



Article

Visual Navigation and Obstacle Avoidance Control for Agricultural Robots via LiDAR and Camera

Chongyang Han^{1,2}, Weibin Wu¹, Xiwen Luo¹ and Jiehao Li^{1,2,*}

¹ Key Laboratory of Key Technology on Agricultural Machine and Equipment, Ministry of Education, College of Engineering, South China Agricultural University, Guangzhou 510642, China; scauhancy@stu.scau.edu.cn (C.H.)

² Department of Electronic Information and Bioengineering, Politecnico di Milano, 20133 Milan, Italy

* Correspondence: jiehao.li@ieee.org

Abstract: Obstacle avoidance control and navigation in unstructured agricultural environments are key to the safe operation of autonomous robots, especially for agricultural machinery, where cost and stability should be taken into account. In this paper, we designed a navigation and obstacle avoidance system for agricultural robots based on LiDAR and a vision camera. The improved clustering algorithm is used to quickly and accurately analyze the obstacle information collected by LiDAR in real time. Also, the convex hull algorithm is combined with the rotating calipers algorithm to obtain the maximum diameter of the convex polygon of the clustered data. Obstacle avoidance paths and course control methods are developed based on the danger zones of obstacles. Moreover, by performing color space analysis and feature analysis on the complex orchard environment images, the optimal H-component of HSV color space is selected to obtain the ideal vision-guided trajectory images based on mean filtering and corrosion treatment. Finally, the proposed algorithm is integrated into the Three-Wheeled Mobile Differential Robot (TWMDR) platform to carry out obstacle avoidance experiments, and the results show the effectiveness and robustness of the proposed algorithm. The research conclusion can achieve satisfactory results in precise obstacle avoidance and intelligent navigation control of agricultural robots.

Keywords: agricultural robot; visual navigation; obstacle avoidance control; LiDAR



Citation: Han, C.; Wu, W.; Luo, X.; Li, J. Visual Navigation and Obstacle Avoidance Control for Agricultural Robots via LiDAR and Camera. *Remote Sens.* **2023**, *15*, 5402. <https://doi.org/10.3390/rs15225402>

Academic Editor: Geert Verhoeven

Received: 6 September 2023

Revised: 12 November 2023

Accepted: 14 November 2023

Published: 17 November 2023



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

1. Introduction

Agricultural robots have been acknowledged as a highly promising frontier technology in agriculture. With the rapid development of modern electronic computers, sensors, the Internet of Things and other technologies, the application of agricultural robots is the best way to improve the efficiency and quality of agricultural production [1–3]. As a new type of intelligent agricultural equipment, agricultural robots integrate various advanced technologies such as artificial intelligence, automatic control and algorithmic development, and are widely used in the fields of agricultural harvesting, picking, transportation, and sorting, especially in the field of farm mobile inspection [4]. The key to the safe and efficient operation of agricultural mobile robots is to have obstacle avoidance and navigation capabilities, which require practicality and stability and, more importantly, simple and low-cost implementation [5]. However, other industrial mobile robots are not fully applicable in agricultural settings owing to technical limitations and differences in application scenarios. Similarly, unstructured agricultural scenarios and random and changeable environments pose huge challenges for agricultural robots. Therefore, the navigation and obstacle avoidance systems of agricultural robots must adopt stable and low-cost algorithms and control methods [6–8].

In order to develop agricultural robots, obstacle avoidance technology is the most important functional module of robots, which can improve the working efficiency of robots, reduce working errors, ensure that robots complete tasks and produce greater benefits.

The obstacle avoidance process of agricultural robot integrates agricultural information perception, intelligent control and communication technology. The robot obtains the surrounding environment information through relevant sensors, including the status, size, and position of obstacles, and then makes a series of obstacle avoidance strategies. Therefore, LiDAR [9], ultrasonic sensors, and vision sensors have been widely used in navigation and obstacle avoidance technology because of their high precision and fast speed. Rouveure et al. described in detail millimeter-wave radar for mobile robots, including obstacle detection, mapping, and general situational awareness. The selection of Frequency Modulated Continuous Wave (FMCW) radar is explained and the theoretical elements of the solution are described in detail in [10]. Ball et al. described a vision-based obstacle detection and navigation system that uses an expensive combination of Global Positioning System (GPS) and inertial navigation systems for obstacle detection and visual guidance [11]. Santos et al. proposed a real-time obstacle avoidance method for steep slope farmland based on local observation, which has been successfully tested in vineyards. The LiDAR combined with depth cameras will be considered in the future to improve the perception of the elevation map [12]. However, high sensor costs have impeded researchers. To address this, Monaca et al. explored the “SMARTGRID” initiative. This project utilizes an integrated wireless safety network infrastructure, incorporating Bluetooth Low Energy (BLE) devices and passive Radio Frequency Identification (RFID) tags, designed for identifying obstacles, workers, and nearby vehicles, among other things [13]. Advanced sensors enable potential obstacle avoidance in intricate agricultural surroundings, although their high-cost ratio to accuracy persists and they necessitate the integration of well-developed and stable navigation and algorithmic technologies.

Similarly, robot navigation in obstacle environments remains a challenging problem, especially in the field of agriculture, where accurate navigation of robots in complex agricultural environments is a prerequisite for completing various tasks [14]. Gao et al. discussed the solution to the navigation problem of Mobile Wheeled Robots (WMRs) in agricultural engineering and analyzed the navigation mechanism of WMRs in detail, but do not delve into their cost [15]. Wangt et al. reviewed the advantages, disadvantages, and roles of different vision sensors and Machine Vision (MV) algorithms in agricultural robot navigation [16]. Durand et al. built a vision-based navigation system for autonomous robots, which they noted is low-cost because trees in an orchard can be quickly and accurately identified using four stereo cameras to provide point cloud maps of the environment [17]. Higuti et al. described an autonomous field navigation system based on LiDAR that uses only two-dimensional LiDAR sensors as a single sensing source, reducing costs while enabling robots to travel in a straight line without damaging plantation plants [18]. Gutonneau et al. proposed an autonomous agricultural robot navigation method based on LiDAR data, including a line-finding algorithm and a control algorithm, and performed well in tests with the ROS middleware and the Gazebo simulator. They focused on the sophistication of the algorithm at the expense of the complexity of the algorithm and the stability of the simulation results [19]. LiDAR has become prevalent in agricultural navigation, including vineyards [20] and cornfields [21]. Nevertheless, the equilibrium between cost and practical outcomes is an aspect frequently disregarded by numerous scholars.

Currently, low cost is vital for the development of agricultural robots. Cheap robotic hardware systems and software systems are always welcome as the users are farmers working in the agricultural industry. Global Navigation Satellite System (GNSS) technology provides autonomous navigation solutions for today’s commercial robotic platforms with centimeter-level accuracy. Researchers are trying to make agricultural navigation systems more reliable by using a fusion of GNSS, GPS technology, and other sensor technologies [22]. Winterhalter et al. proposed a GNSS reference map based approach for field localization and autonomous navigation via crop rows [23]. However, GNSS-based solutions are expensive and require a long preparation phase, while on the other hand, GNSS signals can fail or be interrupted in complex agricultural environments, which is hard for farmers to accept.

