</i>	<i>Q</i>	<i>S</i>	<i>E</i>
Value	100	200	2	30	0.3	1	80	3220

#### 4.4.2. The Results and Analysis

Figures 16–18 show the path planning map, pose planning map, and displacement iteration graph after the grid of non-standard map. They are the planned paths obtained through 100 iterations of ant colony algorithms. The path and poses planned by ant colony algorithm are clearly shown in the figure. At the same time, after 100 iterations, it can be seen from Figure 18 that the path planned by ant colony algorithms was generally regional optimization, and the final selected path was 274.233 mm. It should be noted that the path length here was obtained by converting it through a scale. The above results prove that the role of ant colony algorithm in path planning is very obvious. The planned path was optimized.

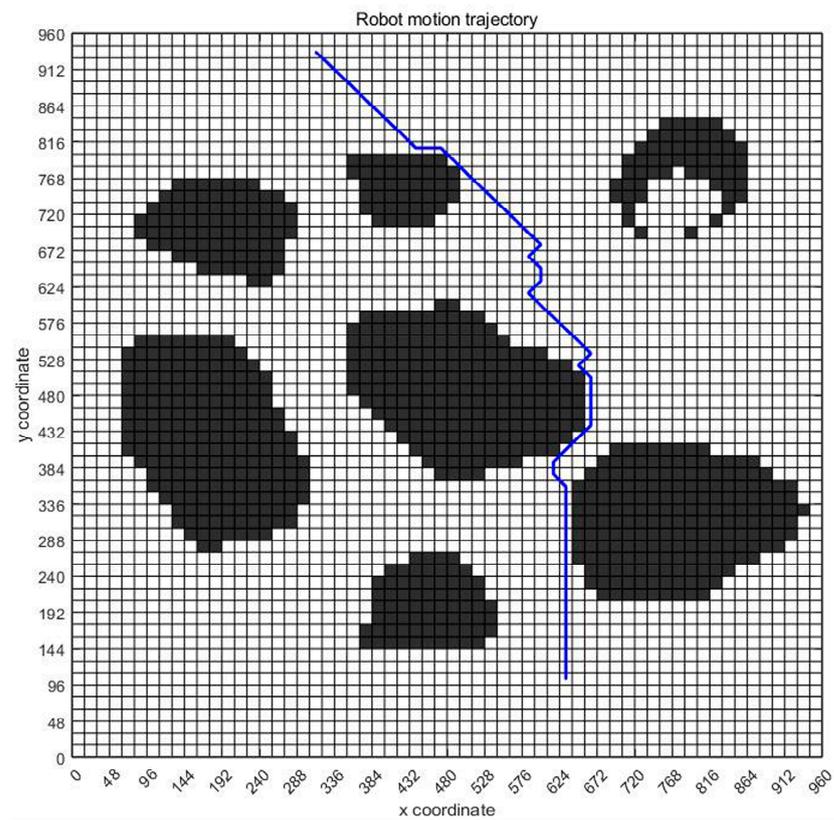


Figure 16. Standard map path planning result.

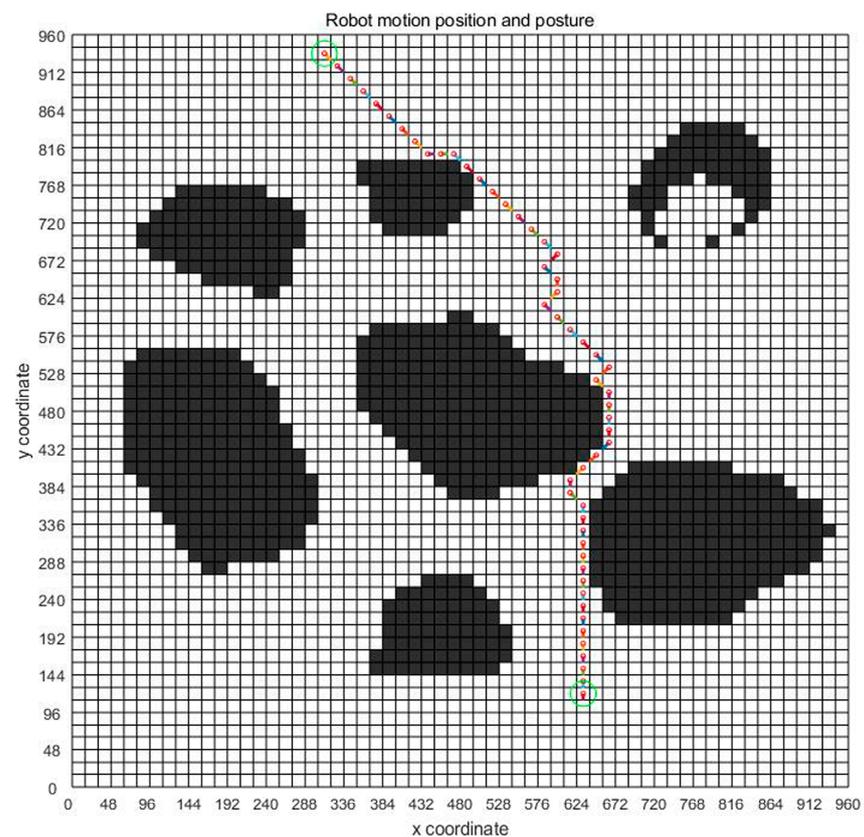


Figure 17. Standard map pose planning map.

