agriculture

Article

Research on Navigation Path Extraction and Obstacle Avoidance Strategy for Pusher Robot in Dairy Farm

Fuyang Tian [1,2], Xinwei Wang [2,3], Sufang Yu [4], Ruixue Wang [5], Zhanhua Song [1,2], Yinfa Yan [1,2], Fade Li [1,2], Zhonghua Wang [6] and Zhenwei Yu [1,2,*]

[1] College of Mechanical and Electronic Engineering, Shandong Agricultural University, Tai'an 271018, China; fytian@sdau.edu.cn (F.T.); szh@sdau.edu.cn (Z.S.); yanyinfa@sdau.edu.cn (Y.Y.); lifade@sdau.edu.cn (F.L.)
[2] Shandong Provincial Key Laboratory of Horticultural Machineries and Equipment, Tai'an 271018, China; 2021110433@sdau.edu.cn
[3] Shandong Provincial Engineering Laboratory of Agricultural Equipment Intelligence, Tai'an 271018, China
[4] College of Life Sciences, Shandong Agricultural University, Tai'an 271018, China; sfyu@sdau.edu.cn
[5] Chinese Academy of Agricultural Mechanization Sciences, Beijing 100083, China; wangruixue@caams.org.cn
[6] College of Animal Science and Technology, Shandong Agricultural University, Tai'an 271018, China; zhwang@sdau.edu.cn
[*] Correspondence: zhenweiyu615@sdau.edu.cn

Abstract: Existing push robots mainly use magnetic induction technology. These devices are susceptible to external electromagnetic interference and have a low degree of intelligence. To make up for the insufficiency of the existing material pushing robots, and at the same time solve the problems of labor-intensive, labor-intensive, and inability to push material in time at night, etc., in this study, an autonomous navigation pusher robot based on 3D lidar is designed, and an obstacle avoidance strategy based on the improved artificial potential field method is proposed. Firstly, the 3D point cloud data of the barn is collected by the self-designed pushing robot, the point cloud data of the area of interest is extracted using a direct-pass filtering algorithm, and the 3D point cloud of the barn is segmented using a height threshold. Secondly, the Least-Squares Method (LSM) and Random Sample Consensus (RANSAC) were used to extract fence lines, and then the boundary contour features were extracted by projection onto the ground. Finally, a target influence factor is added to the repulsive potential field function to determine the principle of optimal selection of the parameters of the improved artificial potential field method and the repulsive direction, and to clarify the optimal obstacle avoidance strategy for the pusher robot. It can verify the obstacle avoidance effect of the improved algorithm. The experimental results showed that under three different environments: no noise, Gaussian noise, and artificial noise, the fence lines were extracted using RANSAC. Taking the change in the slope as an indicator, the obtained results were about −0.058, 0.058, and −0.061, respectively. The slope obtained by the RANSAC method has less variation compared to the no-noise group. Compared with LSM, the extraction results did not change significantly, indicating that RANSAC has a certain resistance to various noises, but RANSAC performs better in extraction effect and real-time performance. The simulation and actual test results show that the improved artificial potential field method can select reasonable parameters and repulsive force directions. The optimized path increases the shortest distance of the obstacle point cloud from the navigation path from 0.18 to 0.41 m, where the average time is 0.059 s, and the standard deviation is 0.007 s. This shows that the optimization method can optimize the path in real time to avoid obstacles, basically meet the requirements of security and real-time performance, and effectively avoid the local minimum problem. This research will provide corresponding technical references for pusher robots to overcome the problems existing in the process of autonomous navigation and pushing operation in complex open scenarios.

Keywords: dairy farm; pusher robot; path extraction; obstacle avoidance

check for updates

Citation: Tian, F.; Wang, X.; Yu, S.; Wang, R.; Song, Z.; Yan, Y.; Li, F.; Wang, Z.; Yu, Z. Research on Navigation Path Extraction and Obstacle Avoidance Strategy for Pusher Robot in Dairy Farm. *Agriculture* **2022**, *12*, 1008. https://doi.org/10.3390/agriculture12071008

Academic Editor: Tomas Norton

Received: 23 May 2022
Accepted: 6 July 2022
Published: 12 July 2022

1. Introduction

Dairy farming is an indispensable part of modern agriculture, which occupies a high proportion in the agricultural industry [1,2]. In China, the traditional technology of dairy farming is relatively backward, and most of them adopt the management mode of small-scale scattered farming, which is not conducive to the development of modern agriculture; as a result, the mode is changing toward the direction of large-scale, factory, and standardization. [3,4]. In the past few years, China's dairy industry has developed rapidly, and its output value and scale are at the forefront of the world [5]. According to the data released by the Ministry of Agriculture and Rural Affairs, China's milk production in 2020 was 35.3 million tons, an increase of 7% over 2015, on the other hand, the proportion of farming with more than 100 heads reached 67.2%, an increase of 18.9% compared with 2015. In this situation, it is undeniable that the dairy industry not only meets the residents' consumption demand for milk, but also increases the income of dairy farmers. In addition, it plays a key role in optimizing the rural industrial structure [6].

The continuous prosperity of the social economy makes the public put forward higher requirements for the quality of dairy products, which indirectly promotes the development of the dairy industry [7,8]. However, the rapid development has also exposed new problems, operators gradually found that the existing high-tech aquatic products could not meet the production needs. For example, in the process of cow feeding, part of the feed will be removed from the feeding area due to the cow's activity, resulting in accumulation, which will lead to the deterioration of uneaten feed in the long run. The current solution is to use manual or manual pushing equipment to push the accumulated feedback into the feeding area. In this situation, enterprises need to arrange more labor or equipment to promote feed [9]. Relying on manual labor will make it impossible for the farm to complete the feeding work in a timely and stable manner; as a result, the milk yield of the cow will be reduced. In this case, the robot used to push feed is very practical.

The accuracy and execution efficiency of multimedia target recognition technology have been greatly improved with the development of deep learning (DL) and machine learning, under the circumstances [10–12]; the application of the technology has been extended to the fields of medical imaging [13], video surveillance [14], and robot navigation [15]. In the wave of technological change, traditional agricultural machinery has ushered in a new opportunity for development. Agricultural robots such as feeding robots, transport robots, and picking robots have begun to apply DL and machine learning techniques [16–18]. Among them, the self-propelled robot has been favored by many scholars as a new research hotspot. Some researchers have studied the technical difficulties of navigating the path extraction of agricultural robots based on visual geometry inference and DL [19]. The classical methods to infer visual geometry include simultaneous localization, mapping, and motion structure. This kind of technology obtains parameter values through sensors such as optical detection and ranging (LiDAR), sound navigation and ranging, optical flow, and stereo and monocular cameras, and uses corresponding algorithms for obstacle avoidance and path planning [20]. Among similar sensors, Lidar has the advantages of high-ranging accuracy, good resolution, and a strong anti-jamming ability. It has been widely used in the perception and extraction of agricultural indoor environmental information, and has become a research hotspot for agricultural production robots [21]. In the research field of push robots, new technologies continue to emerge. DeLaval has developed an automatic mixing and pushing robot using magnetic induction technology, which can independently plan the walking route and speed, and is suitable for automatic mixing and the pushing of different types and quantities of feed. Pavkin et al. [22] concentrated on the simulation modeling of a feed pusher robot using Simulink tools in the Matlab environment to facilitate robot modernization or optimize the final cost for artificial testing of typical system elements and reduce production costs. However, the application of Lidar in the bullpen has not been reported, but the research on bullpen path extraction and obstacle avoidance based on Lidar and machine vision has a certain application value.

At present, the existing research at home and abroad has solved the problem of navigation path extraction in some agricultural scenarios, but the working environment of dairy farms was rarely mentioned. In this study, a new type of machine vision system was developed to fill this gap. The system will be used for extraction and tracking control of the working path of the pusher robot. Taking the cowshed environment as the research object, the self-designed pusher robot and 3D lidar were used to collect the point cloud data of the cowshed. The ground point cloud was removed by point cloud preprocessing, and the pass-through filtering algorithm extracted the point cloud data of the region of interest. Then, the least-squares method (LSM) and random sample consensus (RANSAC) were used to extract fence lines, project them and obtain boundary contour features, and extract fence lines and initial paths. At the same time, a robot navigation path optimization and obstacle avoidance method based on the improved artificial potential field method is proposed, which will provide corresponding technical references for pusher robots to overcome the problems existing in autonomous navigation and pushing operations in complex open scenarios. The system designed in this study could autonomously generate accurate navigation paths for robots in a dynamic farm environment, which will provide technical reference for autonomous navigation of farming robots and the development of precision animal husbandry.

This paper is organized as follows: Section 2 details the materials and methods employed to achieve the research objective. In Section 3, experimental results and discussion of the proposed technique are presented. Finally, in Section 4, the conclusion and future work is provided.

2. Materials and Methods

2.1. The Composition of the Pusher Robot System in the Farm

The pusher robot needs to replace the labor for the feeding process, thereby reducing the feed cost and labor intensity of feeding dairy cows. The pusher robot can meet the functions of autonomous walking and pushing. Therefore, the pusher robot was mainly composed of a vehicle navigation hardware system, pusher operation system, and navigation and operation control system. Among them, the vehicle navigation hardware system and the pushing operation system were the specific execution systems of the instructions, which were responsible for receiving and executing the task instructions issued by the control system to complete the autonomous navigation and pushing operation. The navigation and operation control system were responsible for setting the working mode of the vehicle system, issuing target point instructions, displaying the robot position in real time, and controlling the pushing operation system. Through the fusion and analysis of various sensor information, the pusher robot could realize autonomous positioning and navigation in the natural environment.

The hardware device and the control system communicated in real time via a wireless network to complete the autonomous navigation and operation tasks of the pusher robot on the farm together, as shown in Figure 1. The vehicle navigation hardware system mainly included a robot chassis, drive module, control module, environmental information perception module, communication module, and power supply module.