In this paper, we have developed a low-cost agricultural mobile robot platform based on LiDAR and vision-assisted navigation. The LiDAR is used to acquire information about obstacles in the surrounding environment. An improved density-based fast clustering algorithm is used to find the clustering centers of obstacles, and the maximum diameters of convex polygons in the clustering data are obtained by combining the convex shell algorithm and the rotating jamming algorithm. In addition, an effective orchard path tracking and control method is proposed in this paper, and effective and obvious feature components are selected as the color space components used by the robot in real-time tracking, which initially solves the orchard path feature recognition task and realizes the robot's visual aided navigation. Finally, the feasibility of the proposed method in obstacle avoidance and navigation of agricultural robots is verified by experiments.

The main contributions of this paper are summarized in the following folds.

- An improved density-based fast clustering algorithm is proposed, combined with a convex hull algorithm and rotating jamming algorithm to analyze obstacle information, and an obstacle avoidance path and heading control method based on the dangerous area of obstacles is developed. The feasibility of low-cost obstacle avoidance using LiDAR in agricultural robots is realized.
- By analyzing the color space information of the orchard environment road and track road images, a robot track navigation route and control method based on image features were proposed. A vision camera was used to assist agricultural robots in navigation, and the orchard path feature recognition task was preliminarily solved.
- An agricultural robot navigation and obstacle avoidance system based on a vision camera and LiDAR is designed, which makes up for the shortcomings of poor robustness and easy to be disturbed by the environment of a single sensor, and realizes the stable work of agricultural robots working in the environment rejected by GNSS.

The rest of this article is organized as follows. Section 2 introduces the calibration work of LiDAR and visual camera. Section 3 provides the development and proposed method of robot obstacle avoidance system. Section 4 presents the research of robot vision system. In Section 5, the obstacle avoidance and navigation experiments of the robot are carried out. Finally, we make conclusions work of the whole manuscript.

2. Preliminaries

2.1. LiDAR Calibration and Filtering Processing

Since the process of collecting environmental data by LiDAR is relative to the form of LiDAR polar coordinate system [24], if LiDAR is installed on the robot platform, the coordinate system will be based on the robot. Therefore, it is necessary to re-calibrate the coordinate system relationship between LiDAR and the robot. There are three coordinate definitions in the robot positioning system, which are global coordinate system $X_wO_wY_w$, robot coordinate system $X_vO_vY_v$ and LiDAR coordinate system $X_sO_sY_s$. The relationship between the robot and LiDAR coordinates is determined by the LiDAR's relative position to the robot. Therefore, the global coordinate system and the robot coordinate system are the main research coordinate systems, in which the LiDAR coordinate system is $X_sO_sY_s$, and the typical robot coordinate system relationship is shown in Figure 1. The pose of the robot in the global coordinate system is denoted as (x, y, θ) , with (x, y) representing the robot position, and θ representing the heading angle [25]. The relationship between any P point in the robot coordinate system and the global coordinate system can be obtained by Equation (1).

$$\begin{pmatrix} x_w \\ y_w \end{pmatrix} = \begin{bmatrix} \cos(\theta - \frac{\pi}{2}) & -\sin(\frac{\pi}{2}) \\ \sin(\theta - \frac{\pi}{2}) & \cos(\theta - \frac{\pi}{2}) \end{bmatrix} \times \begin{pmatrix} x_v \\ y_v \end{pmatrix} + \begin{pmatrix} X_v \\ Y_v \end{pmatrix} \quad (1)$$

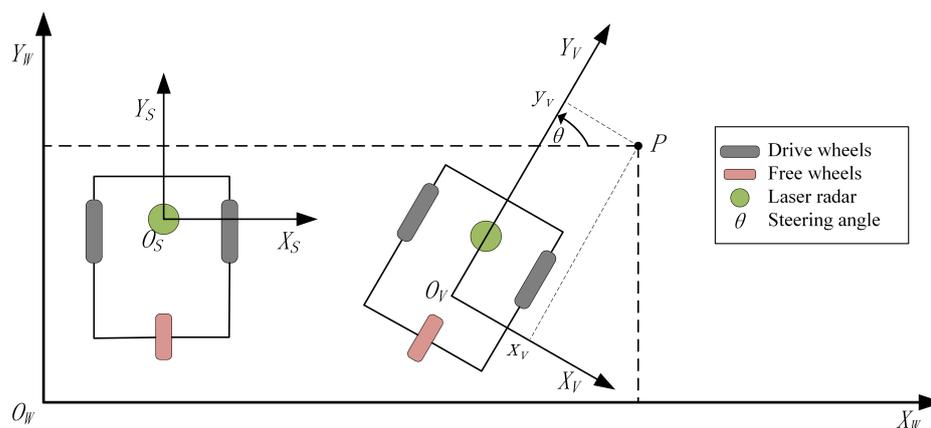


Figure 1. The relationship between robot, LiDAR, and global coordinate system.

The polar coordinate system is the working mode of LiDAR. Firstly, the LiDAR installed on the robot needs to be calibrated to obtain the relationship between LiDAR and robot [26,27]. As shown in Figure 1, the LiDAR coordinate system is $X_sO_sY_s$, so the Cartesian coordinate of LiDAR is (x_s, y_s) , the polar coordinate form is (L, β) . The conversion relationship between Cartesian coordinates and polar coordinates of LiDAR is shown in Equation (2).

$$x_s = L \cos \beta, y_s = L \sin \beta \tag{2}$$

The pose coordinate of the LiDAR is (x_s, y_s, ϕ) , where ϕ is the angle from axis O_vX_v to axis O_sX_s of the robot coordinate system. Suppose the coordinate of LiDAR coordinate point s under robot is (x_{vs}, y_{vs}) , then the rotation translation of the coordinate system is as shown in Equation (3) [28].

$$\begin{pmatrix} x_{vs} \\ y_{vs} \end{pmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \times \begin{pmatrix} x_s \\ y_s \end{pmatrix} + \begin{pmatrix} X_s \\ Y_s \end{pmatrix} \tag{3}$$

According to the above calibration method, a calibration test is carried out. Firstly, the pose of the robot in global coordinates is determined as (X_s, Y_s, ϕ) . The coordinate of the auxiliary calibration object placed in the global coordinate system is (X_{wi}, Y_{wi}) . According to the coordinate system relations in Figure 1, the relational data of the global coordinate system and the LiDAR coordinate system can be calculated.