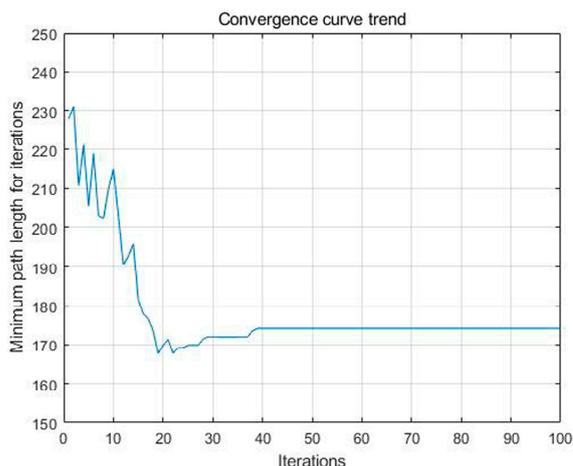


Figure 18. Displacement iteration graph of real environment.

4.5. Integration of Planning Path with Non-Standard Real Environment Map

After planning the path and pose, we opened the originally selected non-standard map of the real environment. Then, we applied the parameters obtained through previous calculations to the graph. Figure 19 is the planning path, Figure 20 is the planning pose. From Figure 19, it can be seen that the planned path was clear and reasonable, avoiding obstacles and ultimately reaching the endpoint. This also achieved the goal of this article. Compared to traditional standard maps, Figure 20 has a more guiding role in robot motion control.

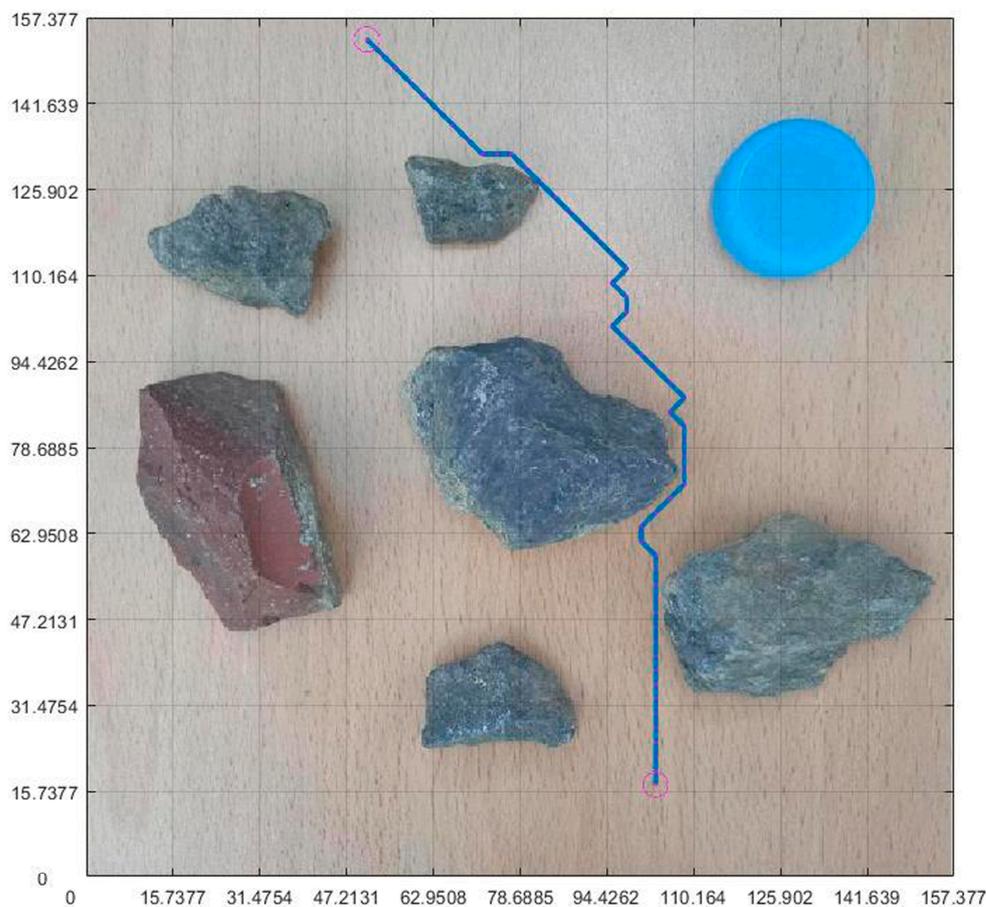
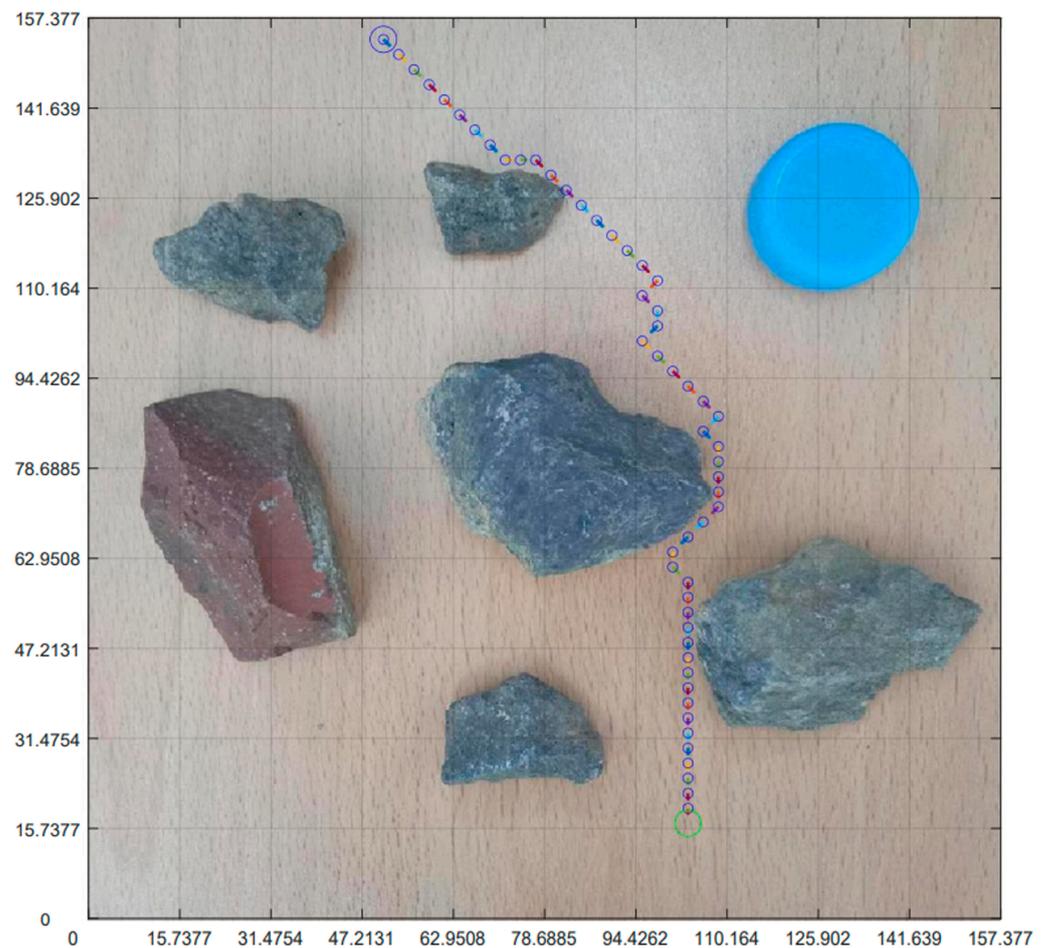


