# APPLICATION OF 3D LIDAR-BASED NAVIGATION PATH DETECTION AND OBSTACLE AVOIDANCE IN POULTRY HOUSES
# /
# *基于三维激光雷达的导航路径检测与避障技术在家禽舍中的应用*

**Kai WANG** [1)], **Khurram YOUSAF** [2)], **Jian SONG** [1)], **Yang BAI** [1)], **Fuxiang XIE** [*1)], **Zhenwei YU** [**3)]
[1)]School of Machinery and Automation, Weifang University, Weifang 261061, China
[2)] Atta-ur-Rahman School of Applied Biosciences, National University of Sciences and Technology, Islamabad 44000, Pakistan
[3)] College of Mechanical and Electronic Engineering, Shandong Agricultural University, Tai'an 271018, China
**Correspondence should be addressed to Fuxiang Xie* xfx608@126.com
***Correspondence should be addressed to Zhenwei Yu;*
zhenweiyu615@126.com *DOI: https://doi.org/10.35633/inmateh-77-86*

**Keywords:** *Caged poultry house; Inspection robot; Obstacle avoidance; Path extraction*

## ABSTRACT

*In this study, an autonomous navigation robot for poultry house inspection was designed, and a path optimization and obstacle avoidance strategy was proposed. First, a filtering algorithm was used to extract regions of interest from the 3D point cloud data collected by the inspection robot in caged poultry houses. Then, the geometric structure of cage-row lines was estimated using the least-squares method and refined using the RANSAC algorithm. The refined lines were projected to obtain boundary contour features. Finally, the A* algorithm was improved by removing redundant nodes, reducing the number of turning points, shortening the total path length, and increasing the weight of the cost estimation. The improved A* algorithm was also validated through physical robot simulation tests. Experimental results showed that compared with the least-squares method (LSM), the RANSAC-based approach achieved cage-row line slope values of 0.223 and 0.224 under Gaussian noise and manually added noise, respectively, demonstrating superior noise robustness and real-time performance. The results further indicate that the improved A* algorithm enhances path planning efficiency, enabling the robot to make timely decisions when encountering static or dynamic obstacles, thereby improving overall stability and reliability.*

## 摘要

*本研究设计了一种用于巡检鸡舍的自主导航机器人，并提出了一种路径优化与避障策略。首先，利用滤波算法提取巡检机器人在笼养鸡舍采集的三维点云数据中的感兴趣区域；其次，利用最小二乘估计鸡舍线几何形状，并利用随机抽样采样算法（RANSAC）进行细化，对细化后的线进行投影，得到边界轮廓特征；最后，对A*算法进行改进，去除冗余节点，减少路径转折点数量，缩短路径长度，并增加代价估计的权重。同时，对改进的A*算法进行了实车仿真测试。实验结果表明，与最小二乘法（LSM）相比，RANSAC算法提取的鸡舍线在高斯噪声和人工噪声下的斜率分别为0.223和0.224，RANSAC算法在抗噪能力和实时性方面均有更好的表现。结果表明，改进的A*算法可以提高路径规划的效率，机器人在检测到动态和静态障碍物时能够及时做出决策，具有更好的稳定性和可靠性。*

## INTRODUCTION

As a major country in poultry farming, China ranks first in the world in terms of annual egg production *(Yang, 2021)*. In 2020, China's total animal husbandry output value reached 3.465 trillion yuan, accounting for 26.6% of the total agricultural output value, and it has become a pillar industry in the farm field. However, due to the impact of poultry epidemics, problems such as low production capacity, high fatality rate, and slow growth rate have been restricting the steady growth of the poultry farming business *(Teng et al., 2018; Zhuang et al., 2018)*. Recently, with the advancement of intensive and large-scale breeding models, the restricting effect of epidemic factors on the modernization process of animal husbandry has become increasingly prominent and has gradually risen to the level of food safety, human health, and ecological environment. Without timely intervention, the impact of hazards cannot be predicted. At the same time, in the modern poultry breeding model, the single house has a huge amount of feeding, and the distribution of multi-column and multi-layer chicken cages makes the traditional manual inspection time-consuming and labor-intensive *(Zhuang & Zhang, 2019)*. The development and application of intelligent inspection robots are precisely effective in solving the current breeding dilemma of caged poultry houses. It not only liberates manpower but also reduces the contact between sick and dead birds and people and cuts off the transmission route of the disease to the greatest extent.

The development of digital image processing and machine learning has enabled intelligent poultry production robots to have more mature technical support and greater market demand in the field of poultry farming. Farm Robotics and Automation has designed a suspended inspection robot that monitors the health of poultry, locates dead poultry, detects litter problems, and measures surrounding environmental conditions through tracks and sensors installed above the ceiling of the poultry house *(Ren et al., 2020)*. Azamat Yeshmukhametov and his team proposed a new methodology for concurrently estimating the agent's spatial position and mapping the adjacent environment. The following approach enables robots to adjust to different lighting and weather environments, ensuring stable operation in several agricultural environments *(Yeshmukhametov, 2024)*. *Han et al. (2023)* employed an improved clustering algorithm to process LiDAR data and combined the convex hull and rotating caliper algorithms to extract obstacle features. At the visual level, they utilized the optimal H component of the HSV color space, together with filtering and erosion operations, to generate guidance trajectories. This approach achieved efficient and robust real-time obstacle avoidance and path control in complex orchard environments, and its effectiveness was verified through experiments. A self-sufficient robot, PoultryBot, has been integrated for operation in poultry houses. It can traverse the house while tracking its pose, sensing obstacles, and retrieving floor eggs *(Vroegindeweij et al., 2018)*. Chang developed a vision-driven mobile poultry robot with a newly engineered mechanism. At a viewing range of 0.75 m, it detects eggs with up to 97.6% accuracy, and the vision plus behavior-logic controller enables reliable obstacle avoidance *(Chang et al., 2020)*.

The above-mentioned poultry intelligent production robots take path extraction, path planning, and autonomous navigation as the breakthrough point of research and development, which effectively decreases the labor intensity of managers and improves operation efficiency and quality.