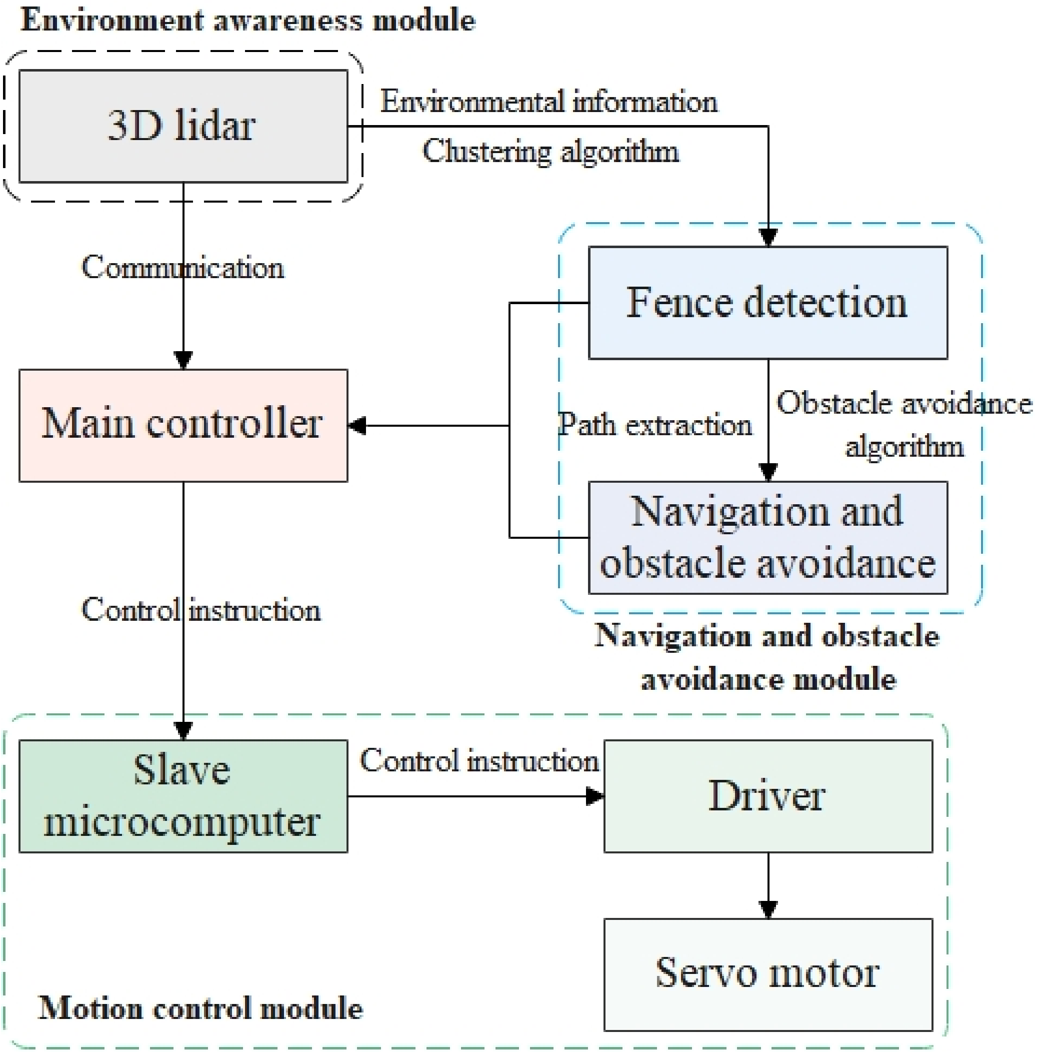


Figure 1. The vehicle navigation hardware system.

According to the task of the farm operation and the needs of the environment, the driving module of the robot adopted a two-wheel differential drive structure, and the steering control of the robot could be realized by setting different speeds for the two driving wheels. This drive system was not only simple in structure and small in turning radius, but also more flexible in movement, which greatly improved the control accuracy of the whole machine. The power system was provided by 60 V lithium battery modules. In order to ensure that the robot had powerful computing functions, the main control unit used Jetson Nano development board (NVIDIA, Shanghai, China), equipped with Tegra X1 heterogeneous SOC (NVIDIA, Shanghai, China), the size of this unit was 100 mm × 80 mm × 29 mm. The basic framework of ROS navigation was built under the Ubuntu 18.04 system, and information was exchanged with the chassis using RS-485 communication. The generated signal was transmitted to the main control unit via USB3.0. The car was equipped with the STM32F103 embedded motherboard (Haoyao, Shenzhen, China) as the underlying controller. According to the speed information provided by the encoder, the odometer data (moving speed, driving distance, and turning angle) of the vehicle system were obtained through kinematics calculation. Finally, the control of the vehicle-mounted system and the pushing operation system was completed through the control algorithm.

The environmental information perception module uses a 16-beam miniature LiDAR (RS-LiDAR-16, Sagitar Juchuang, Shenzhen Sagitar Juchuang Technology Co., Ltd., Shenzhen China). The compact housing of the RS-LiDAR-16, mounted with 16 laser/detector pairs, rapidly spins and sends out high-frequency laser pulses to continuously scan the surrounding environment, collecting real-time 3D point clouds. The 3D space point cloud data and object reflectivity are provided by the distance measurement algorithm, so that the pusher robot can digitally model the cowshed, providing a strong guarantee for its positioning, navigation, and obstacle avoidance. The lidar is installed in the center of the front end of the robot chassis, at a height of 0.6 m from the ground, and its performance parameters are shown in Table 1.

Table 1. Parameters of LiDAR.

Parameters	Values
Wavelength/nm	905
Detecting range/m	1–150
Accuracy/cm	±2
Vertical view/(°)	±15
Horizontal view/(°)	360
Vertical resolution/(°)	2.0
Horizontal resolution/(°)	0.1
Data rate/(pts·s^{-1})	6,000,000
Frame rate/Hz	10
Rotation speed/rpm	600

The 3D schematic diagram and physical map of the installation of each module of the pushing robot are shown in Figure 2. The overall length of the pusher robot is 1.78 m, the width is 1.15 m, the height is 1.40 m, and the rated load is 1 m^3.

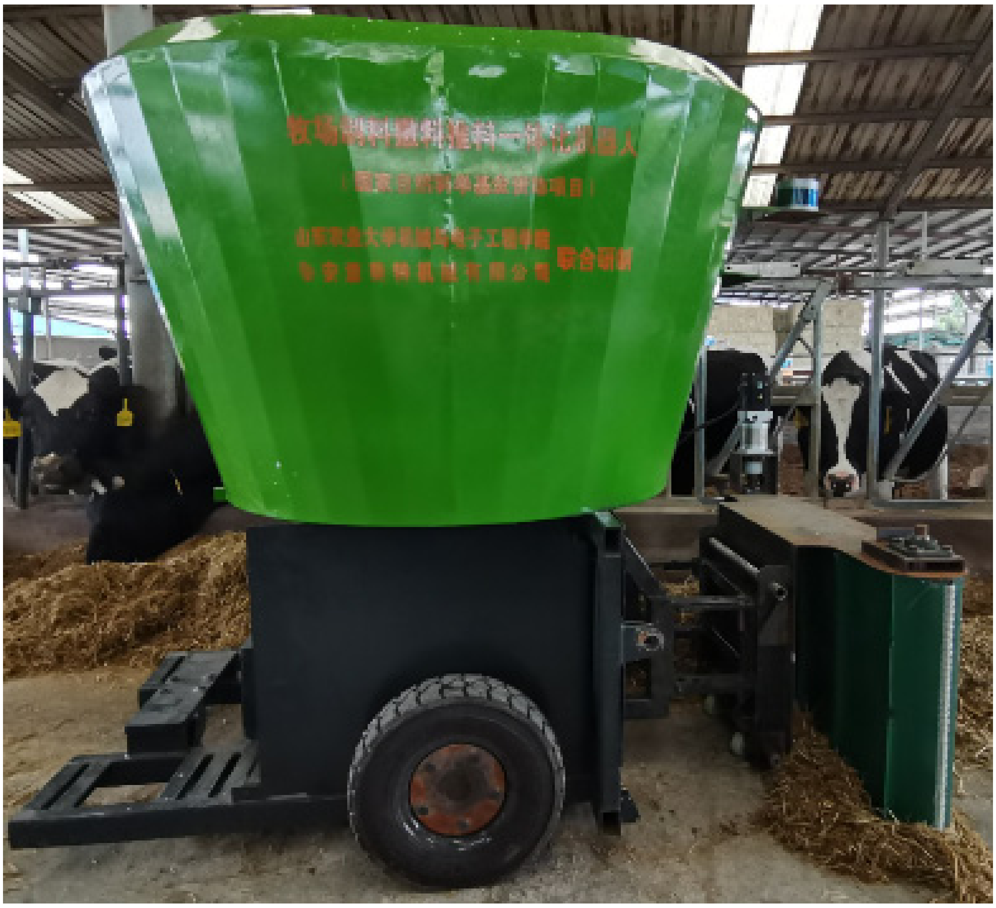

牧场饲料撒料推料一体化机器人
（国家自然科学基金资助项目）
山东农业大学机械与电子工程学院
联合研制

Figure 2. The Pusher Robot.

2.2. Collection and Preprocessing of Point Cloud Data

2.2.1. Collection of Point Cloud Data

The 3D point cloud data of cowsheds were collected in Jinlan Dairy Farm (Figure 3) in Tai'an City, Shandong Province, China from 16 to 30 October 2021. The cowshed is arranged in a double row with a distance (D) of 6.25 m between the two pens. The point clouds of the cowshed are unevenly distributed, with dense clouds at the near end and gradually sparse ones at the far end (Figure 3b). As shown in Figure 3b, taking the geometric center of the lidar as the origin point O, the forward direction of the robot is the positive direction of the *Y*-axis, the vertical *Y*-axis to the left is the positive direction of the *X*-axis, and the *Z*-axis is determined by the right-hand rule to establish a 3D lidar local coordinate system. This study extracts the 3D point cloud data in the range of *X*-axis (0–10 m), *Y*-axis (−20 to 10 m), and *Z*-axis (0–2 m) from the coordinate system as the region of interest (Region of Interest, ROI). Feeding of dairy cows will lead to the cluttered distribution of some of the far-end feeds, and the collected point cloud data will be messier. Therefore, it is necessary to filter out the ground point cloud to reduce the interference of the ground point cloud data on the initial path extraction.

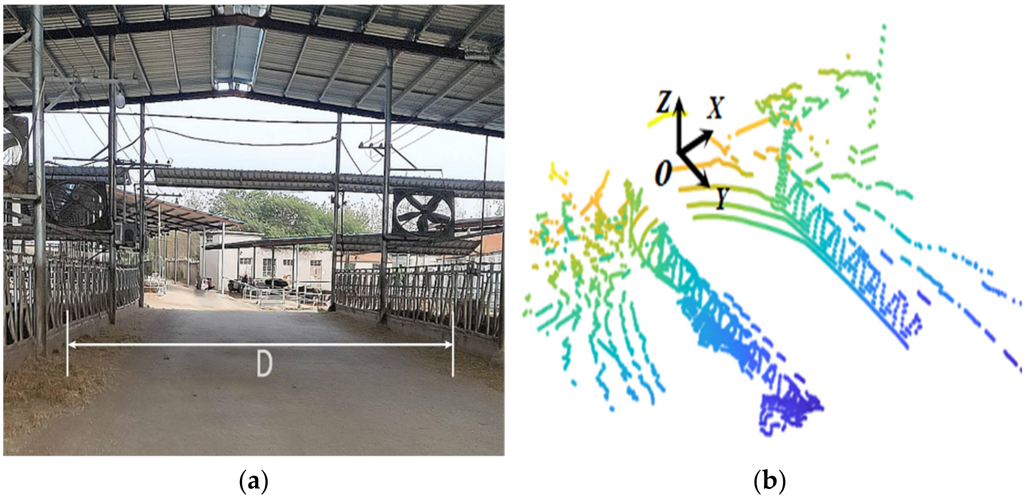


Figure 3. Cattle farm 3D point cloud acquisition: (**a**) Experimental cattle farm, (**b**) 3D point cloud of the original cattle farm.