Figure 19. Integration of planning path and non-standard real environment map.

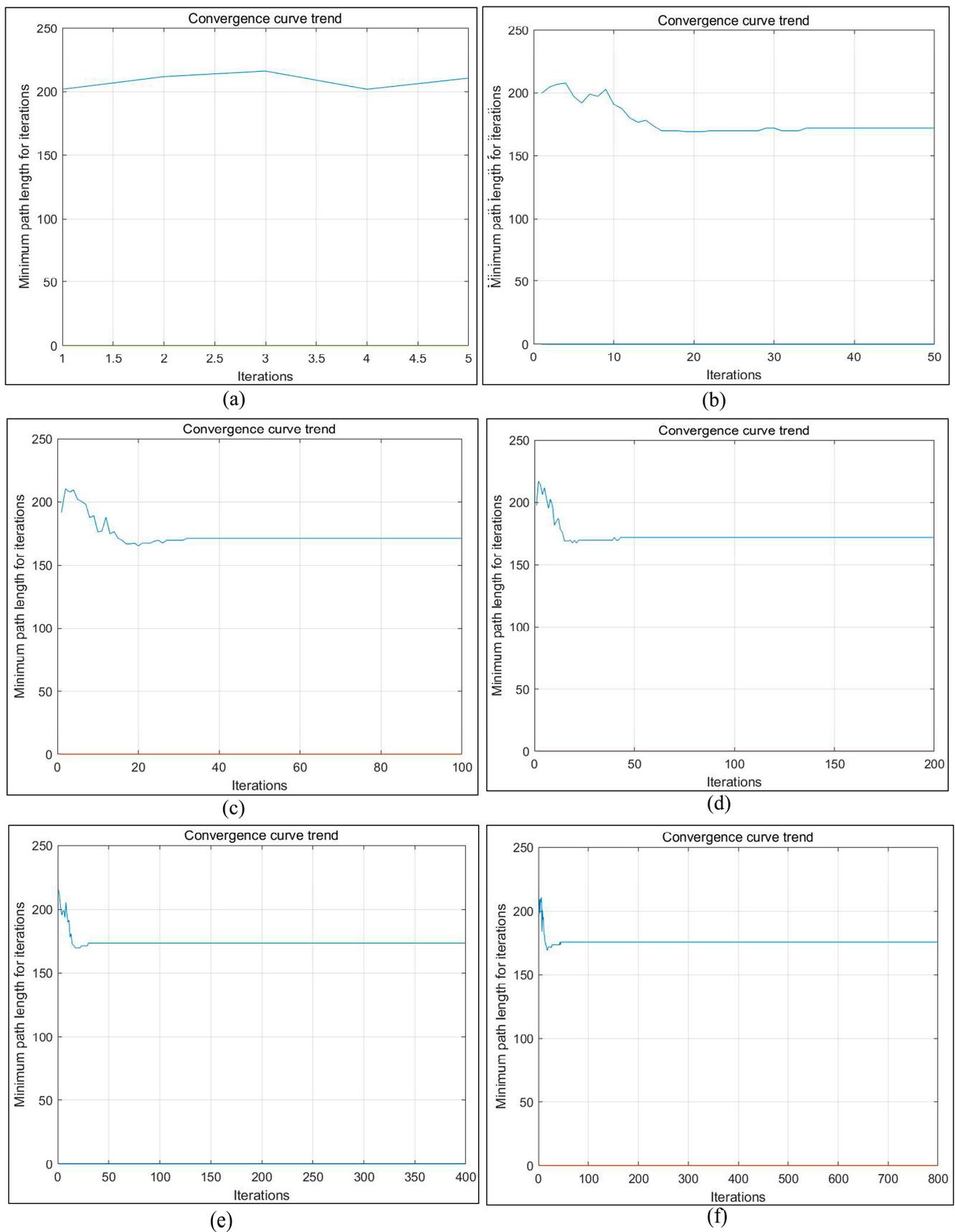


**Figure 20.** Integration of planning pose and non-standard real environment map.

#### 4.6. The Impact of the Number of Iterations

Theoretically speaking, increasing the number of iterations can better find the optimal motion path of the robot, but the greater the number of iterations, the longer the operation time will be. On the one hand, in order to improve the effect of path planning of ant colony algorithm on actual environment map, we conducted a comparison of six sets of experiments with different iterations, as shown in Figure 21.