The center of the auxiliary calibration object in the LiDAR coordinate system is calculated as (X_{si}, Y_{si}) by the least square method. Further, the LiDAR pose coordinate is obtained by the m-estimation method in Equation (4).

$$\begin{pmatrix} x_w \\ y_w \end{pmatrix} = \begin{bmatrix} \cos(\theta - \frac{\pi}{2}) & -\sin(\frac{\pi}{2}) \\ \sin(\theta - \frac{\pi}{2}) & \cos(\theta - \frac{\pi}{2}) \end{bmatrix} \times \left\{ \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \times \begin{pmatrix} x_s \\ y_s \end{pmatrix} + \begin{pmatrix} X_s \\ Y_s \end{pmatrix} \right\} + \begin{pmatrix} X_v \\ Y_v \end{pmatrix} \tag{4}$$

Due to the data collected by LiDAR producing interference cloud signal, such interference noise will have an unstable influence on the detection effect of obstacles [29]. The laser radar data with median filtering effect depends on the filter window options, therefore, this paper adopts an improved median filtering method, which can first determine whether the data point is in the extreme value within the filtering window range. If yes, the point can be filtered; otherwise, the operation will be canceled [30]. Figure 2b shows the effect of the improved median filtering algorithm, the filtering algorithm is shown in Equation (5).

$$x_i = \begin{cases} \text{med}\{x_{i-s}, \dots, x_i, \dots, x_{i+s}\} & x_i > x_{\max} \text{ or } x_i \leq x_{\min} \\ \dots x_i & x_{\min} \leq x \leq x_{\max} \end{cases} \tag{5}$$

$$x_{\min} = \min\{x_{i-s}, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_{i+s}\}$$

$$x_{\max} = \max\{x_{i-s}, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_{i+s}\}$$

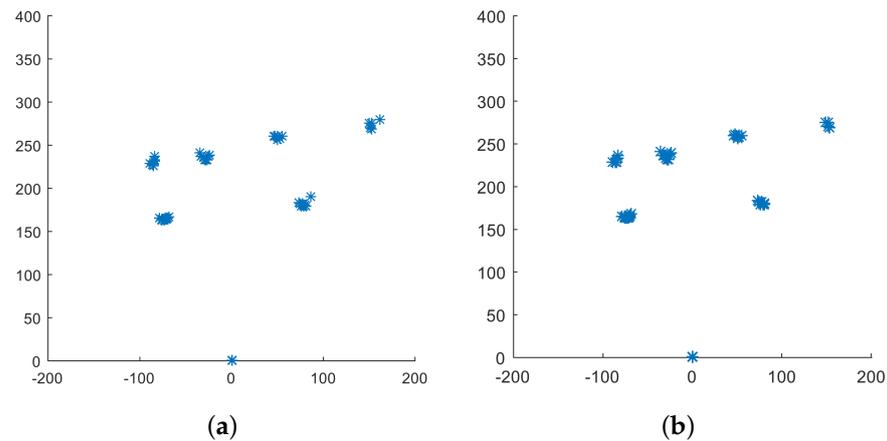


Figure 2. Comparison of median filtering effect before (a) and after (b).

2.2. Camera Visualization and Parameter Calibration

The camera is calibrated using the checkerboard method, with key points acquired. The MATLAB camera calibration function is called, and the camera parameters are calculated by the Zhang Zhengyou calibration algorithm. External parameters of the camera can be visually checked by visualizing 3D images to check the position of the checkerboard calibration board and camera [31,32]. Figure 3a,b are visualized 3D views based on the checkerboard calibration board reference frame and camera reference frame, respectively.

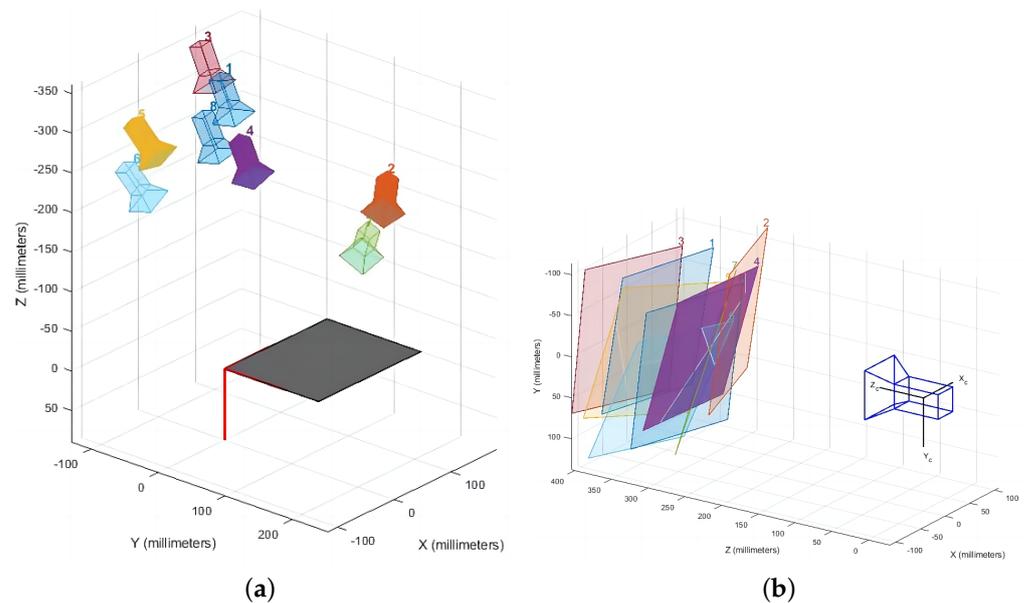


Figure 3. Visual 3D view of different reference frame modes. (a) Reference frame based on checkerboard calibration board; (b) camera reference system.

Finally, the values of internal parameters and external parameters in the camera are calculated by the MATLAB camera calibration algorithm. The matrix A in the camera is as follows:

$$A = \begin{bmatrix} 2.8075 \times 10^3 & 0 & 0 \\ 8.8390 & 2.8217 \times 10^3 & 0 \\ 802.8544 & 650.4047 & 1 \end{bmatrix} \quad (6)$$

The radial distortion coefficient K of the camera lens is as follows:

$$k = [-0.0418 \dots -0.01981 \dots 0.7706] \quad (7)$$

The tangential distortion coefficient P of the camera lens is as follows:

$$p = [0.0024 \cdots 0.0023] \quad (8)$$

3. Obstacle Avoidance System and Algorithm Implementation

In this section, for the obstacle avoidance system designed for autonomous robots, LiDAR is used to collect static obstacles in the moving direction in real-time; in addition, the obstacle information is analyzed by an improved clustering algorithm combined with the convex hull algorithm and the rotating jamming algorithm. Finally, an obstacle avoidance path and heading control method based on the dangerous area of obstacles is developed.