In the agricultural production process, the production robot equipped with automatic navigation technology uses sensors to realize the perception of dynamic environmental information according to the known operation information and environmental information, to carry out path planning and travel control. Commonly used autonomous navigation methods include the Global Navigation Satellite System *(Bijjahalli & Sabatini, 2021)*, Inner Measurement Unit *(Guang et al., 2018)*, Visual sensor *(Santoso et al., 2017)*, Light Detection and Ranging (LiDAR), etc. They are usually used alone or in combination. For caged poultry houses, due to the occlusion of satellite signals by the building of the poultry house, the use of Global Navigation Satellite System technology has great limitations; long-term use of the Inner Measurement Unit will cause errors to accumulate; the Visual sensor is greatly affected by aspects such as light changes and shadows. In contrast, because it delivers accurate distances, fine granularity, and robust performance under interference, LiDAR has risen to prominence in agricultural robotics and is commonly used to perceive and extract indoor environmental information. As per the integrity of the obtained environmental data, navigation path extraction and planning can be segregated into local and global path planning. The global path planning algorithm requires overall information about the poultry house, such as the shape and size of the poultry house. Local path planning only needs the environmental information near the inspection robot to complete the path planning. Therefore, there are two steps to complete the local path planning: first, obtain the surrounding environment information through the sensor; then, analyze and use the obtained information through the optimization algorithm to plan a reasonable and safe driving path. Commonly used algorithms include A* algorithm, the Artificial Potential Field method, the Particle Swarm algorithm, etc.

Although path extraction methods have been reported for various agricultural environments, caged poultry houses remain underexplored. To address this gap, a machine vision system was developed and integrated with a custom inspection robot equipped with a 3D LiDAR sensor to acquire house-scale point cloud data. After ground points were removed and regions of interest were isolated via pass-through filtering, cage-row lines were fitted using a least-squares RANSAC approach, projected to extract boundary-contour features, and subsequently utilized to reconstruct fence lines and generate initial navigation routes. Furthermore, a navigation-path refinement and obstacle-avoidance strategy based on an improved A* algorithm was implemented. The resulting system is capable of autonomously producing accurate navigation paths in dynamic indoor poultry-farming environments.

## MATERIALS AND METHODS

### *2.1. Design of the inspection robot*

### *2.1.1 Overall Design:*

The inspection robot has a length of 0.716 m, a width of 0.600 m, a height of 0.355 m, a rated load of 50kg, and a maximum travel speed of 2 m/s. Figure 1 shows the 3D schematic installation diagram of each module inside the inspection robot and the finished product display.

The inspection robot integrates a vehicle-navigation hardware suite, a control module, and a poultry-behavior monitoring unit - each essential to overall performance. The navigation hardware and abnormal-behavior monitoring subsystems act as execution units: they receive commands from the control system and carry out poultry behavior data collection, analysis, and epidemic alerts. The control system configures vehicle operating modes, assigns waypoints, shows the robot's position in real time, and supervises the abnormal-behavior module. By fusing multi-sensor data, the robot performs self-localization and navigation in natural settings; with adaptive fusion, it conducts inspections autonomously without external infrastructure.

### *2.1.2 Hardware system design:*

Real-time communication between the hardware device and the control system is built with the help of the wireless network, and the related tasks of autonomous navigation and inspection of the inspection robot are completed collaboratively.

Figure 1 shows the vehicle's navigation stack: robot chassis, drive module, control electronics, environmental perception unit, communications, and power supply.

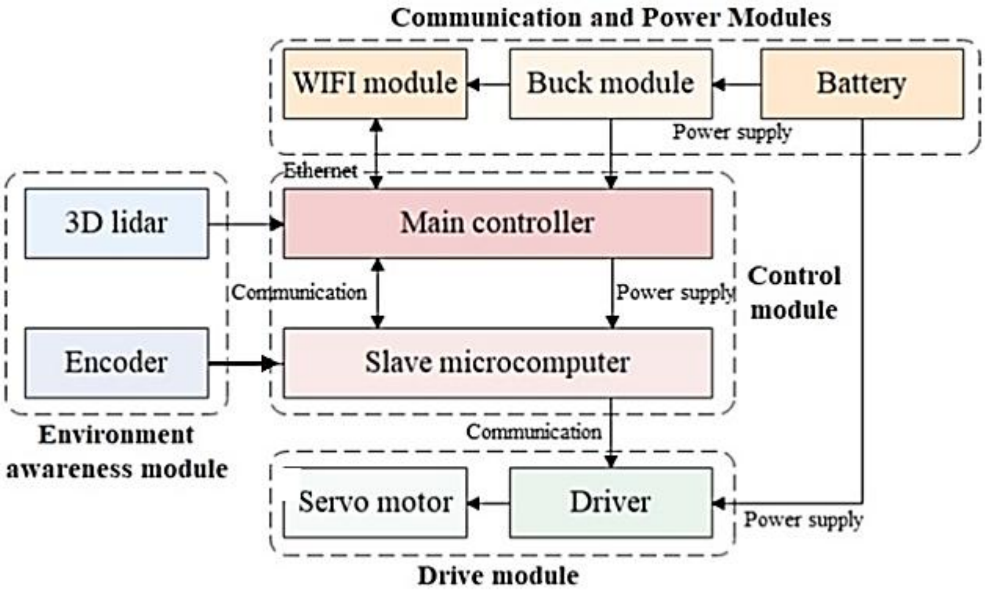


**Fig. 1 – The hardware system of the vehicle navigation**

As per inspection tasks and the farm's environmental requirements, the inspection robot should have the advantages of good terrain adaptability, obstacle-surmounting ability, and flexible steering. The drive system is a dual-wheel differential configuration. Independent motors control the drive wheels on the same server. A reducer is connected between the drive wheels and the servo motor, which can help to maximize the speed up to 2 m/s. When working, the underlying servo controller changes the motion state of the transport robot by adjusting the speed of the servo motor. Steering is achieved by commanding different velocities to the two drive wheels; the rear axle is a differential drive, and the front axle carries two passive wheels. The drive system features a simple structure, a small turning radius, and flexible movement, which enhances the control accuracy of the entire machine. A 48V lithium battery is used to power the robot. The voltage of the power module is 220 V. To provide strong onboard computing, the main controller is an NVIDIA Jetson Nano, built on a Tegra X1 heterogeneous SoC, and uses ROS as the basic navigation framework under the Ubuntu 18.04 system to communicate with the chassis through RS-485. The underlying controller is equipped with the STM32F103 embedded motherboard. Wheel-encoder speeds feed a kinematic model to compute odometry—linear velocity, travel distance, and yaw change. The control algorithm then generates and evaluates commands for the onboard system and the push mechanism.