According to Figure 21, the path length and processing time changes of Figure 22 can be obtained. It can be seen from the figure that under the path planning of the existing ant colony algorithm in a real environment, when the number of iterations was 50 and 800, the moving path displacement was the minimum value of 169.3 mm, and the movement path displacement changed little between 50 and 800 times. However, the processing time of the algorithm increased significantly, and the operation time of 800 times was 13.98 times that of 50 times. Obviously, when other existing parameters remained unchanged, setting the iteration number 50 times can not only improve the operation efficiency but also obtain better-planned paths and poses. This also laid a good foundation for us to conduct other related experiments in the later stage. When the number of iterations was 50 times, the path planning diagram in the real environment was as shown in Figure 23.



**Figure 21.** Convergence curves of different iterations under non-standard maps of real environment. (a) 5 iterations. (b) 50 iterations. (c) 100 iterations. (d) 200 iterations. (e) 400 iterations; (f) 800 iterations.

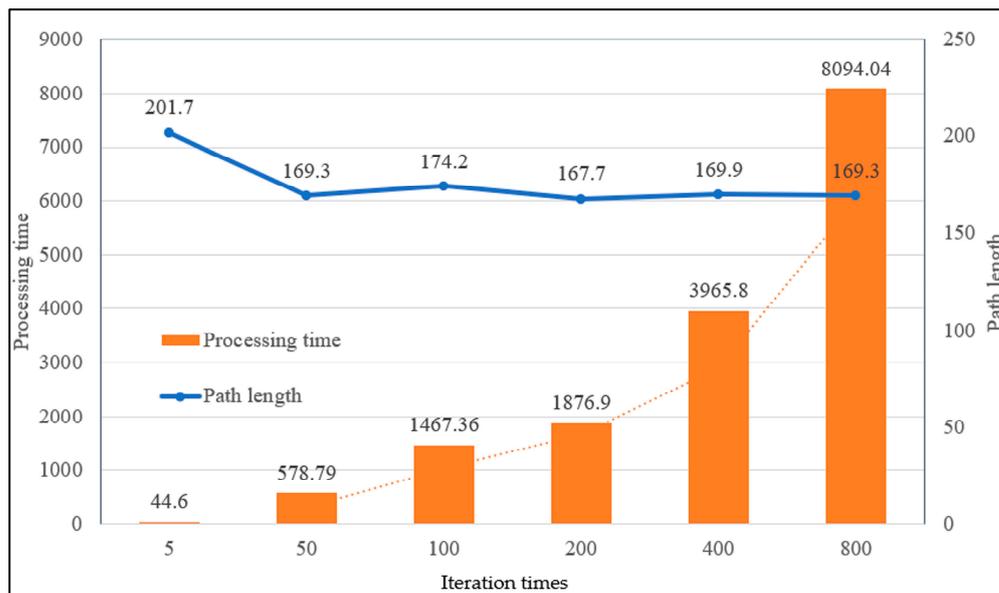


Figure 22. The changes in processing time and path length.