3.1. Improved Clustering Algorithm of Obstacle Point Cloud Information

The present research utilizes an improved algorithm to rapidly and accurately cluster obstacle cloud data, followed by determining the clustering center, which helps locate the obstacle cloud data center [33]. Compared with the traditional K-means algorithm, the clustering algorithm can obtain non-spherical clustering results, describe the data distribution well, and has lower algorithm complexity than the K-means algorithm. This paper first gives a conceptual definition of the algorithm. Two quantities need to be calculated for each data point i , including the local density and the minimum distance above the density of point i .

Local density is defined as follows:

$$\rho_i = \sum_j k(d_{ij} - d_c) \quad (9)$$

where

$$k(x) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{if } x \geq 0 \end{cases} \quad (10)$$

In this paper, the d_c is a truncation distance which refers to the distance between all points from smallest to largest, accounting for two percent of the total. Distance is defined as

$$\begin{aligned} j : \rho_j > \rho_i \\ \delta_i = \min(d_{ij}) \end{aligned} \quad (11)$$

The δ_i denotes the smallest distance of point i among points whose density surpasses that of point i . A higher value indicates that point i is at a further distance from the high-density point, making it more likely to become the cluster center. For point i with the highest global density, δ_i is equal to the distance between point i and the maximum value of all points [34]. The obstacle point distribution, generated by the enhanced clustering algorithm, is presented in Figure 4.

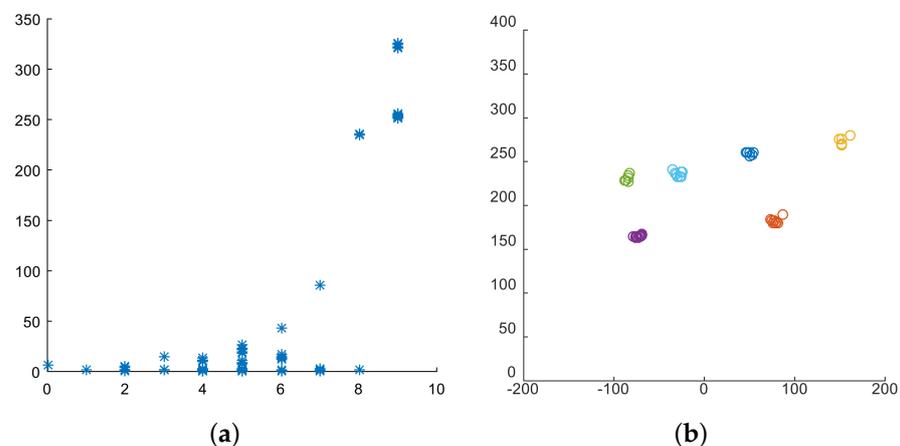


Figure 4. (a) The diagram of data decision; (b) the result of clustering algorithm.

3.2. The Fusion of Convex Hull Algorithm and Rotary Jamming Algorithm

Convex hull is a concept in computer geometry. In a real vector space V , for a given set X , the intersection S of all convex sets containing X is called the convex hull of X . The convex hull of X can be constructed from a linear combination of all points X_1, X_2, \dots, X_n in X . We use the Graham's Scan algorithm to find the convex polygon corresponding to the obstacle clustering data collected by LiDAR.

The basic idea of Graham's Scan algorithm is as follows: the convex hull problem is solved by setting a stack S of candidate points. Each point in the input set X is pushed once, and points that are not vertices in $CH(X)$ (representing the convex hull of X) are eventually popped off the stack. At the end of the calculation, there are only vertices in $CH(X)$ in stack S in the order in which the vertices appear counterclockwise on the boundary. Graham's Scan algorithm is shown in Table 1. If there is more than one such point, the leftmost point is taken as P_0 , meanwhile $|X| \geq 3$. Call the function $TOP(S)$ TO return the point at the TOP of the stack S , and call the function $next\text{-to}\text{-top}(S)$ TO return the point below the TOP of the stack without changing the structure of the stack [35]. The layout of obstacle cloud points processed by Graham's Scan algorithm is shown in Figure 5a.

Table 1. Graham's Scan (X) algorithm.

Steps	Content
Input	The remaining points in $X(P_1, P_2, \dots, P_n)$;
Output	Convex hull results
1	Sort (P_1, P_2, \dots, P_n) by its polar angle counterclockwise to P_0 ;
2	If several points have the same polar angle;
3	Then, remove the rest of the points;
4	Then, leaving only the point farthest from P_0 ;
5	PUSH(P_0, S);
6	PUSH(P_1, S);
7	PUSH(P_2, S);
8	for $i \leftarrow 3$ to m ;
10	Call the function :TOP(S) and P_i form a non-left-turn do;
11	Call function NEXT-TOP(S) and return the following point;
12	POP(S);
13	PUSH(P_i, S);
14	end for
15	return S

To determine the maximum diameter of the obstacle point cloud's convex hull, some scholars have utilized the enumeration technique. However, this method, although easy to implement, sacrifices efficiency due to its time complexity of $O(n^3)$. Consequently, this paper introduces an alternative algorithm, the rotating caliper algorithm, which provides a time complexity of $O(n)$ for computing convex hull diameter. Before introducing the Rotating Calipers Algorithm, we defined the term "heel point pair": if two points P and Q (a convex polygon Q) lie on two parallel tangent lines, they form a heel point pair. What appears in this paper is the "point-edge" heel point pair, as shown in Figure 5b, which occurs when the intersection of one tangent line and polygon is one edge, and the other tangent line is unique to the tangent point of the polygon. The existence of tangents must contain the existence of two different "point-point" pairs of heel points. Therefore, p_1 and q , p_2 , and q are two pairs of "heel point pairs" of convex polygon q , respectively. The Rotating Calipers Algorithm is shown in Table 2.

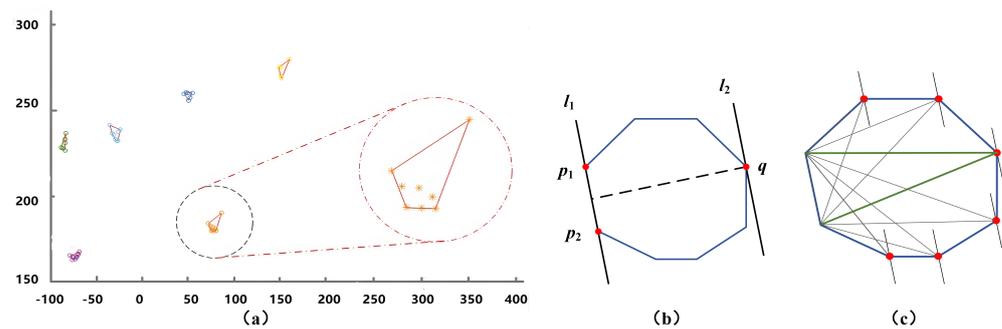


Figure 5. (a) The layout of obstacle cloud points processed by Graham's Scan algorithm; (b) diagram of "point-edge" to heel point; (c) the distance between convex envelope and corresponding edge.