The perception module employs a 16-channel compact LiDAR (RS-LiDAR-16, Sagitar Juchuang, China). Its rotor carries 16 laser/receiver pairs that emit high-rate pulses while spinning, streaming real-time 3D point clouds. The ranging pipeline outputs XYZ points and intensity, enabling a digital model to support obstacle avoidance, localization, navigation. As shown in Fig. 2(b), a local frame is set with origin OOO at the LiDAR center, +Y forward, +X to the robot's left, and +Z by the right-hand rule. The sensor is installed at the front-center of the chassis, 0.6 m above the floor, with specifications listed 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$^{-1}$) | 6000000 |
| Frame rate / Hz | 10 |
| Rotation speed / rpm | 600 |

### *2.2 Preprocessing and collection of point cloud data*

### *2.2.1 Point cloud data collection:*

The data of the Poultry-house were gathered on Nov 2, 2021, at the Animal Husbandry Science and Technology Experiment Station, Shandong Agricultural University, Tai'an, Shandong. The internal environment of the poultry building is shown in Figure 2(b). The four-layer H-shaped chicken cage is used for feeding, and the distance (D) between the two rows of chicken cages is 1.2 m. The birdhouse point cloud data collected by LIDAR can be seen in Figure 2(a). The following figure shows that the number of 3D point clouds is huge and contains more cluttered points. The study clips the cloud to an axis-aligned box centered at the origin: X−2.5–2.5 m, Y−1.5–1.5 m, Z −0.5–0.5 m, which serves as the ROI. There are multiple noise points in the three-dimensional point cloud on the ground of the poultry house; it is important to preprocess the original point cloud data to reduce the influence of the ground point cloud in the initial path extraction.

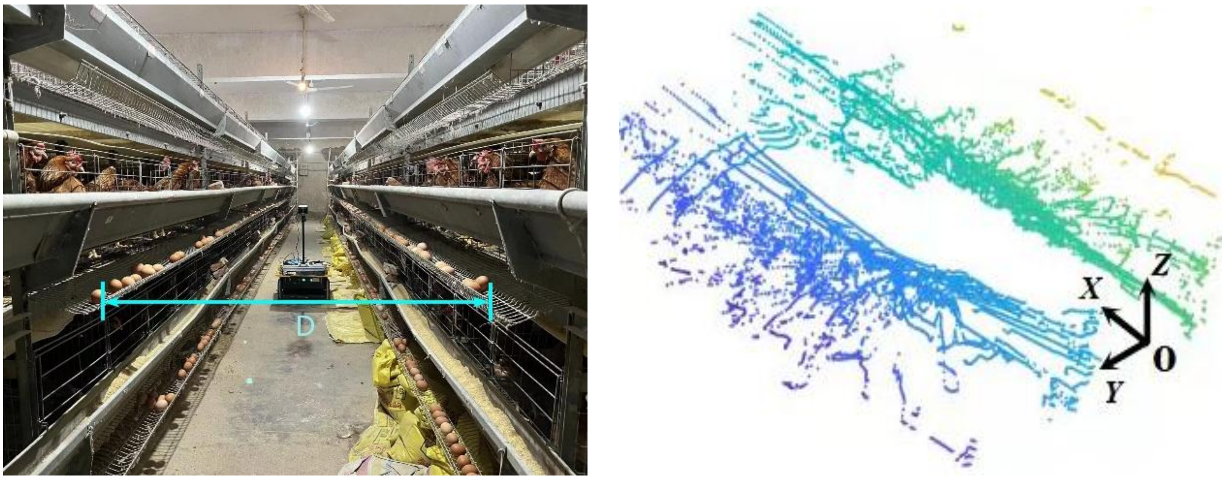


**Fig. 2 -Test environment and original data**
**(a) Experimental environment (b) 3D point cloud raw data**

### *2.2.2 Point cloud data preprocessing:*

The storage format of the poultry house data collected by the LIDAR is a PCD data file. Each point in the file corresponds to the three-axis coordinates. The amount of point cloud data to be processed is substantial. To curb computation and latency, the 3D cloud is first preprocessed. The PCD stream is split into per-frame data, after which a pass-through filter extracts the ROI. A voxel-grid downsampler represents all points within each 0.1 m cube by its centroid, greatly shrinking point count while retaining structure. Residual noise and outliers are pruned via statistical filtering. Finally, to prevent floor returns from corrupting the cage-row-outline extraction, ground and non-ground points are separated using Ground Plane Fitting (GPF) following Zermas' research (*Zermas et al., 2020*).

### *2.3 Ridge line and initial path extraction*

The height threshold was set to 0.1 m according to the height of the chicken cage-row, and combined with experience. At this time, the segmented chicken cage has a better effect. Since the chicken cages are regularly arranged vertically, the point cloud has obvious column characteristics. To extract the arrangement line of the chicken cage, the point cloud of the chicken cage is first projected onto the two-dimensional plane, and then uses the LSM and random sampling consistency method to conduct a controlled experiment to select the optimal extraction method.

The initial path of the robot movement can be calculated from the outline of the chicken cage-row boundary, which is extracted by projecting the chicken cage-row point cloud onto two-dimensional plane. Extracting the initial motion path of the robot by this method can reduce the probability of the mobile robot colliding with the chicken cage-row during operation.

### *2.3.1 The Least Square Method:*

Least squares is a fundamental method widely used in data processing for system identification, error estimation, and prediction *(Hwang & Cho, 2020; Yu & Zhou, 2016)*. It estimates model parameters for point cloud data by minimizing the sum of squared residuals *(Birdal et al., 2020)*. In this study, the cage-row boundary line is obtained by applying least-squares fitting to the point sets located on both sides of the robot's heading direction. The line parameters W are determined by minimizing the objective function F(W), as defined in Eq.(1).