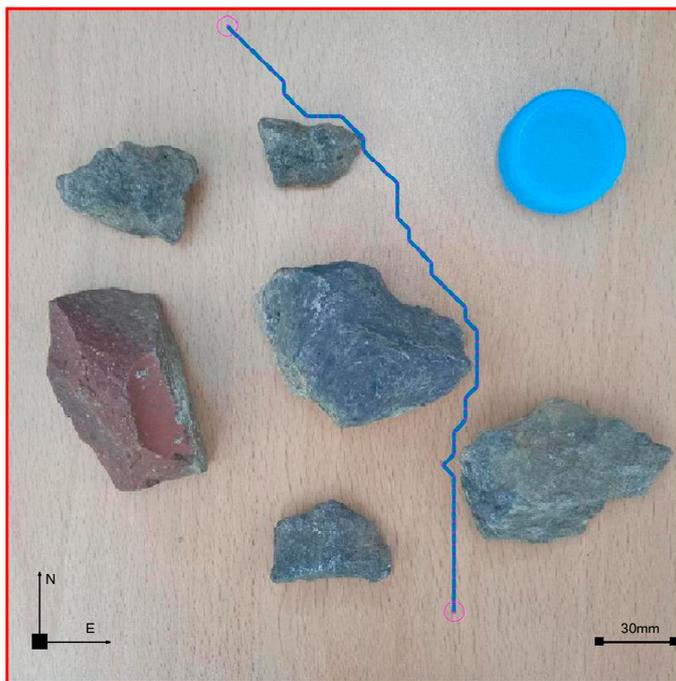
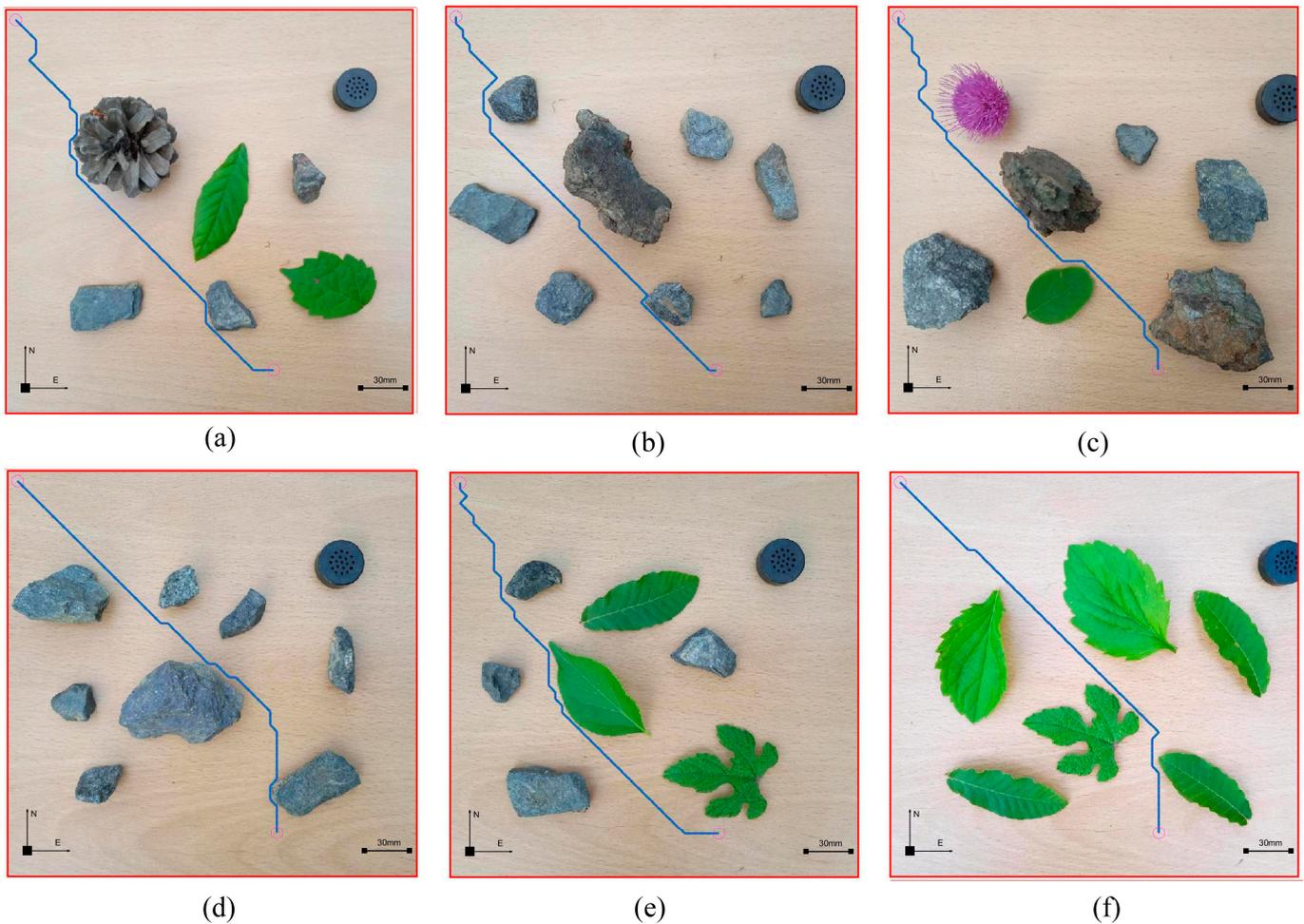


Figure 23. Integration of planning pose and non-standard real environment map with 50 iterations.

4.7. Repeat Implementation

In order to ensure the universality of the method described in this article, we verified it through multiple experiments. We conducted the same experiment on six non-standard real environment maps in Figure 24 and obtained satisfactory planning paths. After multiple experimental verifications, the path planning method proposed in this article was true and effective in non-standard maps in real environments. The path planning method proposed in this article had broad application prospects.



**Figure 24.** Planning paths under non-standard maps in different real environments. (a) Environment 1. (b) Environment 2. (c) Environment 3. (d) Environment 4. (e) Environment 5. (f) Environment 6.

## 5. Conclusions

With the development of technology, robots have become a common intelligent human-assistance tool. Because path planning is the core technology in robot motion control, its importance is obvious. In response to the problem that the maps used for robot path planning are often obtained through sensor measurement, which is time-consuming and laborious. This article studied the method of using of non-standard real environment map to plan the path. By placing calibration objects in the map, the generated non-standard planned paths can reflect the real environment, greatly improving the practicality of the path planning algorithm. In this paper, ant colony algorithms was used for path planning of non-standard map in a real environment, which greatly improved the intelligence of path planning algorithm. The experimental results of path planning based on ant colony algorithms for non-standard maps in virtual and real environments show that the method can realize path planning and pose planning of mobile robots in non-standard environments. Because the time of sensor measurement to obtain map was saved, the robot path planning was more efficient. Since the real environment map was used, the generated mobile robot path and pose were more intuitive. The research results of this article had great reference significance for improving the intelligence level of robot path planning methods in the later stage.