2.2.2. Preprocessing of Point Cloud Data

There are about 16,000 points in each frame of the collected 3D point cloud data of the cowshed, which is a huge amount of data. In order to reduce the amount of calculation, it is first necessary to preprocess the 3D point cloud data of the cowshed to remove noise and outliers [23,24]. Then, use the pass-through filtering algorithm to extract the ROI point cloud; the centroid of the cube is used to represent all points in the cube, and the voxel filtering algorithm downsamples the point cloud to greatly reduce the number of 3D point clouds while preserving the structural features of the 3D point cloud data. Therefore, this study uses a cube with a side length of 0.1 m to downsample the ROI point cloud. There are still many noise points and outliers in the filtered 3D point cloud, so statistical filtering is used to remove the noise and outliers [25]. In order to reduce the interference of the ground point cloud on the cowshed outline extraction, the ground plane fitting (GPF) algorithm proposed in the literature [26] is used to segment the ground and non-ground point clouds.

2.3. Fence and Initial Path Extraction

To make the segmented fence show a better effect, the preprocessed bullpen 3D point cloud was segmented by a high threshold method. The height threshold was determined according to the actual height of the cowshed and empirical methods, and the height threshold here was set to 0.1 m. The fence point cloud has apparent line features. The fence point cloud is projected onto the XY plane, and the fence lines are extracted by LSM and RANSAC, respectively, and the extraction effects of the two are compared. Project the fence point cloud on the XY plane, extract the boundary contour features of the fence point cloud, and calculate the navigation path of the pusher robot according to the fence's boundary outline to improve the mobile robot's accuracy in pushing grass during the operation.

2.3.1. The Least-Squares Method

LSM is a mathematical tool that has been widely used in many disciplines of data processing such as error estimation, system identification and prediction, and forecasting. It finds the best function parameters for point cloud data by minimizing the sum of squared errors. The basic principle is as follows: data $\{(x_i, y_i), i = 1, 2, \ldots, m\}$, obtain the data fitting function $\varphi(x)$. Then, the fitting function $\varphi(x)$ should reflect the changing trend of all data as much as possible, but it is not required to pass all data points; that is to say, there is a certain error between the fitting function $\varphi(x)$ and the actual measured data at x_i. Here, it is represented by ε_i:

$$\varepsilon_i = \varphi(x_i) - f(x_i)(i = 1, \ldots, n)$$

In order to meet the requirement that the fitting function curve can reflect the change trend of all data as much as possible, its 2-norm is required to be a minimum.

$$\parallel E \parallel_2 = \left\{ \sum_{i=1}^{n} [\varphi(x_i) - f(x_i)]^2 \right\}^{\frac{1}{2}}$$

where $\parallel E \parallel_2$ is the 2 norm of error.

In order to facilitate calculation, analysis, and application, the square of 2 norm is usually calculated, namely:

$$\parallel E \parallel_2^2 = \sum_{i=1}^{n} [\varphi(x_i) - f(x_i)]^2$$

This fitting method, which requires the minimum sum of squares of errors, is called the least-squares method.

The Fence Fitting Line was extracted by LSM fitting the point clouds on both sides of the mobile robot's driving direction. When the point cloud coordinates satisfy the minimum value of $F(W)$, W is the parameter matrix of the fitted Fence Fitting Line equation, as shown in Equation (1):

$$F(W) = \min\left(X'W - Y\right)^T \left(X'W - Y\right) \tag{1}$$

where $W = [k\ d]^T$ is the parameter matrix of the fence line; k is the slope of the fence line; d is the fence line intercept, and $m : {X'}_{n\times 2} = [X\ I]$ is the matrix composed of the point cloud X-axis coordinate value matrix $X_{n\times 1}$=$[x_1\ x\ _2 \ldots\ x_n]^T$ and the unit matrix I; $Y_{n\times 1} = [y_1 y_2\ \ldots\ y_n]^T$ is the matrix composed of the Y coordinate values of the point cloud. Taking the derivative of Equation (1), when $X\prime^T X''$ is a positive definite matrix, the parameter matrix W of the fence line equation is shown in Equation (2):

$$W = \left(X'^T X'\right)^{-1} X'^T Y \tag{2}$$

2.3.2. Random Sampling Consistency

The RANSAC method can iteratively estimate the parameters of a mathematical model from a set of observational data sets containing "outliers" [27,28]. The random sampling consensus algorithm can well estimate the model parameters from the data containing a large number of outliers, and can eliminate the interference of outliers on the estimated overall data model, and obtain the global optimal solution. It is an indeterminate algorithm.

It has a certain probability of producing a good result, and to increase the likelihood, the number of iterations must be increased.

The purpose of RANSAC is to find the optimal parameter matrix so that the number of data points that satisfy the matrix is the largest. Usually, h_{33} = 1 is used to normalize the matrix. Since the homography matrix has 8 unknown parameters, at least 8 linear equations are needed to solve, corresponding to the point position information, a set of point pairs can list two equations, and at least 4 sets of matching point pairs are included. The resulting matrix equation is shown in Equation (3):

$$s\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11}h_{12}h_{13} \\ h_{21}h_{22}h_{23} \\ h_{31}h_{32}h_{33} \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3}$$

where S represents the sample data and h_{ij} represents a single element in the normalized matrix.

The RANSAC algorithm randomly selects 4 samples from the matching data set and ensures that the 4 samples are not collinear, calculates the homography matrix, then uses this model to test all data, and calculates the number and projection of data points that satisfy this model. Error (i.e., cost function), if this model is the optimal model, the corresponding cost function is the smallest. The resulting loss function is shown in Equation (4):

$$L = \sum_{i=1}^{n}\left(x_i', \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}}\right)^2 + \left(y_i', \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}}\right)^2 \tag{4}$$

where $x_i{}'$, $y_i{}'$ are the elements in the parameter matrix; x_i, y_i are the elements in the surrogate matrix.

A matrix is obtained by random sampling, and using Equation (3), it is verified whether other points conform to the model, and then the conforming points become "internal points", and the nonconforming points become "external points". Next time, extract points from the "new interior point set" to construct a new matrix, and recalculate the error. The final error is the smallest, and the maximum number of points is the final model. The steps of the RANSAC algorithm:

(1) Randomly extract S sample data from the data set, fit multiple models (the 4 samples cannot be collinear), calculate the transformation matrix H, and record it as model M;

(2) Calculate the projection error of all data in the dataset and the model M, if the error is less than the threshold, add the inner point set I;

(3) If the number of elements in the current interior point set I is greater than the optimal interior point set I_best, then update I_ best = I, and update the number of iterations k as shown in Equation (5):

$$k = \frac{\log(1-p)}{\log(1-w^m)} \tag{5}$$

where p is the confidence level, which is generally taken as 0.995; w is the proportion of "inner points"; m is the minimum number of samples required to calculate the model;

(4) If the number of iterations is greater than k, exit; otherwise, add 1 to the number of iterations, and repeat the above steps.

RANSAC is used to extract the fence lines on both sides, and a subset is selected from the point clouds of the fences on both sides by random sampling to establish a straight-line model. Then, the number of interior points of the straight-line model is calculated to check the correctness of the straight-line model, and iterate continuously to obtain the optimal straight-line model, which is the extracted fence line (Figure 4).

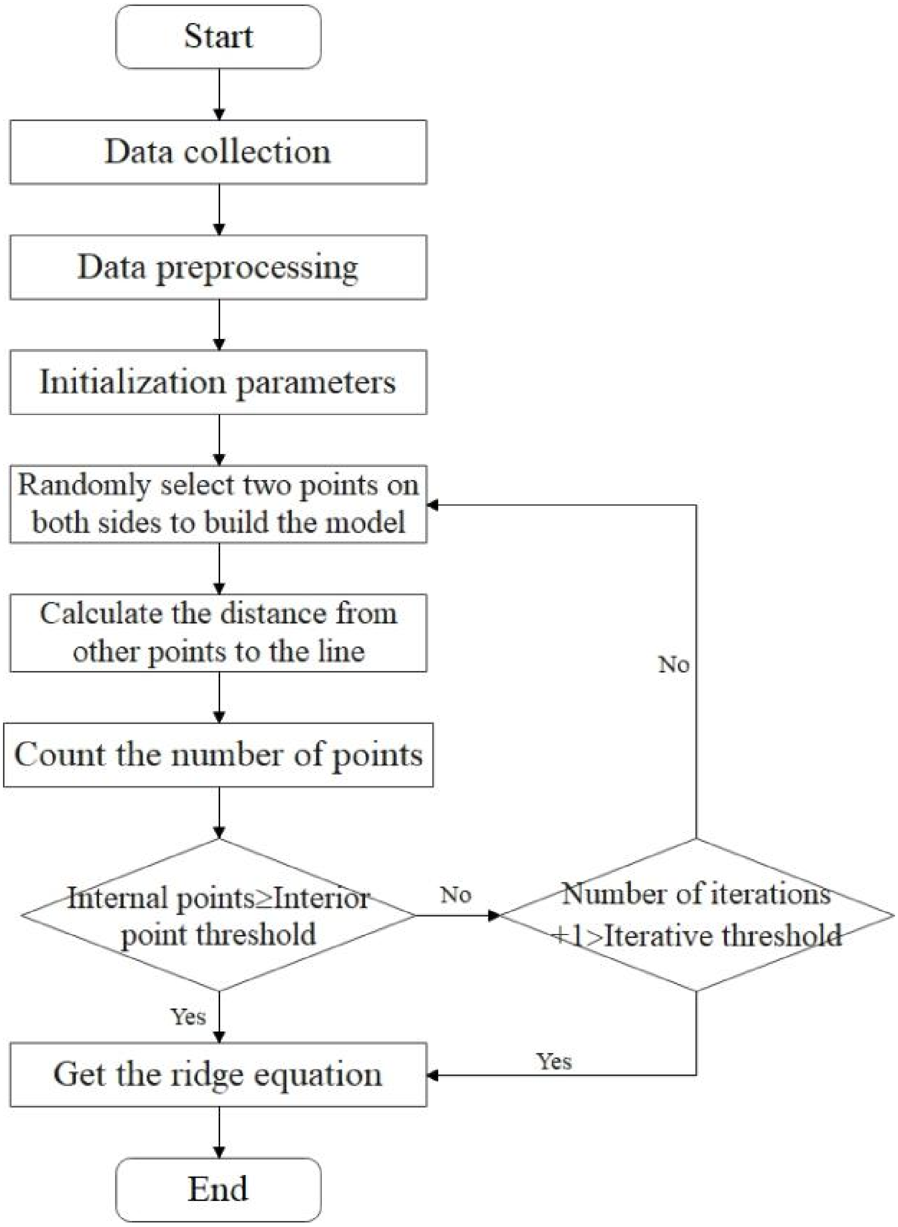

Start
Data collection
Data preprocessing
Initialization parameters
Randomly select two points on both sides to build the model
Calculate the distance from other points to the line
Count the number of points
Internal points≥Interior point threshold
No
Number of iterations +1>Iterative threshold
No
Yes
Get the ridge equation
Yes
End

Figure 4. Flow chart of ridgeline extraction based on RANSAC.

The iteration threshold K_{RANSAC} is a key parameter for random sampling fitting. If the value of K_{RANSAC} is set too large, it will take too long, and if the value of K_{RANSAC} is set too small, the fitting effect will be poor. The selection basis of the K value is shown in Equation (6):

$$K_{RANSAC} = \lg(1-\alpha)/\lg(1-\omega^N) \tag{6}$$

where α is the probability that all points selected in the iterative process are interior points, %; ω is the probability that an interior point is selected from the data, %; N is the total number of data points.