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

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

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

### *2.3.2 Consistency of random sampling:*

The RANSAC estimates model parameters from outlier-contaminated data by repeatedly sampling minimal sets, fitting a hypothesis, and selecting the model with the largest inlier consensus, thereby suppressing outlier influence. *(Ghahremani et al., 2021; He et al., 2022; Lim et al., 2022)*.

RANSAC was applied to extract the cage-row boundary lines on each side. Random subsets of points from the left and right point clouds were repeatedly sampled to generate candidate line models, and the number of inliers was used to evaluate each hypothesis. Through iterative refinement, the optimal linear model was selected as the estimated cage-row boundary line. The cage-row line extraction process is illustrated in Fig.3.

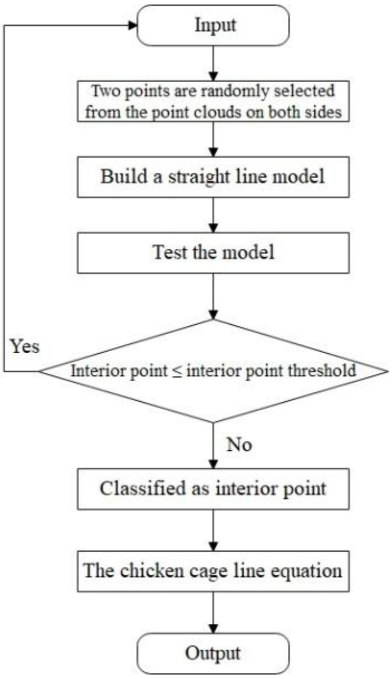

Input
Two points are randomly selected from the point clouds on both sides
Build a straight line model
Test the model
Yes
Interior point ≤ interior point threshold
No
Classified as interior point
The chicken cage line equation
Output

**Fig. 3 - Process of extracting the cage-row boundary line using the RANSAC method**

The iteration threshold $K_{\text{RANSAC}}$ is an important factor for fitting. The low and high values of $K_{\text{RANSAC}}$ will affect the fitting. The selection basis of the *K* value is given in Equation 3:

$$K_{RANSAC} = \lg(1-\alpha)/\lg(1-\omega^{N}) \tag{3}$$

where $\alpha$ is the probability that a sampled subset contains only inliers; $\omega$ is the inlier rate (chance a random point is an inlier); $N$ is the dataset size.

The initial path extracted in this paper is the average value of the left and right chicken cage-row lines. The initial path is defined as shown in Equation 4:

$$y = k_o x + b_o \tag{4}$$

where $k_o = (k_l + k_r)/2$ is the slope of the initial path; $b_o = (b_l + b_r)/2$ is its intercept; $k_l$ and $k_r$ denote the slopes of the cage-row boundary lines on the left and right sides, respectively, $b_l$ and $b_r$ represent the corresponding intercepts.

### 2.4 Initial Path Extraction

#### 2.4.1 Processing of noise:

Processing time for extracting the initial path shows the real-time performance. In the following research, the effects of the initial paths extracted by the LSM and RANSAC methods were evaluated through the two indicators of real-time performance and anti-noise ability, to select the method that can obtain the initial paths with good real-time performance and strong anti-noise ability. In the experiment, Gaussian noise and artificial noise were added to the extracted chicken cage-row point cloud, respectively, and compared with the group without noise to compare the extraction effect. For the Gaussian case, zero-mean noise with variance 0.1 is injected into the ridge point cloud (statistical filtering disabled). Artificial noise is synthesized by inserting spurious points between ridge rows and treating them as interference.

#### 2.4.2 Extraction processing:

The boundary contours are extracted using a vector-based point cloud analysis method. First, the cage-row point clouds and their neighboring points are projected onto the XY-plane, and line segments are fitted to these projections. Based on the approach reported in the literature, the cage-row outline point cloud is then extracted and recorded as the point set $U = \{d_0, d_1 \ldots d_{n-1}\}$. Finally, using Eq. (5), each outline point is evaluated with respect to the cage-row boundary line. Points located outside the boundary are removed, while the remaining interior points are divided into a left point set $U_L$ and a right point set $U_R$.

$$\begin{cases} d_{li} = \frac{|k_l \cdot x_i + b_l - y_i|}{\sqrt{{k_l}^2 + {b_l}^2}} \\ d_{ri} = \frac{|k_r \cdot x_i + b_r - y_i|}{\sqrt{{k_r}^2 + {b_r}^2}} \\ U_L = \left\{\begin{matrix} (k_l \cdot x_i + b_l - y_i) \cdot (k_r \cdot x_i + b_r - y_i) < \\ 0 \cap d_{li} - d_{ri} < 0 | d_i(x_i, y_i) \in U \end{matrix}\right\} \\ U_R = \left\{\begin{matrix} (k_l \cdot x_i + b_l - y_i) \cdot (k_r \cdot x_i + b_r - y_i) < \\ 0 \cap d_{li} - d_{ri} > 0 | d_i(x_i, y_i) \in U \end{matrix}\right\} \end{cases} \tag{5}$$

where:

$d_{li}$ and $d_{ri}$ denote the distances from the outline point $d_i(x_i, y_i)$ to the left and right cage-row boundary lines, respectively;

$i$ is the index of the point in the point set, where $i$= 0, 1, 2, …, $n$-1;

$n$ is the total number of points in the point cloud.

### 2.5 Obstacle avoidance strategy improvement

#### 2.5.1 Traditional A* algorithm:

As a common global path planning algorithm, the A* algorithm is generally used in static road networks and very effective direct search method to find the optimal path *(Li et al., 2022; Tang et al., 2021)*. The A* algorithm evaluates nodes based on the assessment function of the heuristic function, and selects the node with the smallest evaluation function for a single expansion *(Jiang & Sun, 2021)*, in the form of:

$$f(n) = g(n) + h(n) \tag{6}$$

where:

$f(n)$ represents the evaluation function of the robot from the starting point to the target point,

$g(n)$ represents the actual cost value from the starting point to the current node,

$h(n)$ is the estimated cost of the current node to the target node, which is an important part of the cost function.