**Author Contributions:** Conceptualization, F.L.; Data curation, F.L.; Formal analysis, F.L. and Y.-C.K.; Investigation, F.L.; Methodology, F.L.; Resources, F.L.; Software, F.L.; Supervision, B.X.; Validation, F.L.; Visualization, B.X.; Writing—original draft, F.L. and Y.-C.K.; Writing—Review and Editing, F.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Science and Technology Research Project of Henan Province, China (No. 222102210307, No. 182102210510).

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data available on request due to restrictions, e.g., privacy or ethical.

**Conflicts of Interest:** The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Song, Q.; Li, S.; Yang, J.; Bai, Q.; Hu, J.; Zhang, X.; Zhang, A. Intelligent Optimization Algorithm-Based Path Planning for a Mobile Robot. *Comput. Intell. Neurosci.* **2021**, *2021*, 8025730. [CrossRef]
2. Deng, X.; Li, R.; Zhao, L.; Wang, K.; Gui, X. Multi-Obstacle Path Planning and Optimization for Mobile Robot, Expert Systems with Applications. *Expert Syst. Appl.* **2021**, *183*, 115445. [CrossRef]
3. Li, Y.; Li, J.; Zhou, W.; Yao, Q.; Nie, J.; Qi, X. Robot Path Planning Navigation for Dense Planting Red Jujube Orchards Based on the Joint Improved A\* and DWA Algorithms under Laser SLAM. *Agriculture* **2022**, *12*, 1445. [CrossRef]
4. Dong, L.; Yuan, X.; Yan, B.; Song, Y.; Xu, Q.; Yang, X. An Improved Grey Wolf Optimization with Multi-Strategy Ensemble for Robot Path Planning. *Sensors* **2022**, *22*, 6843. [CrossRef]
5. Lin, D.; Shen, B.; Liu, Y.; Alsaadi, F.E.; Alsaedi, A. Genetic Algorithm-Based Compliant Robot Path Planning: An Improved Bi-RRT-Based Initialization Method. *Assem. Autom.* **2017**, *37*, 261–270. [CrossRef]
6. Lu, L.; Han, J.; Dong, F.; Ding, Z.; Fan, C.; Chen, S.; Liu, H.; Wang, H. Joint-Smooth Toolpath Planning by Optimized Differential Vector for Robot Surface Machining Considering the Tool Orientation Constraints. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 2301–2311. [CrossRef]
7. Chen, N.; Zhang, Y.; Cheng, W. Space Detumbling Robot Arm Deployment Path Planning Based on Bi-FMT\* Algorithm. *Micromachines* **2021**, *12*, 1231. [CrossRef]
8. Feng, K.; He, X.; Wang, M.; Chu, X.; Wang, D.; Yue, D. Path Optimization of Agricultural Robot Based on Immune Ant Colony: B-Spline Interpolation Algorithm. *Math. Probl. Eng.* **2022**, *2022*, 2585910. [CrossRef]
9. Gong, L.; Yu, X.; Wang, J. Curve-Localizability-SVM Active Localization Research for Mobile Robots in Outdoor Environments. *Appl. Sci.* **2021**, *11*, 4362. [CrossRef]
10. Schafle, T.R.; Uchiyama, N. Probabilistic Robust Path Planning for Nonholonomic Arbitrary-Shaped Mobile Robots Using a Hybrid A\* Algorithm. *IEEE Access* **2021**, *9*, 93466–93479. [CrossRef]
11. Liu, L.; Wang, B.; Xu, H. Research on Path-Planning Algorithm Integrating Optimization A-Star Algorithm and Artificial Potential Field Method. *Electronics* **2022**, *11*, 3660. [CrossRef]
12. Sun, Y.; Zhang, C.; Liu, C. Collision-Free and Dynamically Feasible Trajectory Planning for Omnidirectional Mobile Robots Using a Novel B-Spline Based Rapidly Exploring Random Tree. *Int. J. Adv. Robot. Syst.* **2021**, *18*, 172988142110166. [CrossRef]
13. Zhang, H.; Tao, Y.; Zhu, W. Global Path Planning of Unmanned Surface Vehicle Based on Improved A-Star Algorithm. *Sensors* **2023**, *23*, 6647. [CrossRef]
14. Zhang, D.; Yin, Y.; Luo, R.; Zou, S. Hybrid IACO-A\*-PSO Optimization Algorithm for Solving Multiobjective Path Planning Problem of Mobile Robot in Radioactive Environment. *Prog. Nucl. Energy* **2023**, *159*, 104651. [CrossRef]
15. Tian, Z.; Guo, C.; Liu, Y.; Chen, J. An Improved RRT Robot Autonomous Exploration and SLAM Construction Method. In Proceedings of the 2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE), Dalian, China, 19–20 September 2020; pp. 612–619. Available online: <https://ieeexplore.ieee.org/document/9230216/> (accessed on 1 July 2023).
16. Zhang, L.; Shi, X.; Yi, Y.; Tang, L.; Peng, J.; Zou, J. Mobile Robot Path Planning Algorithm Based on RRT\_Connect. *Electronics* **2023**, *12*, 2456. [CrossRef]
17. Gao, Q.; Yuan, Q.; Sun, Y.; Xu, L. Path Planning Algorithm of Robot Arm Based on Improved RRT\* and BP Neural Network Algorithm. *J. King Saud Univ.—Comput. Inf. Sci.* **2023**, *35*, 101650. [CrossRef]
18. Tao, Y.; Wen, Y.; Gao, H.; Wang, T.; Wan, J.; Lan, J. A Path-Planning Method for Wall Surface Inspection Robot Based on Improved Genetic Algorithm. *Electronics* **2022**, *11*, 1192. [CrossRef]
19. Zhang, Z.; Lu, R.; Zhao, M.; Luan, S.; Bu, M. Robot Path Planning Based on Genetic Algorithm with Hybrid Initialization Method. *J. Intell. Fuzzy Syst.* **2022**, *42*, 2041–2056. [CrossRef]