Table 2. Rotating Calipers algorithm.

Steps	Content
Input	Point coordinates of the polygon(x_i, y_i);
Output	Maximum diameter(d_{max}) and the "heel point pair"
1	$y_{min} \leftarrow \min(y_i)$;
2	$y_{max} \leftarrow \max(y_i)$, calculate the endpoints y_{min} and y_{max} ;
3	$n \leftarrow 1$;
4	While $n < N$ do;
5	$n + +$;
6	$d_{max} \leftarrow (d_1)$, calculate the distance d_1 between y_{min} and y_{max} ;
7	Design two horizontal tangents l_1 and l_2 through y_{min} and y_{max} ;
8	Rotate l_1 and l_2 until they coincide with the other side of the polygon;
9	$d_{max} \leftarrow \max(d_{max}, d_2)$, Create new heel points;
10	Calculate the new distance d_2 and compare the sizes;
11	End

There are the two furthest points on the convex hull through which a pair of parallel lines can be drawn. By rotating the pair of parallel lines to coincide with an edge on the convex hull, such as the l_1 line in Figure 5b, notice that q is the point farthest from the l_1 line on the convex hull. Points on the convex hull in turn form a single-peak function with the distance generated by the corresponding edge, as shown in Figure 5c. According to the characteristics of the convex hull, enumerating edges counterclockwise can avoid the repeated calculation of the farthest point and obtain the maximum pair of points of the convex hull diameter. The maximum diameter d_{max} of the convex polygon in Figure 5a is 5.65 cm, 15.55 cm, 12.65 cm, 9.43 cm, 8.94 cm, and 12.04 cm, respectively, processed by the rotating jamming algorithm. Finally, the circle with the maximum diameter of the convex polygon d_{max} is taken as the danger range.

3.3. Path Planning and Heading Control of the Robot

3.3.1. Obstacle Avoidance Path Planning

This paper improved the VFH+ path planning method to meet the basic obstacle avoidance requirements of the robot [36]. According to the distribution of obstacles in the surrounding environment on the guiding track of the robot, the passage direction interval is determined according to a certain angle threshold, and then the passage direction and speed are determined through calculation. Then, the robot is taken as the center to establish the coordinate system, as shown in Figure 6a. The steps are as follows.

(1) The path design of this paper adopts the particle filter theory considering the size of the robot, based on the obstacle identification and positioning in the previous section, the radius is expanded by expanding the cluster center circle to half the width of the robot. Through dynamic constraints, the robot will be limited in the turning process by setting a restricted area of d_1 and d_2 on both sides of the robot [37,38]; meanwhile, the dynamic constraint angle is set at 30° . As shown in Figure 6a, both A and B are obstacles, and the

expansion part is the gray part on the edge of the obstacle. The LiDAR distance from the obstacle to the robot can be known from polar coordinates. Assuming the LiDAR scanning range $[0, 360]$, angle α and distance l are expressed as a bar graph, as shown in Figure 6b.

(2) Determine the threshold and obtain the passage interval. According to the threshold $H(i)$, it is defined as follows:

$$H(i) = \begin{cases} 1 & l(i) \leq T \text{ and } l(i) \neq 0 \\ 0 & l(i) > T \text{ or } l(i) = 0 \end{cases} \quad (12)$$

When $H(i) = 0$, it indicates that i direction is passable and vice versa. In combination with the robot specifications and safe distance, the threshold is set at 20 cm, and the angle α and distance l are expressed as bar charts, as shown in Figure 6b.

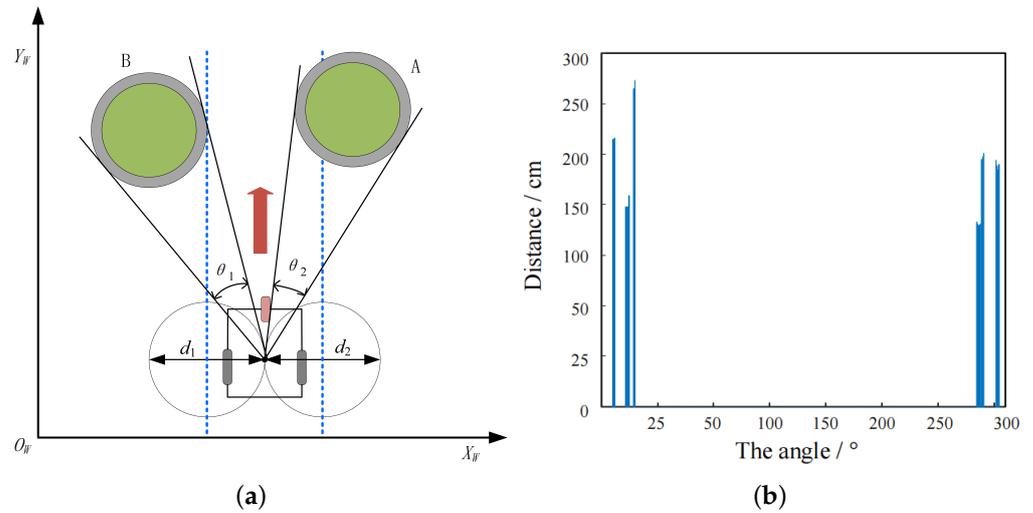


Figure 6. (a) Schematic diagram of obstacle angle determination; (b) the result diagram of obstacle angle determination.

3.3.2. Heading Control of the Robot

The movement of the robot is described by the motion process and motion model through state vectors in different time periods. The input control of the robot is shown in Equation (13) [39].

$$u = [v_A \quad \theta_A \quad \tau_{rows} \quad \alpha_{rows}]^T \quad (13)$$

v_A (m/s) is the linear velocity obtained from the wheel radius, θ_A (rad/s) is the speed measured by the rotational yaw rate, τ_{rows} (m) is the offset of the previous frame tracked by the trajectory, and α_{rows} (m) is a tracking mark.

The movement of the robot can be approximated by a small range of movement through a suitable rotation fine-tuning in each cycle. The forward motion of a robot is determined by speed v_A . The robot deflection is determined by a combination of visual guidance and LiDAR. The robot will prioritize visual tracking when the obstacle distance reaches a safe threshold, which triggers the LiDAR obstacle avoidance program [40,41], $\alpha_{rows} = 0$, the steering angle of the robot is Equation (14).

$$\theta_{v,k+1} = \theta_{v,k} + \hat{\theta}_{vk} \Delta t \quad (14)$$

The tracker refreshes output through the robot's heading, which is defined as the lateral offset of a navigation point (point C in Figure 7) that is perpendicular to the track and heading β_{rows} . On each refresh, navigation point C is perpendicular to the robot heading of the track distance, and then output. It can be seen from Figure 7 that the robot heading changes at frame $k + 1$ as follows [42,43]:

$$\theta_{v,k+1} = \beta_{rows} + \alpha \tag{15}$$

α : the angle of the robot with respect to the track angle β_{rows} , which can be seen in Figure 7.

$$\alpha = \alpha \sin\left(\frac{\gamma + \tau_{rows}}{l_{AC}}\right) \tag{16}$$

where γ is described as follows:

$$\gamma = (l_{AC} - v_{A,k} \Delta t) \sin(\beta_{rows} - \theta_{v,k}) \tag{17}$$

l_{AC} is the distance from point A to point C.