2.4. Work Path Extraction

2.4.1. Noise Processing

The real-time performance is evaluated by the processing time of extracting grid lines [29], and the anti-noise ability is the resistance ability of the fence line extraction method to two kinds of noise [30]. This study evaluates the effect of LSM and RANSAC in extracting fence lines from the aspects of real-time performance and anti-noise ability, so that the extraction method can have strong real-time performance and anti-noise ability. Add Gaussian noise and man-made noise to the fence point cloud; among them, Gaussian noise cancels the statistical filtering process, and processes the Gaussian noise with the mean value of 0 and the variance of 0.1 on the point cloud of the fence. The artificial noise is achieved by adding interfering points between two fence point clouds.

2.4.2. Path Extraction

The boundary contour data are extracted from the fence point cloud using the point cloud vector method. First, the fence point cloud and its adjacent points projected on the XY plane are fitted with straight lines. Then, a point P in the fence point cloud is selected, and its adjacent point cloud is set $P_k = \{P_0, P_1 \ldots P_{k-1}\}$, a straight line is set $u{\cdot}x + v{\cdot}y = c\left(u^2 + v^2 = 1\right)$, and the LSM is used to fit the straight line. At this time, the vector $q(u,\ v)$ is the normal vector of the point P. Then, referring to the content of the literature, the method where the maximum angle between adjacent points is greater than the set threshold is used to extract the fence outline point cloud, and the fence outline point cloud is recorded as the point set $U = \{d_0, d_1 \ldots d_{n-1}\}$. Finally, the positional relationship between the fence line and the fence outline data points is judged by Equation (7). After removing the data outside the fence line, the point cloud data inside the fence line are divided into the left point set (U_L) and the right point set (U_R).

$$\begin{cases} d_{li} = \frac{|k_l \cdot x_i + b_l - y_i|}{\sqrt{{k_l}^2 + {b_l}^2}} \\ d_{ri} = \frac{|k_r \cdot x_i + b_r - y_i|}{\sqrt{{k_r}^2 + {b_r}^2}} \\ U_L = \left\{ \begin{matrix} (k_l \cdot x_i + b_l - y_i) \cdot (k_r \cdot x_i + b_r - y_i) < \\ 0 \cap d_{li} - d_{ri} < 0 | d_i(x_i, y_i) \in U \end{matrix} \right\} \\ U_R = \left\{ \begin{matrix} (k_l \cdot x_i + b_l - y_i) \cdot (k_r \cdot x_i + b_r - y_i) < \\ 0 \cap d_{li} - d_{ri} > 0 | d_i(x_i, y_i) \in U \end{matrix} \right\} \end{cases} \tag{7}$$

In the formula, d_{li} and d_{ri} are the distance between the fence outline point $d_i(x_i,\ y_i)$ and the left and right fence lines, m; i is the index number of the point set, $i = 0, 1, 2, \ldots, n-1$; n is the number of point clouds of fence outline.

When the cows have been feeding for a period of time, the cows will push the forage to the outside at will, causing part of the forage to enter the area where they cannot eat, and the shape of the forage pile becomes irregular and the thickness of the pile becomes uneven. Therefore, the pusher robot is operated along the outermost part of the no-eating area, and the forage in the no-eating area is pushed to the eating area. As shown in Figure 5, during the operation of the pushing robot, the sideline of the auger always coincides with the inner boundary of the inaccessible area. The initial path is approximated by the translation transformation of the fence line fitted by the two methods. In this study, the intercept (b_1) of the initial path was used as the index, the width of the edible area was 70 cm, and the length of the auger of the pushing robot was 110 cm.

$$b_1 = b_0 + m_1 + \frac{1}{2} m_2 \tag{8}$$

where b_0 is the intercept of the fence line, m_1 is the width of the edible area, and m_2 is the length of the auger of the pushing robot.

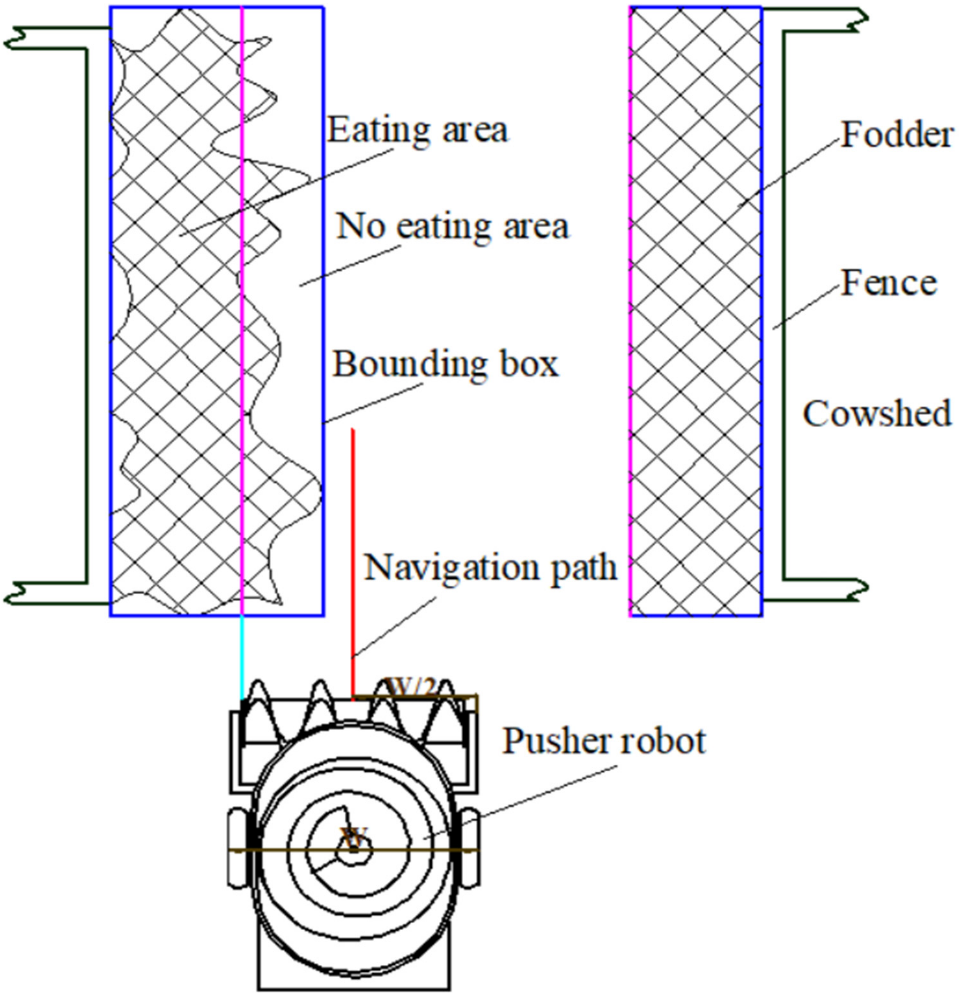


Figure 5. Working diagram of grass pushing robot.

2.5. Improvement of Obstacle Avoidance Strategy

The pusher robot will inevitably encounter obstacles during the operation of the cattle farm. The obstacle avoidance module is an essential part of the pusher robot to ensure that it can pass through obstacles during mobile operations. This paper chooses the artificial potential field method as the basis of the obstacle avoidance algorithm, and improves and analyzes the situation that it falls into the local optimum point.

2.5.1. Artificial Potential Field Method

Artificial Potential Field (APF) is an obstacle avoidance strategy represented by artificially defined virtual forces [31,32]. The mobile robot is assumed to be a point, which moves in a virtual force field, which is composed of the gravitational field of the target point to the robot and the repulsion field of the obstacle to the robot. The gravitational field is generated by the target point, and the repulsive force field is composed of the force field generated by all obstacles [33].

As shown in Figure 6, the repulsive force of the obstacle acting on the mobile machine is denoted as F_{rep}, and the direction is from the obstacle to the mobile robot; the gravitational force of the target point acting on the mobile robot is recorded as F_{att}, and the direction is from the mobile robot to the target point, then the force that the mobile robot receives at this position is the combined force of the repulsion force F_{rep} and the gravitational force F_{att} is F.

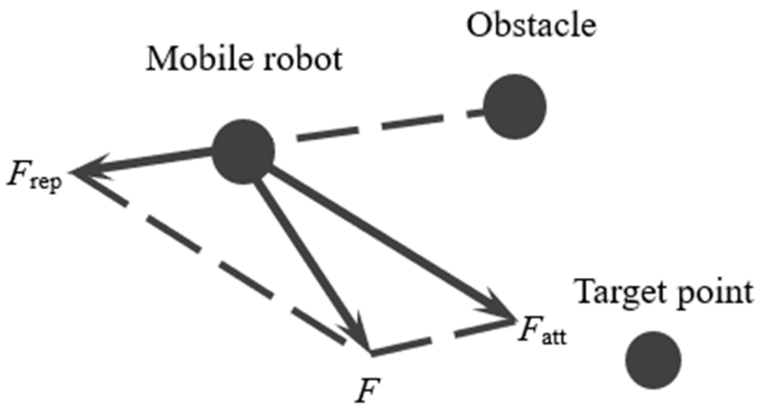


Figure 6. Stress diagram of artificial potential field method.

In the process of path planning, the environment of the unmanned vehicle is treated in a two-dimensional space, but the entire potential field distribution is three-dimensional. As shown in Figure 7, the gravitational potential energy leads to the generation of the third dimension, which is the main force in the process of path planning of the unmanned vehicle. The obstacles in the driving environment form peaks in the potential field distribution map. Under the action of the potential field, the unmanned vehicle can only move from the high potential energy point to the low potential energy point, so that the unmanned vehicle will not hit the obstacles, and it can safely plan the obstacle avoidance route.

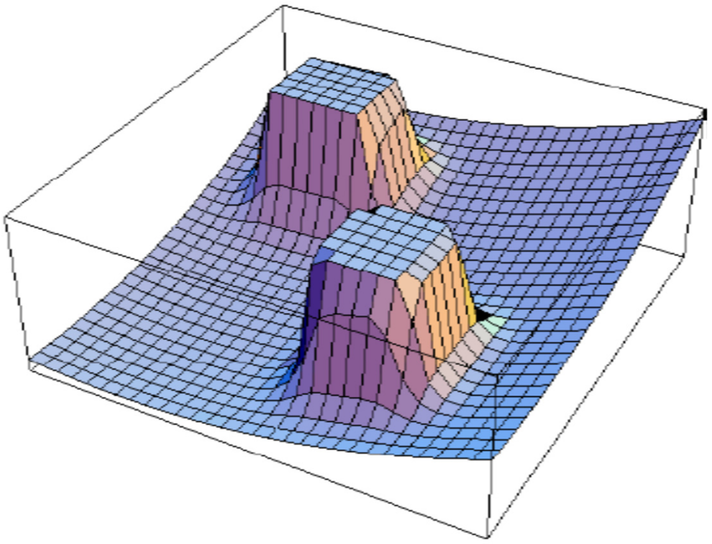


Figure 7. Three-dimensional diagram of artificial potential field obstacle avoidance.

Let the positions of the mobile robot, the target point, and the obstacle, be denoted as $q = (x, y)^T$ and $q_g = (x_g, y_g)$, respectively, and $q_{obs} = (x_{obs}, y_{obs})$ is the gravitational potential field generated by the target point to the mobile robot, and *Uatt*(*q*) is the repulsive potential field generated by the obstacle to the mobile robot.