20. Salvetti, F.; Angarano, S.; Martini, M.; Cerrato, S.; Chiaberge, M. Waypoint Generation in Row-Based Crops with Deep Learning and Contrastive Clustering. In *Machine Learning and Knowledge Discovery in Databases*; Amini, M.-R., Canu, S., Fischer, A., Guns, T., Novak, P.K., Tsoumakas, G., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2023; Volume 13718, pp. 203–218. [[CrossRef](#)]
21. Gong, F. Application of Artificial Intelligence Computer Intelligent Heuristic Search Algorithm. *Adv. Multimed.* **2022**, *2022*, 5178515. [[CrossRef](#)]
22. Zhao, J.C.; Ye, M. An Optimization Method for Satellite Data Structure Design Based on Improved Ant Colony Algorithm. *IEEE Access* **2023**, *11*, 64941–64956. [[CrossRef](#)]
23. Wu, S.; Li, Q.; Wei, W. Application of Ant Colony Optimization Algorithm Based on Triangle Inequality Principle and Partition Method Strategy in Robot Path Planning. *Axioms* **2023**, *12*, 525. [[CrossRef](#)]
24. Cheng, J. Dynamic Path Optimization Based on Improved Ant Colony Algorithm. *J. Adv. Transp.* **2023**, *2023*, 7651100. [[CrossRef](#)]
25. Zhang, C.; Wang, H.; Fu, L.-H.; Pei, Y.-H.; Lan, C.-Y.; Hou, H.-Y.; Song, H. Three-Dimensional Continuous Picking Path Planning Based on Ant Colony Optimization Algorithm. *PLoS ONE* **2023**, *18*, e0282334. [[CrossRef](#)] [[PubMed](#)]
26. Fei, T.; Wu, X.; Zhang, L.; Zhang, Y.; Chen, L. Research on improved ant colony optimization for traveling salesman problem. *Math. Biosci. Eng.* **2022**, *19*, 8152–8186. [[CrossRef](#)] [[PubMed](#)]
27. Dai, X.; Wei, Y. Application of Improved Moth-Flame Optimization Algorithm for Robot Path Planning. *IEEE Access* **2021**, *9*, 105914–105925. [[CrossRef](#)]
28. Wang, D.; Chen, S.; Zhang, Y.; Liu, L. Path Planning of Mobile Robot in Dynamic Environment: Fuzzy Artificial Potential Field and Extensible Neural Network. *Artif Life Robot.* **2021**, *26*, 129–139. [[CrossRef](#)]
29. Yan, Z.; Wang, Y.; Fan, J.; Huang, Y.; Zhong, Y. An Efficient Method for Optimizing Sensors' Layout for Accurate Measurement of Underground Ventilation Networks. *IEEE Access* **2023**, *11*, 72630–72640. [[CrossRef](#)]
30. Wang, X.; Ma, X.; Li, Z. Research on SLAM and Path Planning Method of Inspection Robot in Complex Scenarios. *Electronics* **2023**, *12*, 2178. [[CrossRef](#)]
31. Maldonado-Valencia, R.I.; Rodriguez-Garavito, C.H.; Cruz-Perez, C.A.; Hernandez-Navas, J.S.; Zabala-Benavides, D.I. Planning and Visual-Servoing for Robotic Manipulators in ROS. *Int. J. Intell. Robot. Appl.* **2022**, *6*, 602–614. [[CrossRef](#)]
32. Venkatesh, V.; Li, L.; McLinden, M.; Coon, M.; Heymsfield, G.M.; Tanelli, S.; Hovhannisyan, H. A Frequency Diversity Algorithm for Extending the Radar Doppler Velocity Nyquist Interval. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 2462–2470. [[CrossRef](#)]
33. Cui, S.; Wang, S.; Wang, R.; Zhang, S.; Zhang, C. Learning-Based Slip Detection for Dexterous Manipulation Using GelStereo Sensing. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–10. [[CrossRef](#)]
34. Cao, Y.; Nor, N.M.; Samad, Z. An Assistant Algorithm Model for a Mobile Robot to Pass through a Concave Obstacle Area. *SN Appl. Sci.* **2023**, *5*, 219. [[CrossRef](#)]
35. Abdulsahab, J.A.; Kadhim, D.J. Classical and Heuristic Approaches for Mobile Robot Path Planning: A Survey. *Robotics* **2023**, *12*, 93. [[CrossRef](#)]
36. Zhang, D.; Luo, R.; Yin, Y.; Zou, S. Multi-Objective Path Planning for Mobile Robot in Nuclear Accident Environment Based on Improved Ant Colony Optimization with Modified A\*. *Nucl. Eng. Technol.* **2023**, *55*, 1838–1854. [[CrossRef](#)]
37. Bosdelekidis, V.; Bryne, T.H.; Sokolova, N.; Johansen, T.A. Navigation Algorithm-Agnostic Integrity Monitoring Based on Solution Separation with Constrained Computation Time and Sensor Noise Overbounding. *J. Intell. Robot. Syst.* **2022**, *106*, 7. [[CrossRef](#)]
38. Li, W.; Tan, M.; Wang, L.; Wang, Q. A Cubic Spline Method Combing Improved Particle Swarm Optimization for Robot Path Planning in Dynamic Uncertain Environment. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 172988141989166. [[CrossRef](#)]
39. Yang, Y.; Chen, Z. Optimization of Dynamic Obstacle Avoidance Path of Multirotor UAV Based on Ant Colony Algorithm. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1299434. [[CrossRef](#)]
40. Condori, M.; Albesa, F.; Altobelli, F.; Duran, G.; Sorrentino, C. Image Processing for Monitoring of the Cured Tobacco Process in a Bulk-Curing Stove. *Comput. Electron. Agric.* **2020**, *168*, 105113. [[CrossRef](#)]
41. Miao, Z.; Yu, X.; Li, N.; Zhang, Z.; He, C.; Li, Z.; Deng, C.; Sun, T. Efficient Tomato Harvesting Robot Based on Image Processing and Deep Learning. *Precis. Agric.* **2023**, *24*, 254–287. [[CrossRef](#)]
42. Banafian, N.; Fesharakifard, R.; Menhaj, M.B. Precise Seam Tracking in Robotic Welding by an Improved Image Processing Approach. *Int. J. Adv. Manuf. Technol.* **2021**, *114*, 251–270. [[CrossRef](#)]
43. Gong, C.; Yang, Y.; Yuan, L.; Wang, J. An Improved Ant Colony Algorithm for Integrating Global Path Planning and Local Obstacle Avoidance for Mobile Robot in Dynamic Environment. *Math. Biosci. Eng.* **2022**, *19*, 12405–12426. [[CrossRef](#)] [[PubMed](#)]
44. Hou, W.; Xiong, Z.; Wang, C.; Chen, H. Enhanced Ant Colony Algorithm with Communication Mechanism for Mobile Robot Path Planning. *Robot. Auton. Syst.* **2022**, *148*, 103949. [[CrossRef](#)]
45. Qu, F.; Yu, W.; Xiao, K.; Liu, C.; Liu, W. Trajectory Generation and Optimization Using the Mutual Learning and Adaptive Ant Colony Algorithm in Uneven Environments. *Appl. Sci.* **2022**, *12*, 4629. [[CrossRef](#)]
46. Wang, J.; Xu, Z.; Zheng, X.; Liu, Z. A Fuzzy Logic Path Planning Algorithm Based on Geometric Landmarks and Kinetic Constraints. *Inf. Technol. Control* **2022**, *51*, 499–514. [[CrossRef](#)]
47. Loyola, O.; Kern, J.; Urrea, C. Novel Algorithm for Agent Navigation Based on Intrinsic Motivation Due to Boredom. *Inf. Technol. Control* **2021**, *50*, 485–494. [[CrossRef](#)]
48. Yue, M.; Ning, Y. WIP Vehicle Control Method Based on Improved Artificial Potential Field Subject to Multi-Obstacle Environment: WIP Vehicle Control Method Based on Improved Artificial Potential Field. *Inf. Technol. Control* **2020**, *49*, 320–334. [[CrossRef](#)]