By combining the above equations, the angle of robot can be obtained as shown in Equation (18).

$$\theta_{v,k+1} = \begin{cases} \beta_{rows} + \alpha \sin\left(\frac{(l_{AC} - v_{A,k} \Delta t) \sin(\beta_{rows} - \theta_{v,k}) + \tau_{rows}}{l_{AC}}\right) & \alpha_{rows=1} \\ \theta_{v,k} + \theta_{v,k} \Delta t & \text{others} \end{cases} \tag{18}$$

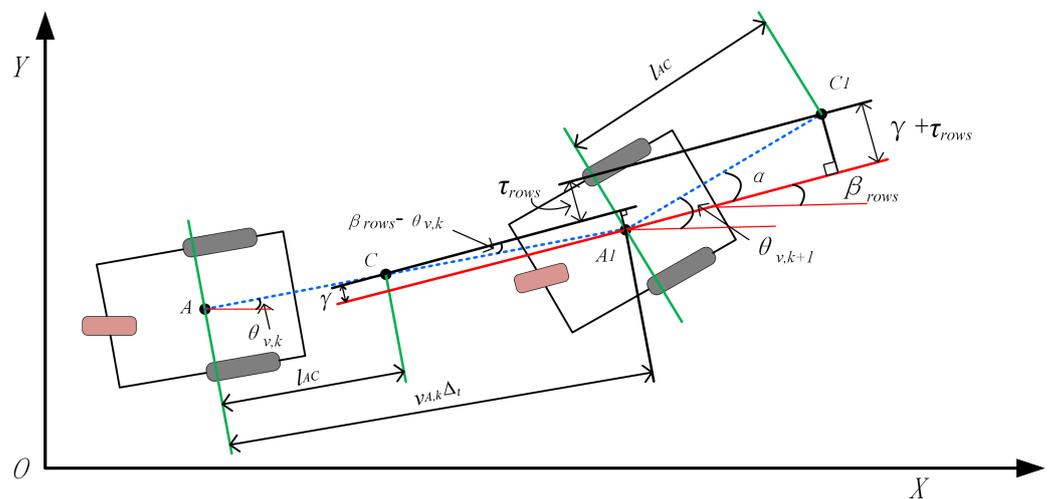


Figure 7. The course change chart of avoidance on robot.

4. Visual Navigation System

In order to achieve the visual guidance of the autonomous robot, in this section, statistical analysis of color space was carried out on the track-guiding road images and the surrounding environment road images collected by the vision system. The working conditions were selected as follows (straight line, left turn, right turn, intersection, and Y-shaped intersection). Through the comparative analysis results of color space, HSV color space was adopted as the preferred color space. On the H component of the track guide road image, the rectangular mean value is filtered, and the threshold value is obtained by Otsu’s algorithm for binarization processing. Finally, the ideal track guide image is obtained by corrosion processing.

4.1. Color Space of Track Road and Environment Road

In this section, trajectory road and environmental road images collected by robots are researched, the driving conditions of the robot are as follows: straight line driving, left turn, right turn, intersection and Y-shaped intersection, as shown in Figure 8. The red circle represents the robot’s walking track road image, and the black circle represents the surrounding environment road image. We analyzed the color system of the track road image and the environment road image, respectively, including the first-order moment (mean value) and second-order moment (variance) of the RGB component and HSV com-

ponent [44]. The mean, which is the first moment, establishes the average intensity of every color component. The second moment, known as variance, indicates the variation in colors within the area under observation, commonly referred to as non-uniformity.

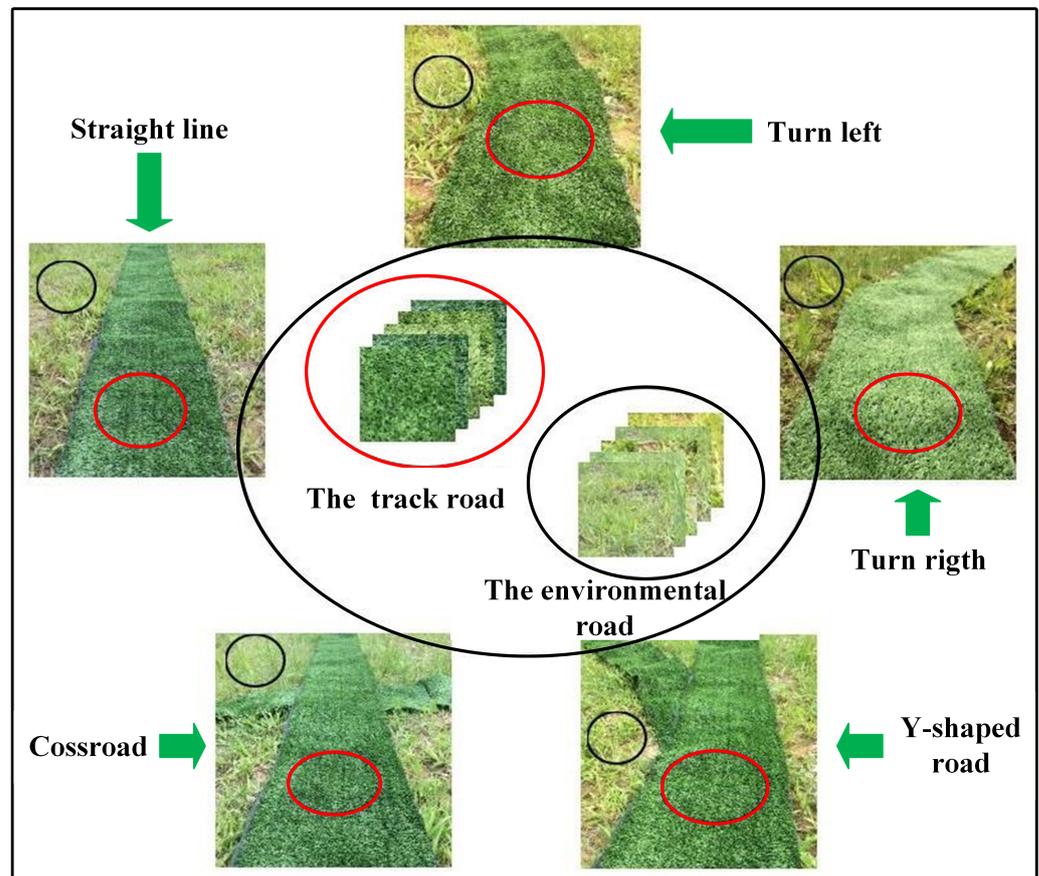


Figure 8. The image information acquisition of autonomous robot.