When the mobile robot is far away from the target point, the target point should generate a larger gravitational force for the mobile robot to move the mobile robot towards the target point. At the same time, when the mobile robot is at the target point, the robot should be at the zero-force point, so the gravitational potential field function is expressed as:

$$U_{att}(q) = \frac{1}{2}\xi\rho^2(q, q_g) \tag{9}$$

where ξ is the gain coefficient of the gravitational field, and $\rho(q,\ q_g)$ represents the distance between the target point and the current position of the mobile robot (expressed in Euclidean distance).

2.5.2. Improvement of Artificial Potential Field Method

The artificial potential field method converts the complex environmental information around the mobile robot into a simple force field model, which can achieve a relatively good obstacle avoidance effect in general [34,35]. However, due to the limitations of the definition of the gravitational potential field function and the repulsive potential field function itself, there may be situations in which the set target cannot be reached as expected and local minima appear before reaching the set target point. The reason for the above situation is mainly due to the defects brought by the definition of the gravitational potential field function and the repulsive potential field function itself. If the gravitational and repulsive forces are zero when the mobile robot reaches the target point, then the target point is the global optimal point. Considering the above problems, the distance between the target point and the robot is introduced into the repulsion function, and the repulsion field function expression is redefined:

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left[\frac{1}{\rho(q,q_{obs})} - \frac{1}{\rho_0}\right]^2 \left(X - X_{goal}\right)^n, \rho(q,q_{obs}) \le \rho_0 \\ 0\,, \qquad \rho(q,q_{obs}) > \rho_0 \end{cases} \tag{10}$$

where $(X - X_{goal})$ is the distance between the robot and the target, and n is a constant and greater than 0. Similarly, the repulsive force on the mobile robot is the negative gradient of the repulsive force field, and the repulsive force F_{rep} (q) is expressed as:

$$F_{rep}(q) = \begin{cases} F_{rep1} + F_{rep2}, \rho(q,q_{obs}) \le \rho_0 \\ 0 \qquad , \rho(q,q_{obs}) > \rho_0 \end{cases} \tag{11}$$

$$F_{rep1} = \eta\left[\frac{1}{\rho(q,q_{obs})} - \frac{1}{\rho_0}\right]\frac{1}{\rho^2\rho(q,q_{obs})}\left(X - X_{goal}\right)^n \tag{12}$$

$$F_{rep2} = \frac{n}{2}\eta\left[\frac{1}{\rho(q,q_{obs})} - \frac{1}{\rho_0}\right]^2\left(X - X_{goal}\right)^{n-1} \tag{13}$$

In the formula, the direction of F_{rep1} is from the obstacle to the mobile robot, and the direction of F_{rep2} is from the target robot to the target point.

3. Results Analysis

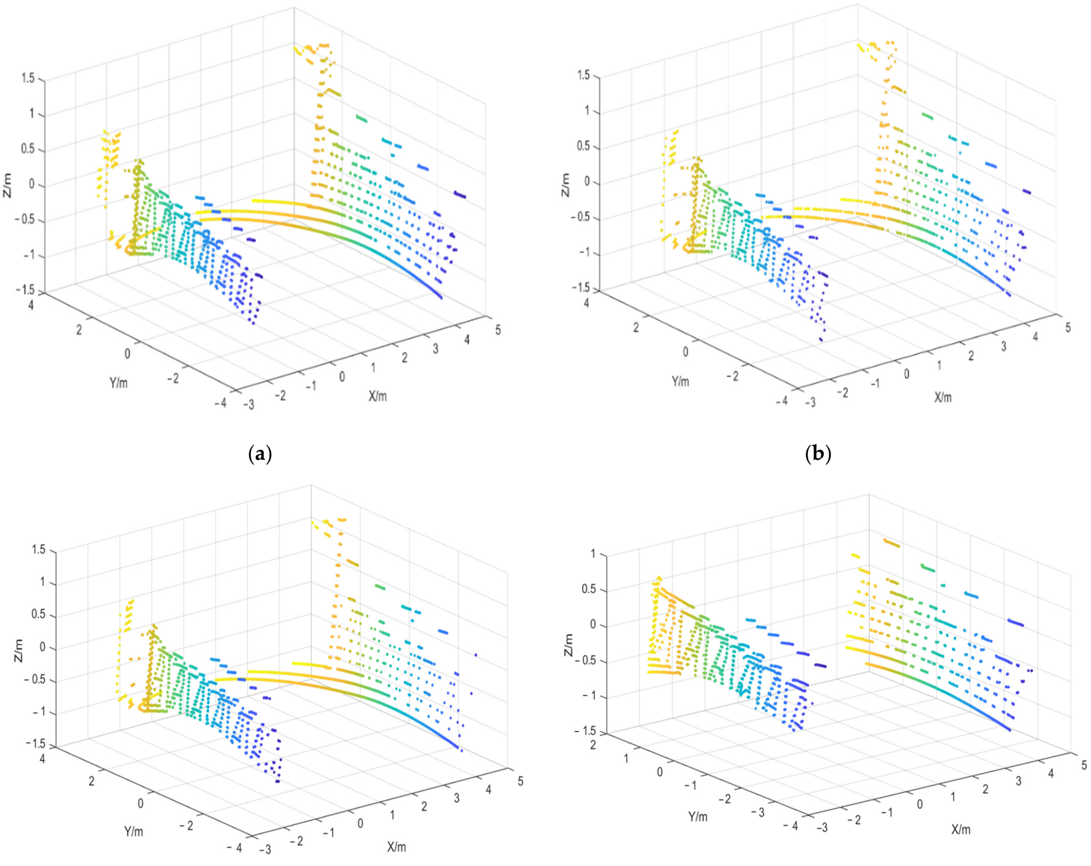
3.1. Cowshed Point Cloud Preprocessing Results

In order to clarify the influence of the preprocessing method of point cloud data on the pusher robot in different motion states, the pusher robot collected 3D point cloud data in static and moving (forward speed is 0.5 m/s) states. From the collected 3D point cloud data, 200 frames of point clouds were selected for preprocessing. The number of preprocessed point clouds and the processing time of the filtering algorithm are shown in Table 2. It can be seen from the table that the pre-processed average point cloud numbers of the data collected by the robot at rest and in motion were 3257 and 3249, and the total average processing time was 0.338 and 0.319 s. There was no significant change in the number of point clouds and the total average processing time, which indicated that the preprocessing method selected in this study was suitable for machines in different motion states. Comparing the processing time of through filtering, downsampling filtering, and statistical filtering, it was found that statistical filtering took the longest time (0.122 s), accounting for 37.5% of the total preprocessing time, which was not conducive to real-time processing.

Table 2. Cowshed point cloud preprocessing results.

Motion States	Preprocessing Methods	Number of Point Clouds	Processing Times/s
Static	Through the filter	9872	0.0763
	Statistical filter	8253	0.125
	Down-sampled filter	4777	0.0684
	Ground plane filter	3257	0.0641
Motion	Through the filter	9865	0.0653
	Statistical filter	8255	0.119
	Down-sampled filter	4765	0.0676
	Ground plane filter	3249	0.0642

The visualization results of the preprocessed 3D point cloud data are shown in Figure 8. The ROI point cloud was extracted by pass-through filtering (Figure 8a), and the number of processed point clouds was 9872, which was reduced by 40%, and significantly reduced the number of point clouds; the number of point clouds after voxel downsampling filtering was 4777, which was reduced by 70%, and still retained the structural features of the original point cloud data (Figure 8b); the number of point clouds after statistical filtering was 8253, and 7% of outliers were removed (Figure 8c); after fitting the ground plane, the number of point clouds was 3257, which reduced the ground point cloud data by 79%, and retained the fence information (Figure 8d).



(**a**) (**b**)

(**c**) (**d**)

Figure 8. Visualization diagram of point cloud preprocessing: (**a**) Through filtered point cloud, (**b**) Statistical filtered point cloud, (**c**) Down sampling filtered point cloud, (**d**) Point cloud after ground removal.

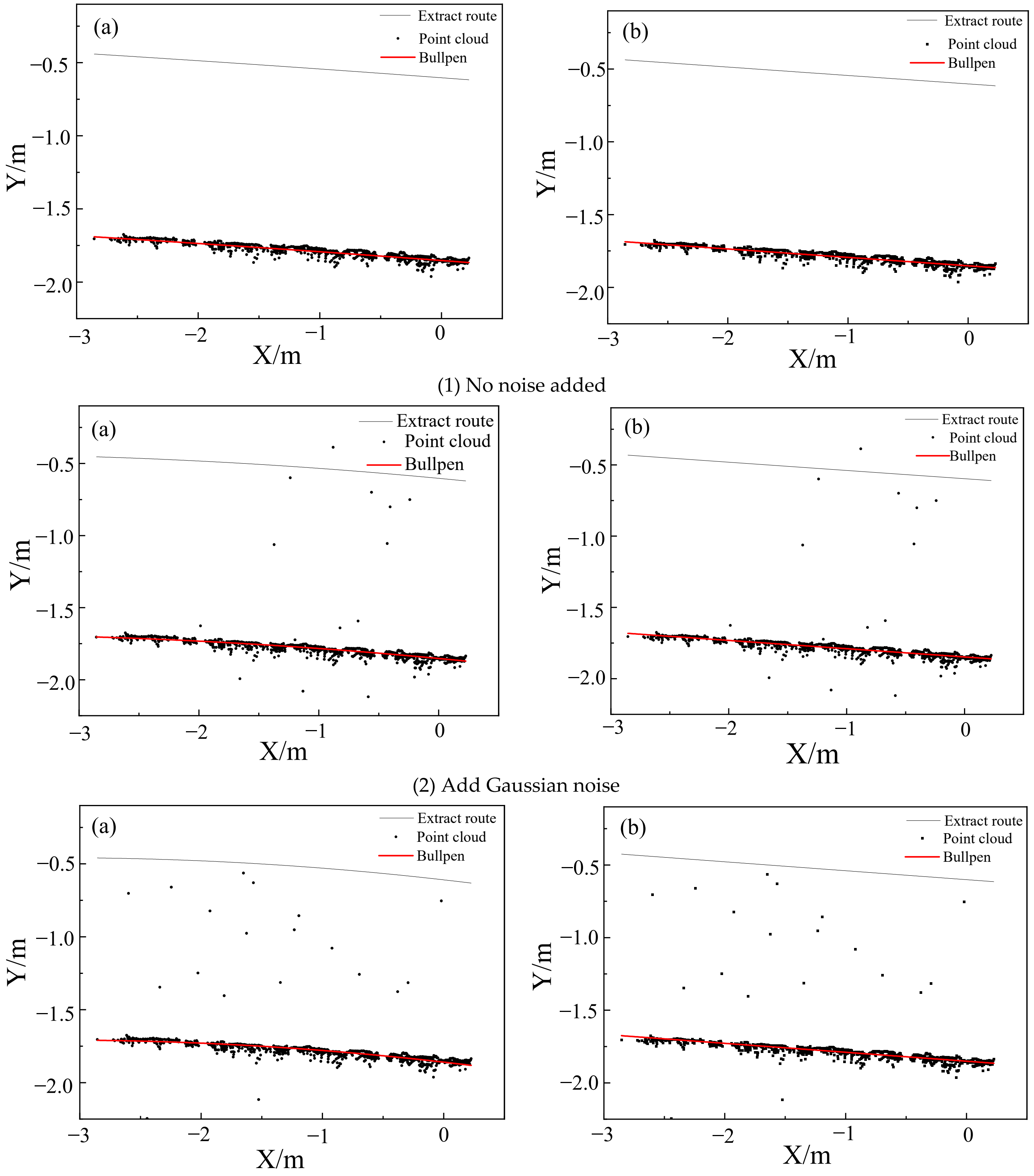
3.2. Extraction Results of Fence Lines and Paths