Generally, the Euclidean distance from the current node to the target node is taken, which is defined as follows:

$$h(n) = \sqrt{(x_{goal} - x_n)^2 + (y_{goal} - y_n)^2} \tag{7}$$

where:

$(x_n, y_n)$ is the center coordinate of the current position; $(x_{goal}, y_{goal})$ is the center coordinate of the target point.

### *2.5.2 A* algorithm improvement:*

Although the old A* algorithm can carry out effective global path planning for the target point, it has many problems, such as many expansion nodes and a wide search range, and it has many turning points. This will reduce the control efficiency of the mobile robot and increase the probability of collision. Based on the above considerations, this paper makes the following improvements to the traditional A* algorithm:

(1) Priority to expand the target direction point

The search method of the traditional A* algorithm is to expand the nodes in the four directions adjacent to the current node each time it expands, which will lead to many unnecessary expansion nodes. At the same time, in the subsequent process, redundant extension nodes will be searched repeatedly, which increases the path planning time.

In this study, the orientation of the target point is decided by calculating the variation between the vertical and horizontal coordinates of the target and current node, respectively. Secondly, the obstacles of the neighboring nodes are checked in the orientation of the target point. If at least one of the adjacent nodes in the orientation of the target point is passable, only the adjacent nodes in the orientation of the target point will be expanded. If the adjacent nodes in the orientation of the target point are all obstacles, the four-neighbor expansion is adopted.

(2) Improve the heuristic function

From the working principle of the A* algorithm, it can be found that the search performance is mainly affected by $h(n)$. The improved expression of $f(n)$ is as follows:

$$f(n) = g(n) + \left(1 + \frac{d}{D}\right) h(n) \tag{6}$$

where:

$d$ is regarded as the distance from the current position of the robot to the target point, and $D$ is the distance from the starting position to the target point.

The search speed of the improved heuristic function is increased, and the optimal path is detectable at the same time.

## RESULTS AND DISCUSSION

### *3.1 Preprocessing results and analysis*

In this experiment, the 3D point cloud data of the mobile robot in the stationary state and the forward speed of 0.4 m/s were collected, respectively, to test the adaptability of the point cloud preprocessing method of the poultry house to different motion states of the mobile robot. From the acquired dataset, 100 point-cloud frames were preprocessed. Table 2 reports the resulting point counts and the runtime of each filtering step. It can be seen from the processing results in Table 2 that the average number of point clouds after preprocessing was 5730 and 5613, respectively, in the two states of stationary and 0.4m/s, and the total preprocessing time of each frame was about 0.524s and 0.469s, respectively. The number of point clouds and the total time of the two states did not vary significantly, showing the preprocessing method of the poultry house point cloud selected in this paper has certain applicability to different motion states of the robot.

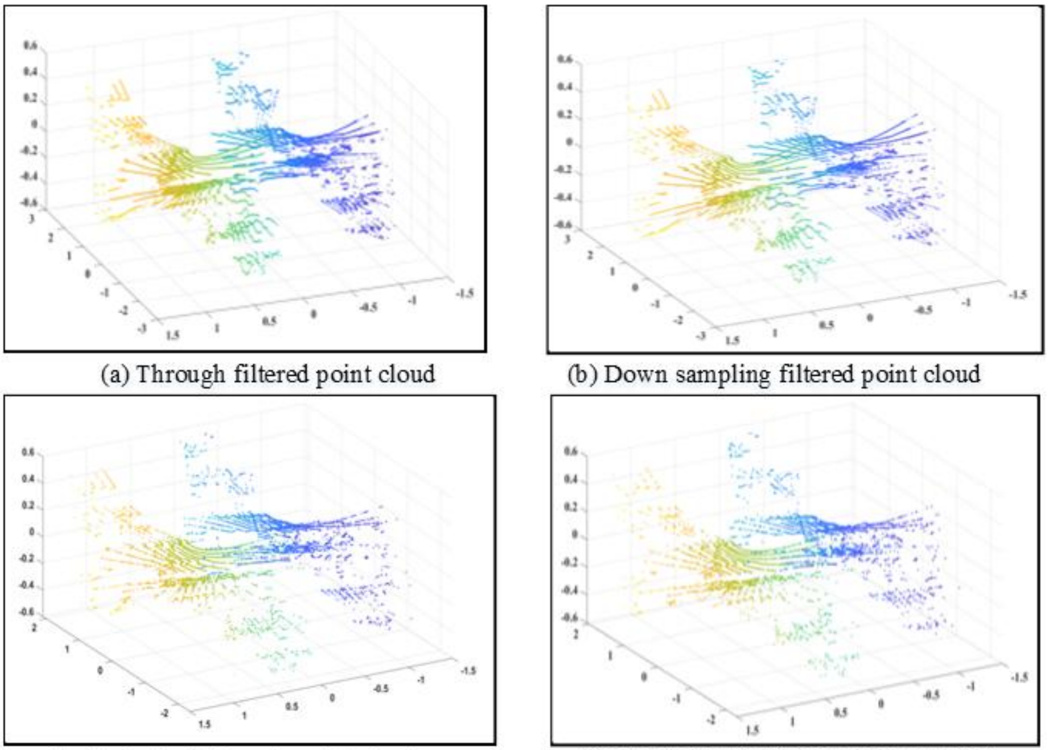
**Table 2**

**Pretreatment results of the chicken farm point cloud**

| States of Robot motion | Preprocessing methods | Number of point clouds | Processing times / s |
|---|---|---|---|
| Static | Through the filter | 16457 | 0.079 |
| | Down-sampled filter | 7658 | 0.054 |
| | Statistical filter | 6994 | 0.307 |
| | Ground plane filter | 5730 | 0.084 |

| States of Robot motion | Preprocessing methods | Number of point clouds | Processing times / s |
|---|---|---|---|
| Motion | Through the filter | 16385 | 0.063 |
| | Down-sampled filter | 7582 | 0.048 |
| | Statistical filter | 6837 | 0.283 |
| | Ground plane filter | 5613 | 0.075 |

The result of the three-dimensional point cloud of the poultry house after preprocessing is illustrated in Figure 4 (a, b, c, and d). The results are the ROI point clouds of the poultry house extracted by the straight-through filtering method. The total number of extracted point clouds was 16,437, and about 41% lower than the number of 3D point clouds in the original poultry house. This method effectively reduces the amount of calculation.