49. Zagradjanin, N.; Rodic, A.; Pamucar, D.; Pavkovic, B. Cloud-Based Multi-Robot Path Planning in Complex and Crowded Environment Using Fuzzy Logic and Online Learning. *Inf. Technol. Control* **2021**, *50*, 357–374. [[CrossRef](#)]
50. Wei Wei, W.W.; Wei Wei, F.G.; Fan Gao, R.S.; Rafał Scherer, R.D.; Robertas Damasevicius, D.P. Design and Implementation of Autonomous Path Planning for Intelligent Vehicle. *J. Internet Technol.* **2021**, *22*, 957–964. [[CrossRef](#)]
51. Ge, H.; Ying, Z.; Chen, Z.; Zu, W.; Liu, C.; Jin, Y. Improved A\* Algorithm for Path Planning of Spherical Robot Considering Energy Consumption. *Sensors* **2023**, *23*, 7115. [[CrossRef](#)]
52. Li, X.; Wang, L.; An, Y.; Huang, Q.-L.; Cui, Y.-H.; Hu, H.-S. Dynamic Path Planning of Mobile Robots Using Adaptive Dynamic Programming. *Expert Syst. Appl.* **2023**, *235*, 121112. [[CrossRef](#)]
53. Na, Y.; Li, Y.; Chen, D.; Yao, Y.; Li, T.; Liu, H.; Wang, K. Optimal Energy Consumption Path Planning for Unmanned Aerial Vehicles Based on Improved Particle Swarm Optimization. *Sustainability* **2023**, *15*, 12101. [[CrossRef](#)]
54. Yu, L.; Wang, X.; Hou, Z.; Du, Z.; Zeng, Y.; Mu, Z. Path Planning Optimization for Driverless Vehicle in Parallel Parking Integrating Radial Basis Function Neural Network. *Appl. Sci.* **2021**, *11*, 8178. [[CrossRef](#)]
55. Ma, H.; Pei, W.; Zhang, Q. Research on Path Planning Algorithm for Driverless Vehicles. *Mathematics* **2022**, *10*, 2555. [[CrossRef](#)]
56. Zhang, H.; Sun, J.; Yang, B.; Shi, Y.; Li, Z. Optimal Search and Rescue Route Design Using an Improved Ant Colony Optimization. *Inf. Technol. Control* **2020**, *49*, 438–447. [[CrossRef](#)]
57. Barzegar, A.; Doukhi, O.; Lee, D.-J. Design and Implementation of an Autonomous Electric Vehicle for Self-Driving Control under GNSS-Denied Environments. *Appl. Sci.* **2021**, *11*, 3688. [[CrossRef](#)]

**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.