In the experiment, Gaussian noise and artificial noise were added to the cowshed point cloud data, and the LSM and RANSAC were used to extract the fence lines and the initial paths. The visualization results were shown in Figure 8 and Table 3. It could be found from Figure 9 that both methods could process the fence point cloud data, and the processing effect was better. From Table 3, the value of the slope of the right fence line extracted by the LSM was −0.095 after adding artificial noise. Compared with no noise added (the value of the slope was about −0.061), there was a larger error and the extraction effect was reduced. The LSM considers the shortest distance from the overall point cloud to the extraction line and the phenomenon that the extracted fence line deviates when many data are deviating from the fence point cloud. The results of the RANSAC extracting fence lines without adding noise, adding Gaussian noise, and adding artificial noise (take the value of the slope of the right fence line as an example) were about −0.058, 0.058, and −0.061, respectively, and there was no significant change, indicating that RANSAC has certain resistance to both Gaussian noise and man-made noise. It could be seen from Table 2 that after adding artificial noise, the intercept value of the initial path extracted by the LSM was about −0.610, and the group without noise (the intercept was about −0.603) was quite different; The intercepts value extracted by the RANSAC in the three groups were −0.602, −0.603, and −0.601, and the intercept changes were not obvious, indicating that the RANSAC was better for initial path extraction.

Table 3. Extraction of bullpen line and initial path analysis by LSM and RANSAC.

Point Cloud Noise	Methods	Intercept of Bullpen b_0/m	Line Slope of Bullpen k_0	Intercept of Initial Path b_1/m	Processing Time/10^{-3} s
Without noise	LSM	−1.853 ± 0.001	0.062 ± 0.002	−0.603 ± 0.001	2.352 ± 0.125
	RANSAC	−1.852 ± 0.001	0.056 ± 0.001	−0.602 ± 0.001	1.157 ± 0.012
Gaussian noise	LSM	−1.853 ± 0.005	0.079 ± 0.012	−0.603 ± 0.005	2.638 ± 0.119
	RANSAC	−1.847 ± 0.005	0.058 ± 0.037	−0.597 ± 0.005	1.162 ± 0.015
Artificial noise	LSM	−1.860 ± 0.007	0.095 ± 0.007	−0.610 ± 0.007	2.931 ± 0.124
	RANSAC	−1.851 ± 0.006	0.062 ± 0.005	−0.601 ± 0.006	1.165 ± 0.013

From the analysis of the processing time of the two methods, it could be found that there was no significant difference in the processing time of RANSAC in the non-noise group, the Gaussian noise group, and the artificial noise group. The processing time (2.352 × 10^{-3} s) of the LSM without noise group was significantly different from the other two groups, indicating that the processing time of RANSAC was less affected by noise than the LSM method. The running time of RANSAC is significantly lower than that of LSM transform, and RANSAC can obtain better real-time performance by selecting a reasonable number of iterations.



(1) No noise added

(2) Add Gaussian noise

(3) Add man-made noise

Figure 9. Results of extracted ridgeline and initial path by three methods under different noise. (**a**) Least squares method; (**b**) Random sampling consistency.

3.3. Simulation Analysis of Obstacle Avoidance Algorithm

In response to the problem that the traditional artificial potential field method is prone to the defect of falling into minimal values, a new repulsion field function is proposed, and the selection range of the repulsion field gain coefficient is analyzed. In order to verify the effectiveness of the improved artificial potential field method when there are multiple obstacles, and the influence of parameter selection on the obstacle avoidance effect. This paper conducts simulation experiments in MATLAB BR2020a. Since there are cows and feed belts on both sides of the fence, it is only necessary to place obstacles at the proximal and distal ends of the feed belts. The simulation space adopts a 10 × 10 grid map. It sets the coordinates of the starting position of the robot to be (0, 0) marked with a square, and the coordinates of the end point of the target point to be (10, 10), marked with a triangle, and the obstacle coordinate points marked with circles are set between the two to simulate the actual situation.

As shown in Figure 10, five obstacle coordinate points were set, with the coordinates being (1.1, 1.2), (3, 2.4), (5.5, 5.5), (6, 2), and (8, 8.5). Then, the simulation parameters were set: the improved repulsion potential field parameter gain coefficient value η = 5, the gravitational potential field gain coefficient value ξ = 15, the value of the maximum distance that obstacles affect the mobile robot was 1.5 m, and the iterative step size of the mobile robot was 0.1 m. The simulation results showed that the mobile robot could successfully move from the starting point to the target point and achieve the effect of avoiding obstacles.

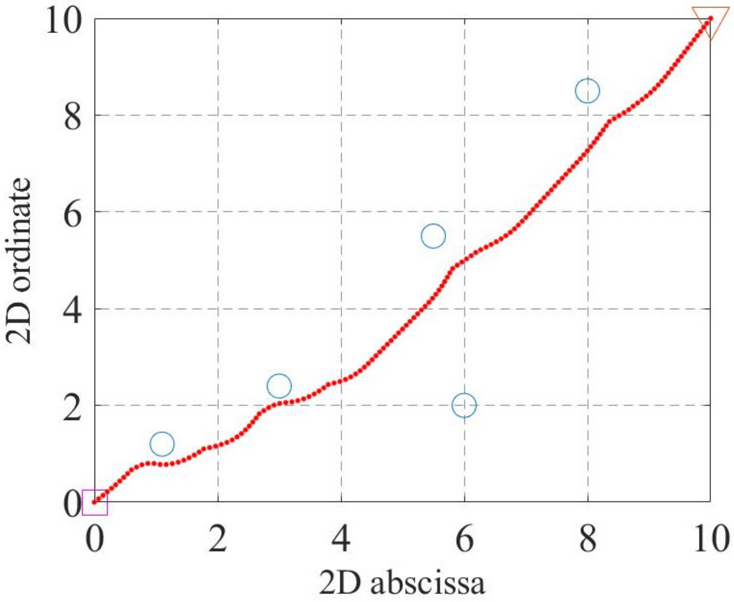


Figure 10. Simulation results of improved artificial potential field method.

To illustrate the influence of the improved artificial potential field algorithm parameter selection on the obstacle avoidance effect, the obstacle avoidance effect of the mobile robot in the moving process is analyzed from the perspective of reducing the repulsion force and the gain coefficient of the gravitational potential field. Figure 11a showed the simulation results when the repulsive potential field gain coefficient was too small. It could be clearly seen that although there would be some collisions with obstacles during the movement, the robot could still move to the target point in the end. Figure 11b shows the simulation results when the gravitational potential field gain coefficient was too small. It can be clearly seen that the mobile robot would oscillate back and forth at certain positions. In addition, the robot cannot reach the set target point.

Figure 12 shows the situation where the obstacle is located on the extension line between the robot and the target point, and the target point is within the repulsive potential field of the obstacle. At this time, the coordinates of the target point are set to (8, 8), the coordinates of the obstacle are set to (9, 9), and the maximum action radius of the repulsive potential field of the obstacle is 1.5 m. The mobile robot reaches the target point smoothly according to the planned path.

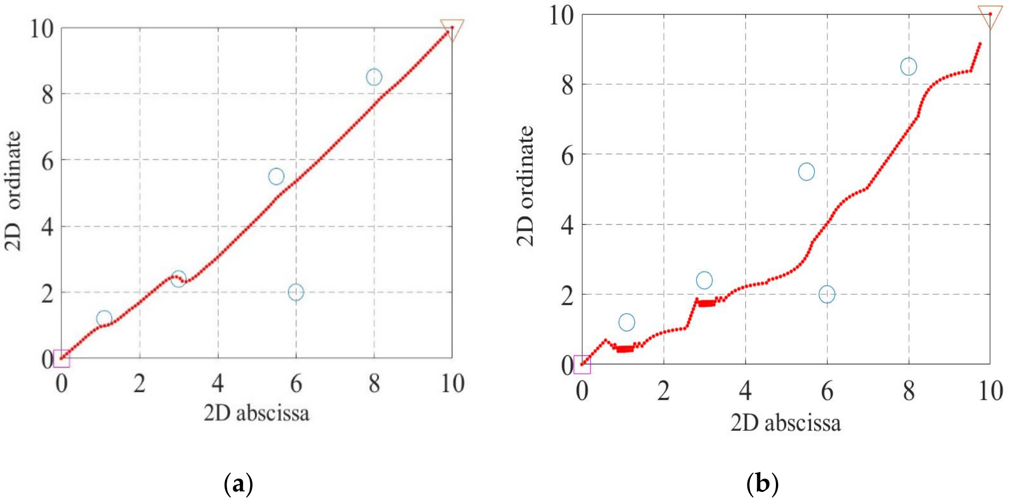


Figure 11. Simulation results when the gain coefficient is not selected: (**a**) The repulsion gain coefficient is too small; (**b**) Gravitational gain coefficient is too small.

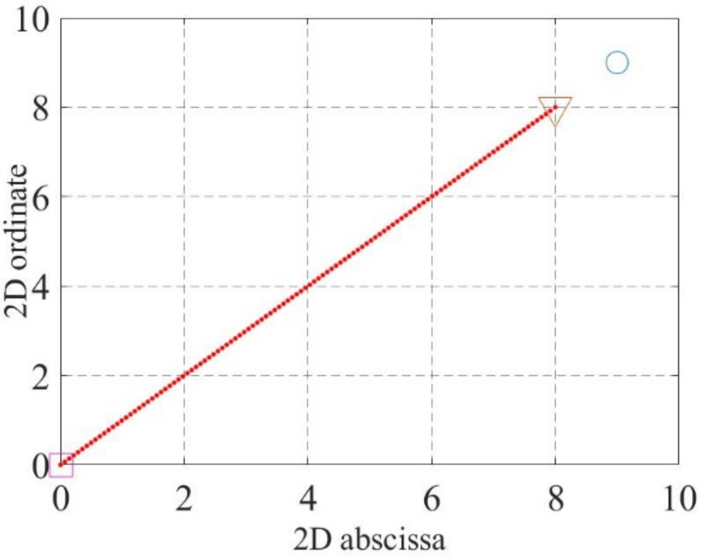
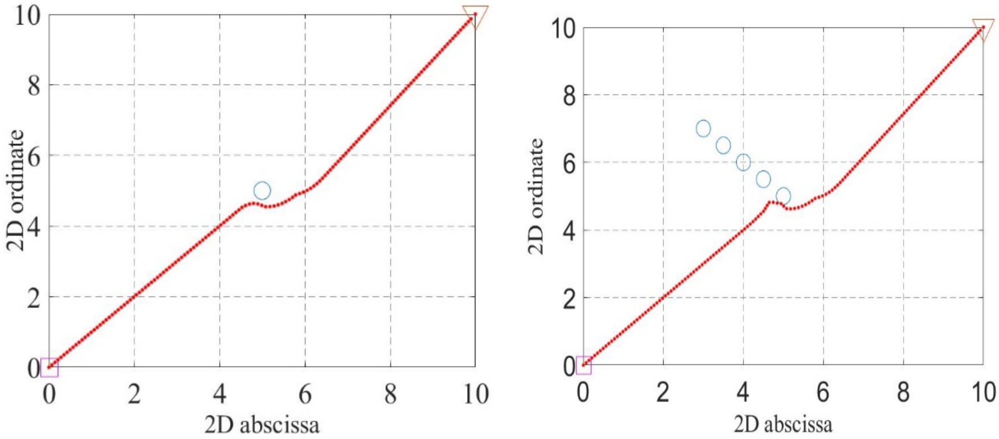


Figure 12. Simulation results when the target point is in the middle.