(a) Through filtered point cloud

(b) Down sampling filtered point cloud

(c) Statistical filtered point cloud

(d) Point cloud after ground removal

**Fig. 4 - Schematic diagram of point cloud pretreatment**

### *3.2 Extraction of the initial path*

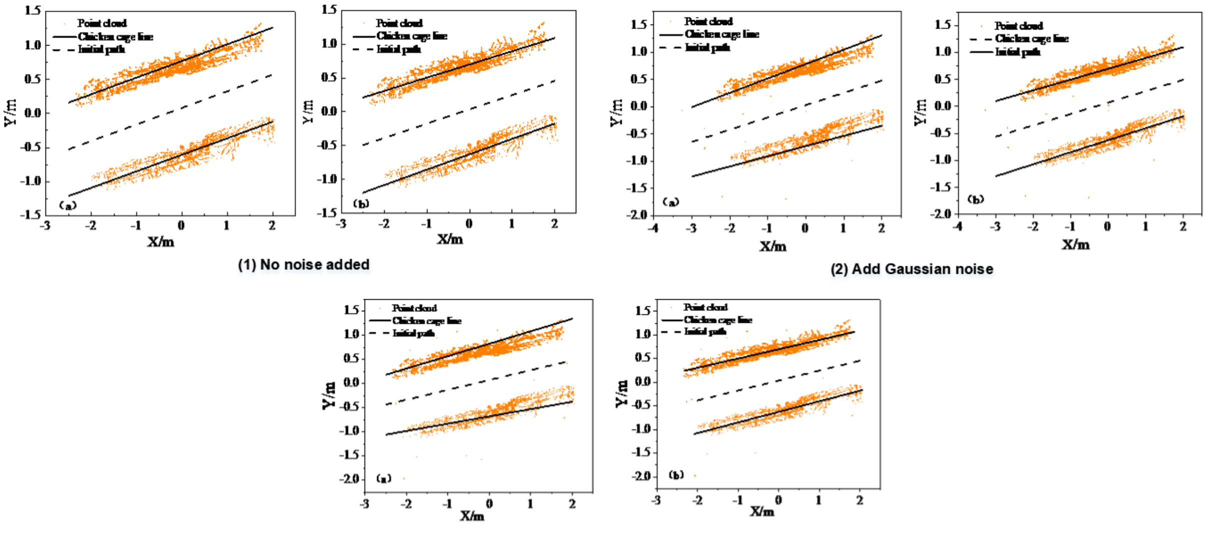
Gaussian and synthetic noise were injected into the poultry-house point clouds, and line/path extraction was evaluated against results from the clean (no-noise) dataset. The visualization results of extracting chicken cage-row lines and initial paths using LSM and RANSAC are explained in Figure 5 and Table 3. It can be observed from Figure 5 that both the LSM and RANSAC methods can process the chicken cage-row point cloud data. It can be seen from Table 3 that after adding artificial noise, the result of LSM extracting chicken cage-row lines (example of the slope of the chicken cage-row line on the right) was about 0.187, which was quite different from the group without noise (slope was about 0.243), and the extraction effect was decreased. LSM fits the line by minimizing orthogonal distances from all points to it. When more data deviates from the chicken cage-row point cloud, the extracted chicken cage-row line will deviate.

Using the right-side cage-row line as an example, RANSAC produced slopes of 0.227 without noise, 0.223 with Gaussian noise, and 0.224 with artificial noise, showing minimal sensitivity to either disturbance. Table 3 further indicates that the LSM-based initial path has an intercept of 0.0719 after adding artificial noise, clearly lower than the 0.0828 obtained on the noise-free data. In contrast, the RANSAC intercepts across the three conditions, 0.0359, 0.0365, and 0.0368, change only slightly. The anti-noise ability of RANSAC was better than that of LSM, which indicated that RANSAC was a good option for primary path extraction.

**Table 3**

**Cage-row boundary line and initial path extraction using LSM and RANSAC**

| Point cloud noise | Method | Cage-row intercept $b_0$ / m | Cage-row slope $k_0$ | Initial path intercept $b_1$ / m | Initial path slope $k_1$ | Processing time / $10^{-3}$ s |
|---|---|---|---|---|---|---|
| Without noise | LSM | -0.606 | 0.242 | 0.0828 | 0.243 | 0.059 |
| | RANSAC | -0.628 | 0.227 | 0.0359 | 0.212 | 0.0126 |
| Gaussian noise | LSM | -0.722 | 0.187 | 0.0306 | 0.224 | 0.0638 |
| | RANSAC | -0.626 | 0.223 | 0.0365 | 0.211 | 0.0137 |
| Artificial noise | LSM | -0.681 | 0.152 | 0.0719 | 0.204 | 0.0625 |
| | RANSAC | -0.626 | 0.224 | 0.0368 | 0.210 | 0.0142 |



(1) No noise added

(2) Add Gaussian noise

(3) Add man-made noise

**Fig. 5 - Comparison of cage-row boundary line and initial path extracted by LSM and RANSAC**

*(a) Least square method; (b) Random sampling consistency.*

Runtime analysis shows that RANSAC is markedly faster than least squares across the clean, Gaussian-noise, and artificial-noise datasets. With a suitable iteration budget, it offers stronger real-time performance. Considering the anti-noise ability and real-time performance to extract the initial path, this study adopts the RANSAC method to extract the initial path.