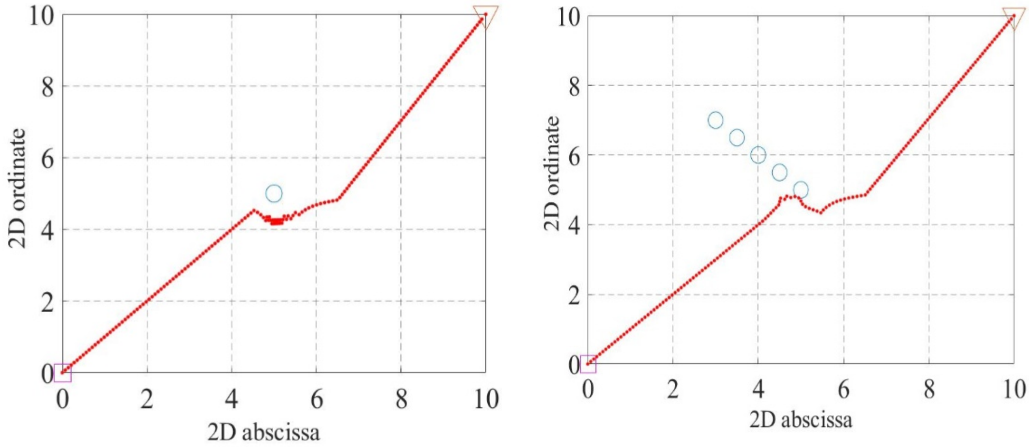
Figure 13 shows the simulation results when the obstacle is set between the starting point and the target point, and the resultant force is on the connection line between the two, in which the position coordinates of the obstacle in Figure 13a are (5, 5); the position coordinates of the obstacles in Figure 13b are (5, 5), (4.5, 5.5), (4, 6), (3.5, 6.5), and (3, 7). It can be seen that the robot can successfully get rid of the minimum point and avoid obstacles when it falls into a local minimum value during the movement process, and finally can move to the target point.

Take single obstacle and multiple obstacles as examples, the simulation results of the traditional artificial potential field method are shown in Figure 14 below. It can be seen that compared with the improved artificial potential field method, the traditional artificial potential field method will fall into oscillation when the robot is close to the obstacles, and requires more steps when it is away from the obstacles, which is not conducive to the rapid obstacle avoidance of the robot.



(**a**) (**b**)

Figure 13. Simulation results when obstacles are in the middle: (**a**) Single obstacle; (**b**) Multiple obstacles.



(**a**) (**b**)

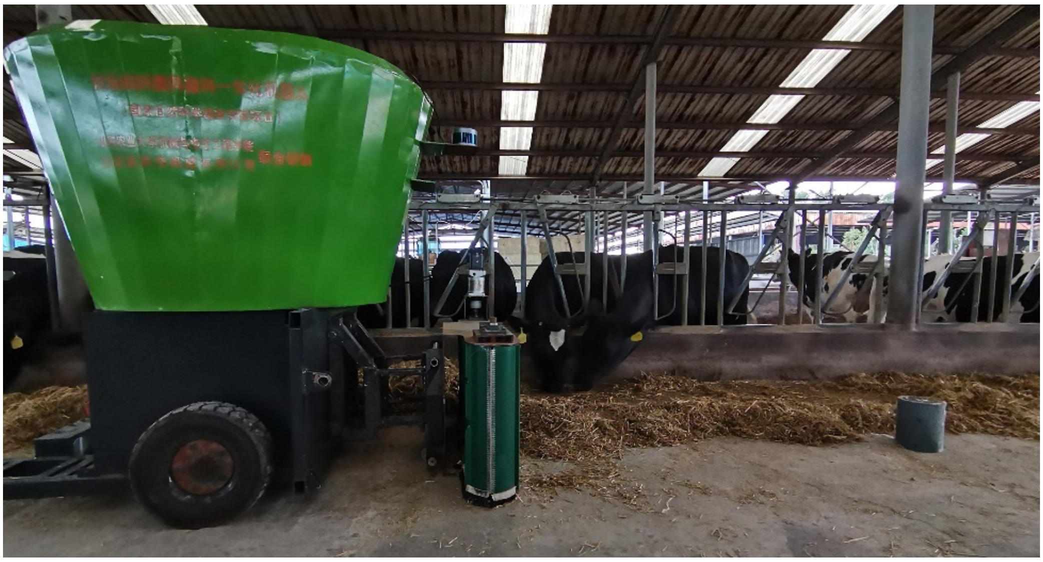
Figure 14. Simulation results under traditional artificial potential field: (**a**) Single obstacle; (**b**) Multiple obstacles.

The above results show that the improved artificial potential field method has better performance under reasonable parameter selection. The forward path is predicted and judged before the robot moves, and simplifies the restricted obstacles; that is, the robot only affects the repulsive force of the obstacles on the target side within the affected range; then, a reasonable virtual target point is set near the simplified obstacle, and the improved repulsion function guides the robot to quickly generate a smooth, stable, and collision-free path in a complex environment. Moreover, the rationality of the selection of the gravitational potential field gain coefficient and the repulsive potential field gain coefficient directly affects the obstacle avoidance effect. The algorithm can realize the obstacle avoidance function of a mobile robot.

3.4. Experimental Research on Obstacle Avoidance

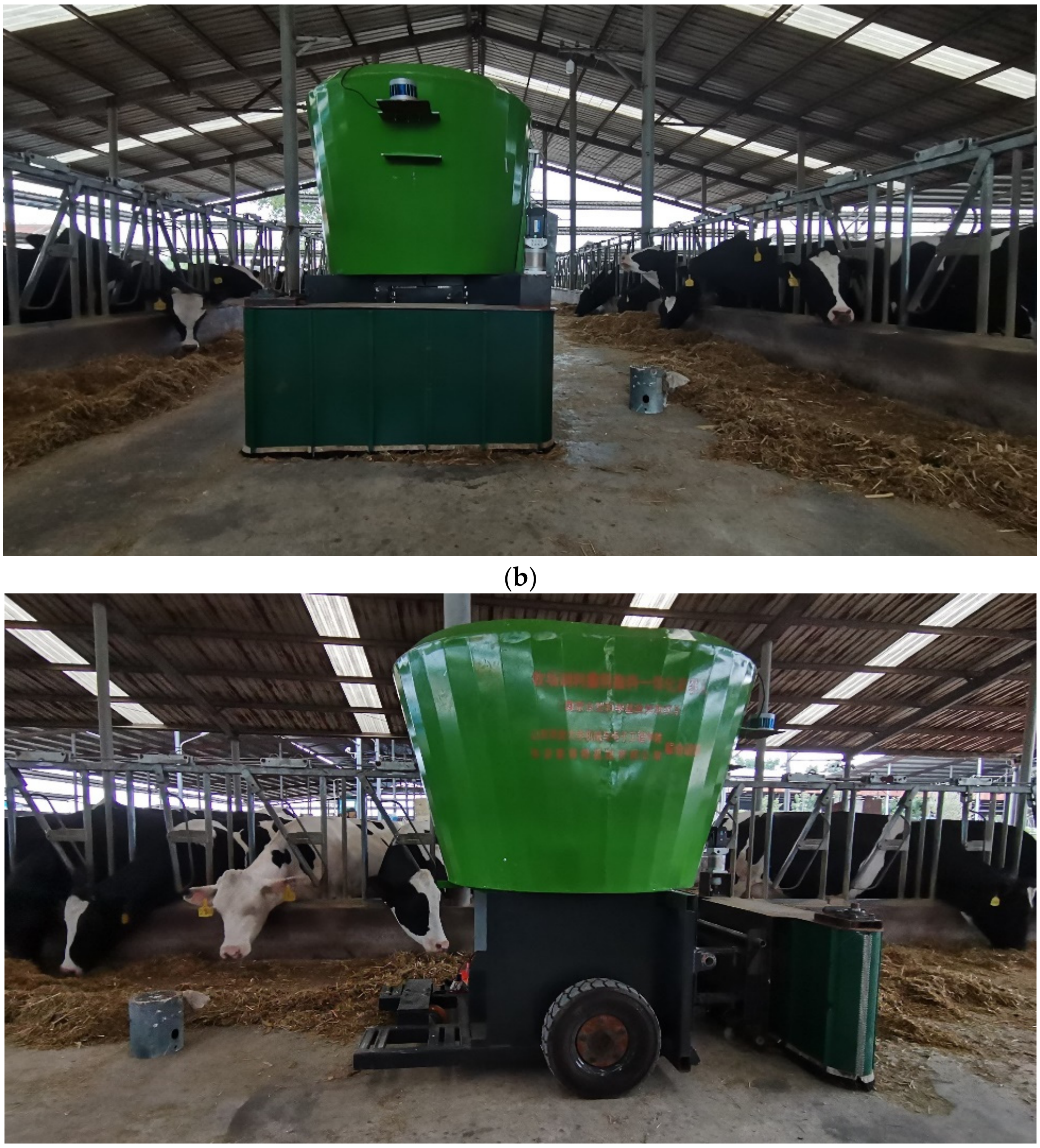
A simple obstacle avoidance test is carried out on the designed pusher robot in this section. The static obstacle is set up in the experiment. In the three scenarios, the maximum speed of the inspection robot is set to 0.5 m/s. Due to the low vehicle speed, the influence radius of obstacles is set to 1.5 m; objects detected within 1.5 m in front of the robot are regarded as obstacles. In the obstacle avoidance experiment, after placing the static obstacle objects in the test area, the robot is initialized to collect and model the surrounding environment information. The robot is instructed to move along the extracted path to verify the robot's response to static obstacles. After the robot completes the map construction and bypasses the obstacles, it quickly moves in the direction of the robot.

The pusher robot performs linear work toward the target until the robot moves to the position shown in Figure 15a. At this time, the robot enters the influence range of the obstacle. Under the combined action of attraction and repulsion, it deflects an angle to the right to drive. Figure 15b shows the position where the robot is closest to the obstacle. Under the action of the resultant force, the robot gradually crosses the obstacle until it successfully reaches the end point (Figure 15c). During the entire driving and obstacle avoidance process, the closest distance to the obstacle is 0.41 m. The optimized path increases the shortest distance value of the obstacle point cloud from the navigation path from 0.18 to 0.41 m, where the average time is 0.059 s and the standard deviation is 0.007 s, which shows that the optimization method can optimize the path in real time to avoid obstacles, basically meeting the requirements of security and real-time performance, and effectively avoiding the local minimum problem The entire obstacle avoidance path is relatively smooth, which can successfully avoid obstacles and reach the destination point. The test proves that the pusher robot can efficiently extract the working path, make timely decisions when detecting static obstacles, avoid collisions with obstacles, and has good stability and reliability.



(**a**)

Figure 15. *Cont.*



(**b**)

(**c**)

Figure 15. Obstacle avoidance scene and process: (**a**) Find the obstacle; (**b**) Get around obstacle; (**c**) Keep working.

4. Conclusions

Based on the research on the existing cowshed fence line extraction method, a robot navigation path optimization method based on the improved artificial potential field method is proposed. This method improves the safety of the unmanned driving of the feeding machine, and provides theoretical support and development basis for the intelligent agricultural equipment in the dairy farm.

(1) The functional requirements of the pushing robot were analyzed and the hardware system was designed. According to the functional requirements of each module, the model was selected and designed, and the research and experimental platform of the pusher robot was built.

(2) The performance of LSM and RANSAC for extracting ridge lines and initial paths was evaluated from the aspects of real-time performance and anti-noise capability. Under three different environments: no noise, Gaussian noise, and artificial noise, the fence lines were extracted using RANSAC, and the obtained results were about −0.058, 0.058, and −0.061, respectively. Compared with LSM, RANSAC was less affected by

noise in processing time (between 1.157 × 10^{-3} and 1.165 × 10^{-3} s). It was shown that the running time, anti-noise ability, and extraction effect of RANSAC were better.

(3) The obstacle avoidance method of the pusher robot was optimized. The target point influence factor was introduced into the repulsive potential field function, and the parameter selection strategy of the improved artificial potential field method was analyzed. Finally, the correctness and feasibility of the obstacle avoidance method were verified by simulation experiments.

Author Contributions: Conceptualization, Z.Y. and F.T.; methodology, F.L. and Z.W.; validation, Y.Y.; formal analysis, Z.S.; investigation, R.W.; resources, S.Y.; writing—original draft preparation, X.W.; writing—review and editing, Z.Y. and F.T.; project administration, F.T.; funding acquisition, Z.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded partly supported by National Natural Science Foundation of China, grant number 32072786, and the Shandong Agricultural University Research Foundation, grant number 030-72195.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are included in the article.

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

References

1. Dong, C.Y.; Ma, B.B.; Lu, C.X. Spatio-temporal dynamics of feed grain demand of dairy cows in China. *Sustainability* **2020**, *12*, 663. [CrossRef]
2. Lovarelli, D.; Brandolese, C.; Leliveld, L.; Finzi, A.; Riva, E.; Grotto, M.; Provolo, G. Development of a new wearable 3D sensor node and innovative open classification system for dairy cows' behavior. *Animals* **2022**, *12*, 1447. [CrossRef] [PubMed]
3. Tan, M.; Hou, Y.; Zhang, L.; Shi, S.; Long, W.; Ma, Y.; Zhang, T.; Li, F.; Oenema, O. Operational costs and neglect of end-users are the main barriers to improving manure treatment in intensive livestock farms. *J. Clean. Prod.* **2021**, *289*, 125149. [CrossRef]
4. Zhang, M.; Wang, X.; Feng, H.; Huang, Q.; Xiao, X.; Zhang, X. Wearable Internet of Things enabled precision livestock farming in smart farms: A review of technical solutions for precise perception, biocompatibility, and sustainability monitoring. *J. Clean. Prod.* **2021**, *312*, 127712. [CrossRef]
5. da Borso, F.; Kic, P.; Kante, J. Analysis of management, labor and economics of milking systems in intensive goat farms. *Agriculture* **2022**, *12*, 513. [CrossRef]
6. Escalante, H.; Castro, L.; Amaya, M.P.; Jaimes, L.; Jaimes-Estevez, J. Anaerobic digestion of cheese whey: Energetic and nutritional potential for the dairy sector in developing countries. *Waste Manag.* **2018**, *71*, 711–718. [CrossRef] [PubMed]
7. Wu, X.; Lu, Y.; Xu, H.; Lv, M.; Hu, D.; He, Z.; Liu, L.; Wang, Z.; Feng, Y. Challenges to improve the safety of dairy products in China. *Trends Food Sci. Technol.* **2018**, *76*, 6–14. [CrossRef]
8. Maitiniyazi, S.; Canavari, M. Exploring Chinese consumers' attitudes toward traceable dairy products: A focus group study. *J. Dairy Sci.* **2020**, *103*, 11257–11267. [CrossRef]
9. Yang, S.; Liang, S.; Zheng, Y.; Tan, Y.; Xiao, Z.; Li, B.; Liu, X. Integrated navigation models of a mobile fodder-pushing robot based on a standardized cow husbandry environment. *Trans. ASABE* **2020**, *63*, 221–230. [CrossRef]
10. Yu, Z.; Liu, Y.; Yu, S.; Wang, R.; Song, Z.; Yan, Y.; Li, F.; Wang, Z.; Tian, F. Automatic Detection Method of Dairy Cow Feeding Behaviour Based on YOLO Improved Model and Edge Computing. *Sensors* **2022**, *22*, 3271. [CrossRef]
11. Rahim, T.; Usman, M.A.; Shin, S.Y. A survey on contemporary computer-aided tumor, polyp, and ulcer detection methods in wireless capsule endoscopy imaging. *Comput. Med. Imaging Graph.* **2020**, *85*, 101767. [CrossRef]
12. Hassan, S.A.; Rahim, T.; Shin, S.Y. Real-time uav detection based on deep learning network. In Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, 16–18 October 2019.
13. Shen, D.; Wu, G.; Suk, H.I. Deep Learning in medical image analysis. *Annu. Rev. Biomed. Eng.* **2017**, *19*, 221–248. [CrossRef]
14. Wu, X.; Sahoo, D.; Hoi, S.C.H. Recent advances in deep learning for object detection. *Neurocomputing* **2020**, *396*, 39–64. [CrossRef]
15. Luong, M.; Pham, C. Incremental learning for autonomous navigation of mobile robots based on deep reinforcement learning. *J. Intell. Robot. Syst.* **2020**, *101*, 1–11. [CrossRef]
16. Su, D.; Kong, H.; Qiao, Y.; Sukkarieh, S. Data augmentation for deep learning based semantic segmentation and crop-weed classification in agricultural robotics. *Comput. Electron. Agric.* **2021**, *190*, 106418. [CrossRef]

17. Seo, D.; Cho, B.-H.; Kim, K.-C. Development of monitoring robot system for tomato fruits in hydroponic greenhouses. *Agronomy* **2021**, *11*, 2211. [CrossRef]
18. Zheng, Y.Y.; Kong, J.L.; Jin, X.B.; Wang, X.Y.; Zuo, M. CropDeep: The crop vision dataset for deep-learning-based classification and detection in precision agriculture. *Sensors* **2019**, *19*, 1058. [CrossRef] [PubMed]
19. Thombre, S.; Zhao, Z.; Ramm-Schmidt, H.; Vallet Garcia, J.M.; Malkamaki, T.; Nikolskiy, S.; Hammarberg, T.; Nuortie, H.; Bhuiyan, M.Z.; Sarkka, S.; et al. Sensors and AI techniques for situational awareness in autonomous ships: A review. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 64–83. [CrossRef]
20. Chen, M.; Tang, Y.; Zou, X.; Huang, Z.; Zhou, H.; Chen, S. 3D global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and SLAM. *Comput. Electron. Agric.* **2021**, *187*, 106237. [CrossRef]
21. Yang, H.; Guo, H.; Dai, W.; Nie, B.; Qiao, B.; Zhu, L. Bathymetric mapping and estimation of water storage in a shallow lake using a remote sensing inversion method based on machine learning. *Int. J. Digit. Earth* **2022**, *15*, 789–812. [CrossRef]
22. Pavkin, D.Y.; Shilin, D.V.; Nikitin, E.A.; Kiryushin, I.A. Designing and simulating the control process of a feed pusher robot used on a dairy farm. *Appl. Sci.* **2021**, *11*, 10665. [CrossRef]
23. Uddin, K.; Jeong, T.H.; Oh, B.T. Incomplete region estimation and restoration of 3D point cloud human face datasets. *Sensors* **2022**, *22*, 723. [CrossRef] [PubMed]
24. Lee, M.-Y.; Lee, S.-H.; Jung, K.-D.; Lee, S.-H.; Kwon, S.-C. A novel preprocessing method for dynamic point-cloud compression. *Appl. Sci.* **2021**, *11*, 5941. [CrossRef]
25. Liu, W.; Chen, R.; Cai, H.; Luo, W. Robust statistical methods for impulse noise suppressing of spread spectrum induced polarization data, with application to a mine site, Gansu province, China. *J. Appl. Geophys.* **2016**, *135*, 397–407. [CrossRef]
26. Zermas, D.; Morellas, V.; Mulla, D.; Papanikolopoulos, N. 3D model processing for high throughput phenotype extraction—The case of corn. *Comput. Electron. Agric.* **2020**, *172*, 105047. [CrossRef]
27. Dal Poz, A.P.; Yano Ywata, M.S. Adaptive random sample consensus approach for segmentation of building roof in airborne laser scanning point cloud. *Int. J. Remote Sens.* **2019**, *41*, 2047–2061. [CrossRef]
28. Liu, M.; Wang, X.; Li, L.; Gao, E. Improved random sampling consistency algorithm employed in three-dimensional point cloud registration. *Laser Optoelectron. Prog.* **2018**, *55*, 165–171.
29. Xia, H.; Yang, B.; Li, Y.; Wang, B. An improved CenterNet model for insulator defect detection using aerial imagery. *Sensors* **2022**, *22*, 2850. [CrossRef]
30. Chen, Y.; Feng, K.; Lu, J.; Hu, Z. Machine vision on the positioning accuracy evaluation of poultry viscera in the automatic evisceration robot system. *Int. J. Food Prop.* **2021**, *24*, 933–943. [CrossRef]
31. Yang, W.; Wu, P.; Zhou, X.; Lv, H.; Liu, X.; Zhang, G.; Hou, Z.; Wang, W. Improved artificial potential field and dynamic window method for amphibious robot fish path planning. *Appl. Sci.* **2021**, *11*, 2114. [CrossRef]
32. Li, L.; Wu, D.; Huang, Y.; Yuan, Z.-M. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Appl. Ocean. Res.* **2021**, *113*, 102759. [CrossRef]
33. Sudhakara, P.; Ganapathy, V.; Priyadharshini, B.; Sundaran, K. Obstacle avoidance and navigation planning of a wheeled mobile robot using amended artificial potential field method. *Procedia Comput. Sci.* **2018**, *133*, 998–1004. [CrossRef]
34. Zhao, T.; Li, H.; Dian, S. Multi-robot path planning based on improved artificial potential field and fuzzy inference system. *J. Intell. Fuzzy Syst.* **2020**, *39*, 7621–7637. [CrossRef]
35. Hou, P.; Pan, H.; Guo, C. Simulation research for mobile robot path planning based on improved artificial potential field method recommended by the AsiaSim. *Int. J. Model. Simul. Sci. Comput.* **2017**, *8*, 1750046. [CrossRef]