### *3.3 Obstacle Avoidance Simulation and Experiment*

The inspection robot will inevitably encounter obstacles or staff during the inspection process. Therefore, whether it is a static or dynamic obstacle, the robot needs to make timely decisions to prevent collisions. At the same time, considering that the distance between two adjacent rows of chicken cages is narrow (generally 0.8-1.2 m), the inspection robot cannot avoid or detour the obstacles ahead. Thus, this study adopts the following obstacle avoidance strategy: after the inspection robot recognizes the obstacle, it stops moving at a distance of 0.5 m from the obstacle and waits for the dynamic obstacle to leave; if the obstacle is static, it will be prompted by wireless signal transmission.

#### *3.3.1 Simulation Analysis of Obstacle Avoidance:*

A simulation program was developed in MATLAB 2020a to verify the effectiveness of the improved A* algorithm in obstacle avoidance. The simulation environment was constructed as an 8*8 grid map, in which obstacles were placed along the robot's prospective path. The starting position coordinates of the robot are marked with squares, the end coordinates of the target point are marked with triangles, and obstacle coordinates, highlighted with circles, were placed between the robot and the goal to mimic real conditions. As shown in Figure 6, the obstacle lies on the line connecting the robot and its target, which is positioned ahead of the target. Where the target point and obstacle coordinates were set to (6.5, 6.5) and (7, 7), respectively.

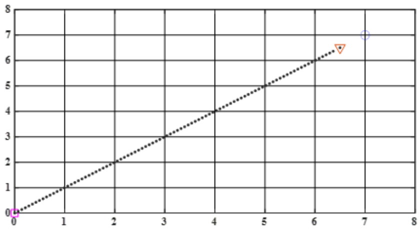
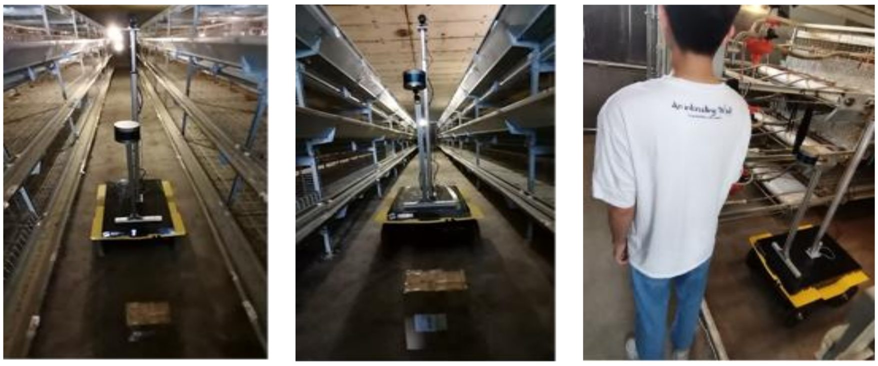


**Fig. 6 - Simulation results with the middle target point**

The data showed that the improved A* algorithm performed better in the simulation environment. The algorithm can recognize the obstacle avoidance operation of the robot by predicting and judging the obstacles ahead of the robot before moving, and simplifying the way of restricted barriers.

### *3.3.2 Experimental data on Obstacle Avoidance:*

A basic obstacle-avoidance trial was conducted with the proposed inspection robot. Three scenes were prepared: small static obstacles, large static obstacles, and moving obstacles. The robot's top speed in all tests was limited to 0.3 meters per second. Given this low speed, an influence radius of 0.5 meters was adopted, and any object detected within 0.5 meters ahead was treated as an obstacle. For the static cases, small and large objects were placed in the test range; the robot first adjusted to sense and demonstrate the surroundings, then followed the extracted path to assess its response. For the dynamic obstacle test, a person was instructed to suddenly move into the robot's forward path to simulate a transient obstruction. To simplify the experimental process, both the dynamic and static obstacles were assumed by humans. The experimenter stood still and acted as a stationary obstacle. After the robot successfully mapped and avoided this obstacle, the experimenter then moved rapidly toward the robot to represent a dynamic obstacle, and the robot's response was observed.



(a) Small obstacle avoidance (b) Large obstacle avoidance (c) Dynamic Obstacle Avoidance

**Fig. 7 - Obstacle avoidance scene and process**

As seen in Figure 7(a) and (b), the robot stops at 0.48 and 0.45 m away from the obstacles when facing static obstacles of different sizes. As shown in Figure 7(c), when the experimenter suddenly appeared in the forward motion of the robot as a dynamic obstacle, the machine stopped quickly. The tests demonstrate that the inspection robot efficiently extracts operational paths, responds promptly to static and moving obstacles, avoids collisions, and maintains stable and reliable performance.

## CONCLUSIONS

Based on research on the extraction method of chicken cage-row lines in poultry houses, this study proposed an enhanced A* algorithm to improve the navigation path of inspection robots. The improvement of the real-time and safety of the algorithm on the autonomous navigation of the robot is verified by experiments.

(1) The robot's functions were specified, and the hardware architecture was developed. Module models were selected and engineered, and a research testbed for the inspection robot was assembled.

(2) The performance of LSM and RANSAC in extracting chicken cage-row lines and primary paths was assessed in terms of extraction effect, real-time performance, and anti-noise ability, and the processing time, initial path intercept, and slope were analyzed. The RANSAC has better real-time performance, anti-noise ability, and extraction effect.

(3) Optimized the obstacle avoidance and path planning methods of the inspection robot. The simulation and actual test results showed that the improved A* algorithm could improve the efficiency of path planning, and the robot could make timely decisions on detecting static and dynamic and obstacles, and had better stability and reliability.

## DATA AVAILABILITY STATEMENT

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.'

## COMPETING INTERESTS

The author(s) declare no competing interests.

## FUNDING

This study was partly supported by the National Key Research Program of China, grant number 2021YFD1300405, and the Shandong Province Key Research and Development Program, grant number 2022TZXD0017-02.

Natural Science Foundation of Shandong Province-ZR2025MS836-The study of seeding orderly of the air-assisted seed metering device and the mechanism of evenly sowing at the high speed situation.

2022CXGC010612; Shandong Province Key R&D Program (Major Science and Technology Innovation Project)-Research and Development of High-Efficiency Production Equipment for Specialty Vegetables-2022CXGC010612.

Hunan Province Science and Technology Innovation Plan Project -2022JJ80343-Research on the seeding mechanism and sowing depth control of the small-particle-sized vegetable seed seeder based on Beidou navigation.

## ETHICS APPROVAL STATEMENT

All experiments were conducted with the approval of the ethical committee.

## ACKNOWLEDGMENTS

The authors of this study acknowledge the support provided by The Weifang University and Shandong Agricultural University, China.

## REFERENCES

[1] Bijjahalli, S., Sabatini, R. (2021) A High-Integrity and Low-Cost Navigation System for Autonomous Vehicles（适用于自动驾驶车辆的高可靠性低成本导航系统）[J]. *IEEE Transactions on Intelligent Transportation Systems*, Vol 22, no. 1, pp. 356-369. https://doi.org/10.1109/tits.2019.2957876;

[2] Birdal, T., Busam, B., Navab, N., Ilic, S., Sturm, P. (2020) Generic Primitive Detection in Point Clouds Using Novel Minimal Quadric Fits (基于新型最小二次曲面拟合的点云通用原始检测) [J]. *IEEE Trans Pattern Anal Mach Intell*, Vol 42, no. 6, pp. 1333-1347. https://doi.org/10.1109/TPAMI.2019.2900309;

[3] Chang, C.L., Xie, B.X., Wang, C.H. (2020) Visual Guidance and Egg Collection Scheme for a Smart Poultry Robot for Free-Range Farms (自由放养农场智能禽类机器人的可视化引导与取蛋方案) [J]. *Sensors (Basel)*, Vol 20, no. 22. https://doi.org/10.3390/s20226624;

[4] Charalampous, K., Kostavelis, I., Gasteratos, A. (2015) Thorough robot navigation based on SVM local planning (基于支持向量机局部规划的全面机器人导航) [J]. *Robotics and Autonomous Systems*, Vol 70, pp. 166-180. https://doi.org/10.1016/j.robot.2015.02.010;

[5] Ghahremani, M., Williams, K., Corke, F., Tiddeman, B., Liu, Y., Wang, X., Doonan, J.H. (2021) Direct and accurate feature extraction from 3D point clouds of plants using RANSAC (基于 RANSAC 算法从植物三维点云中进行直接且精确的特征提取) [J]. *Computers and Electronics in Agriculture*, Vol 187, p. 106240. https://doi.org/10.1016/j.compag.2021.106240;

[6] Guang, X., Gao, Y., Leung, H., Liu, P., Li, G. (2018) An Autonomous Vehicle Navigation System Based on Inertial and Visual Sensors (基于惯性与视觉传感器的自动驾驶车辆导航系统) [J]. *Sensors*, Vol 18, no. 9, p. 2952. https://doi.org/10.3390/s18092952;

[7] He, Y., Zhang, X., Zhang, Z., Fang, H. (2022) Automated detection of boundary line in paddy field using MobileV2-UNet and RANSAC（基于新型最小二次曲面拟合的点云通用原始检测)[J]. *Computers and Electronics in Agriculture*, Vol 194, p. 106697. https://doi.org/10.1016/j.compag.2022.106697;

[8] Hwang, H., Cho, G. (2020) Global Least Squares Path Modeling: A Full-Information Alternative to Partial Least Squares Path Modeling (一种替代偏最小二乘路径模型的全信息方法) [J]. *Psychometrika*, Vol 85, no. 4, pp. 947-972. https://doi.org/10.1007/s11336-020-09733-2;

[9] Ilci, V., Toth, C. (2020) High Definition 3D Map Creation Using GNSS/IMU/LiDAR Sensor Integration to Support Autonomous Vehicle Navigation (基于 GNSS/IMU/激光雷达传感器融合的高精度 3D 地图生成技术在自动驾驶车辆导航中的应用) [J]. *Sensors* (Basel), Vol 20, no. 3. https://doi.org/10.3390/s20030899;

[10] Jiang, H., Sun, Y. (2021) Research on global path planning of electric disinfection vehicle based on improved A* algorithm (基于改进 A*算法的电动消毒车全局路径规划研究) [J]. *Energy Reports*, Vol 7, pp. 1270-1279. https://doi.org/10.1016/j.egyr.2021.09.137;

[11] Li, C., Huang, X., Ding, J., Song, K., Lu, S. (2022) Global path planning based on a bidirectional alternating search A* algorithm for mobile robots (基于双向交替搜索 A*算法的移动机器人全局路径规划) [J]. *Computers & Industrial Engineering*, Vol 168, p. 108123. https://doi.org/10.1016/j.cie.2022.108123;

[12] Lim, Y.B., Ogawa, T., Hori, Y. (2022) Detection of restoration work by applying the RANSAC algorithm to the point cloud data from laser scanning: Case Study at Ostia, The International Archives of the Photogrammetry (基于激光扫描点云数据应用 RANSAC 算法检测修复工程：奥斯蒂亚案例研究) [J]. *Remote Sensing and Spatial Information Sciences*, Vol XLVI-2/W1-2022, pp. 315-321. https://doi.org/10.5194/isprs-archives-XLVI-2-W1-2022-315-2022;

[13] Malavazi, F.B.P., Guyonneau, R., Fasquel, J.-B., Lagrange, S., Mercier, F. (2018) LiDAR-only based navigation algorithm for an autonomous agricultural robot (基于纯激光雷达的自主农业机器人导航算法) [J]. *Computers and Electronics in Agriculture*, Vol 154, pp. 71-79. https://doi.org/10.1016/j.compag.2018.08.034;

[14] Patle, B.K., Babu L, G., Pandey, A., Parhi, D.R.K., Jagadeesh, A. (2019) A review: On path planning strategies for navigation of mobile robot (综述：移动机器人导航路径规划策略研) [J]. *Defence Technology*, Vol 15, no. 4, pp. 582-606. https://doi.org/10.1016/j.dt.2019.04.011;

[15] Ren, G., Lin, T., Ying, Y., Chowdhary, G., Ting, K.C. (2020) Agricultural robotics research applicable to poultry production: A review (适用于家禽生产的农业机器人研究：综述) [J]. *Computers and Electronics in Agriculture*, Vol 169, p. 105216. https://doi.org/10.1016/j.compag.2020.105216;