The statistical results for the mean value of the RGB component in both the trajectory road and environmental road images are displayed in Figure 9a below. The threshold values of the RGB components of the trajectory road images are all lower than those of the ambient road images, and only the G and B component thresholds of the trajectory road images are higher than those of the ambient road images in the right turn condition. The statistical results of the second moment (variance) are shown in Figure 9b below. The threshold values of the RGB components of the trajectory road images are all higher than those of the ambient road images, which also reflects the large difference between the trajectory road images and the ambient road images. The statistical results of the first moment (mean) of the HSV component of the track road image and the environmental road image are shown in Figure 9c below. The H-component threshold is the lowest for the track and environmental road images, and the H-component threshold is higher for the track image than for the environmental road image, which is consistent across the five operating conditions. The statistical results for the second moment (variance) are displayed in Figure 9d, indicating the lowest H-component threshold which presents some degree of stability.

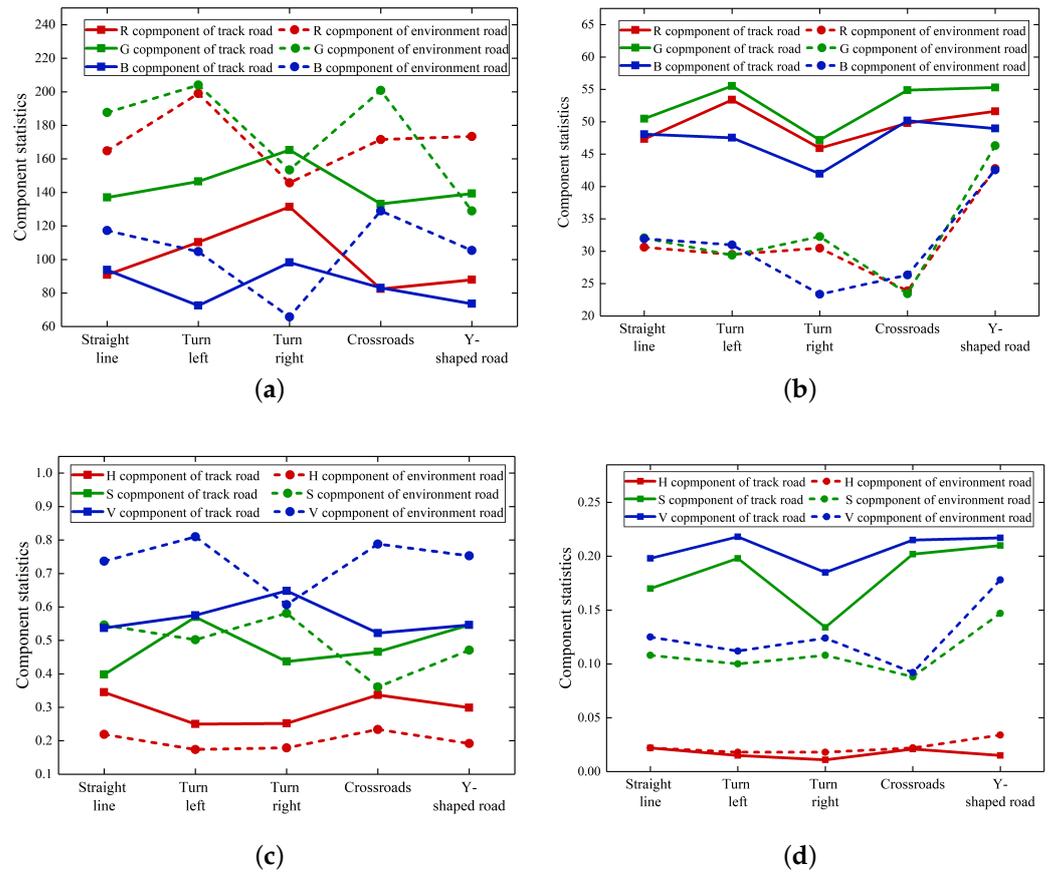


Figure 9. The mean and variance analysis results of RGB and HSV components of trajectory road image and environment road image. (a) The mean value of the RGB component; (b) the variance of the RGB component; (c) the mean value of the HSV component; (d) the variance of the HSV component.

4.2. The Processing of Road Image of Guiding Trajectory

The distinction between the trajectory road image and the environment road image is key to the visual navigation of the robot. Analyzing the color information of trajectory road images can help robots judge and autonomously navigate based on the image information. The processing of track road images includes statistical analysis of image RGB, rgb, $L^*a^*b^*$, HSV, and HIS color space, as shown in Figure 10. RGB is the basic color system of an image and is one of the most widely used color systems. HSV represents hue, saturation, and value, respectively, [45]. This color system more realistically reflects human reasoning and color observations than the RGB system. HSI, which stands for color, saturation, and intensity, is a framework that separates the intensity factor from the color knowledge presented by a color image [46].

Switching between RGB and HIS is accomplished by

$$H = \begin{cases} \theta & B \leq G \\ 360 - \theta & B > G \end{cases} \quad (19)$$

then

$$\theta = \cos^{-1} \left\{ \frac{0.5[(R - G) + (R - B)]}{[(R - G)^2 + (R - G)(G - B)]^{1/2}} \right\} \quad (20)$$

Saturation S:

$$S = 1 - \frac{3}{(R + G + B)} [\min(R, G, B)] \quad (21)$$

Brightness I:

$$I = \frac{1}{3}(R + G + B) \quad (22)$$

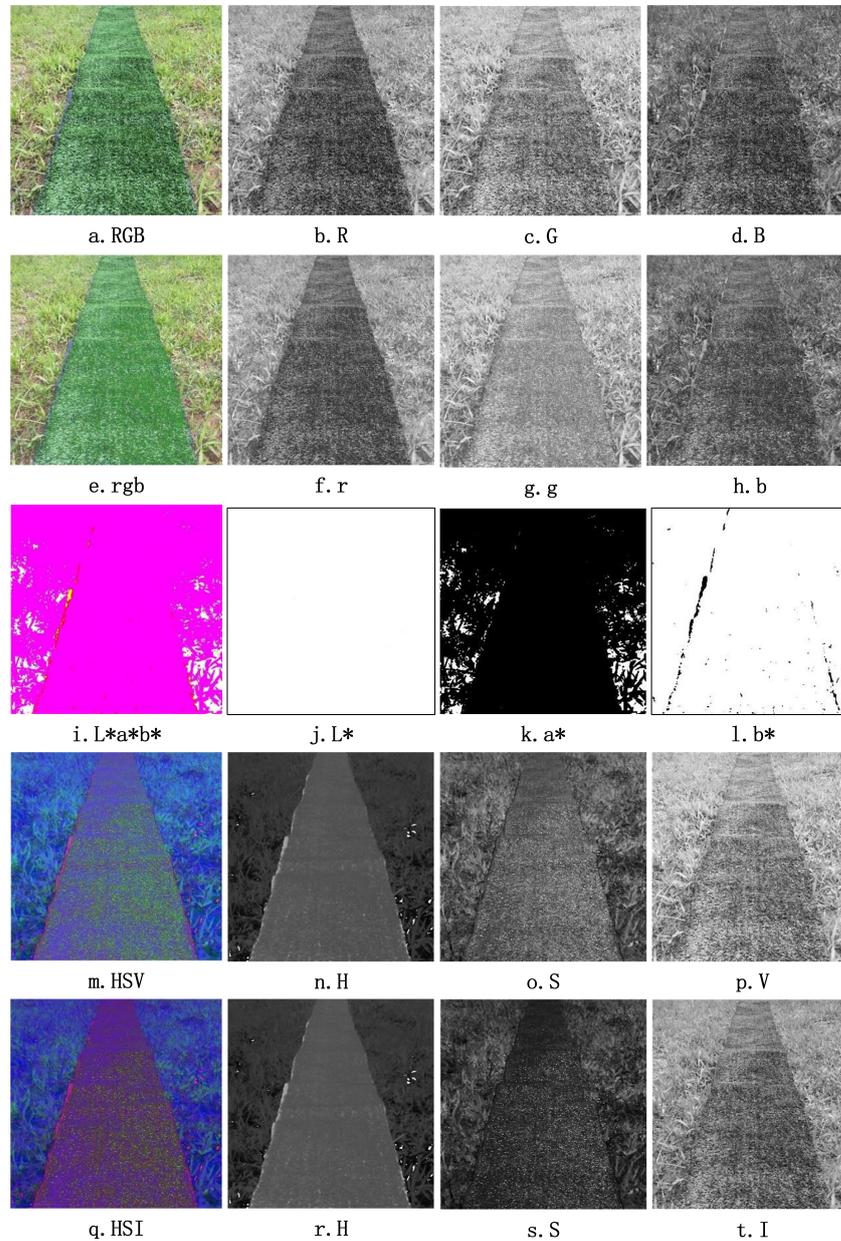


Figure 10. The color space diagram of environmental road and track road, where (a–d) the images of RGB; (e–h) the images of rgb; (i–l) the images of L*a*b*; (m–p) the images of HSV; (q–t) the images of HSI.

The comparison and analysis results of color space between the trajectory road image and the environment road image show that the H-component segmentation effect is the best in HSV color space. First, we filter the guide trajectory of the H-component map of the road by using the rectangular mean filter, which has a faster calculation speed. The filtering method is shown in Equation (23).

$$g(x, y) = \frac{1}{M} \sum_{f \in S} f(x, y) \quad (23)$$

Otsu's algorithm is used to calculate the optimal threshold of the image [47]. Let the gray level of the gray image of the trajectory be L , then the gray range is $[0, L - 1]$. The threshold is calculated as shown in Equation (24).

$$T = \text{Max} \left[w_0(t) \cdot (u_0(t) - u)^2 + w_1(t) \cdot (u_1(t) - u)^2 \right] \quad (24)$$

where T represents the threshold, w_0 is the background proportion, u_0 is the background mean, w_1 is the foreground proportion, u_1 is the foreground mean, and u is the average of the entire image. Otsu's algorithm was used to calculate the gray image thresholds of the guide track straight lines as $T_1 = 0.2706$ and $T_2 = 0.2667$, respectively. In the end, by constructing positive diamond structure elements with equal diagonal lines, the distance from the origin of the structure element to the furthest point of the positive diamond was defined as 25 pixels. Corrosion is defined in Equation (25).

$$A \ominus B = \{z \mid (B)_z \cap A^c = \emptyset\} \quad (25)$$

where $A \ominus B$ denotes that A is corroded by B , and z is the set of all points z where B in A is translated by z . A^c is the complement of A . The linear trajectory driving effect of the robot after corrosion treatment is shown in Figure 11.

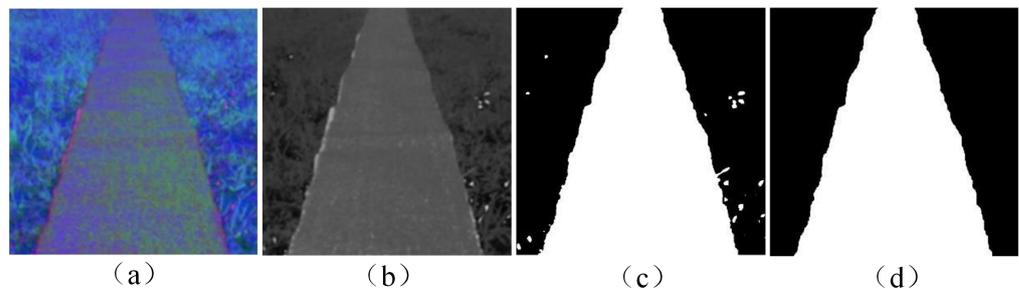


Figure 11. Comparison of image processing effect of guide track road. (a) The HSV diagram of linear trajectory after mean filtering; (b) the filtered H component of HSV color space; (c) the trajectory diagram after threshold segmentation; (d) the track image after corrosion treatment.

4.3. Fuzzy Logic Vision Control System Based on Visual Pixels

In this section, to improve the robustness of the control system, a visual pixel-based fuzzy logic vision system is designed based on fuzzy control theory for the final visual trajectory images obtained by the autonomous robot. The principle is as follows: 11 seed points are selected on the vertical centerline of the trajectory image, a horizontal line is made from the above seed points, and the left and right sides are scanned to determine the path widths of the left and right sides, respectively. If the number of pixels on the left and right sides is the same, it is determined to be in the middle of the path. If the number of pixels on the left side is greater than the number of pixels on the right side, the right side of the deviation path is determined, and then the fuzzy control system makes a shift to the left centerline. If the number of pixels on the right is greater than the number of pixels on the left, the left side of the deviation path is determined, and then the fuzzy control system will make an offset motion to the right center line [48]. In this paper, the fuzzy logic vision control system is designed with two inputs and two outputs. The inputs are PCLeft and PCRight and the outputs are speed instruction v and angle instruction θ , as shown in Equation (26).

$$\begin{aligned} \text{PCLeft} &= \frac{1}{11} \sum_{i=1}^{11} pcl_i \\ \text{PCRight} &= \frac{1}{11} \sum_{i=1}^{11} pcr_i \end{aligned} \quad (26)$$

where pcl_i and pcr_i are the number of pixel points on the left and right side of the scan line, respectively.

Fuzzy membership functions of input variables are obtained in Equations (27) and (28), where each input variable is fuzzy with three membership functions: Small (S), Medium (M) and Large (L). Membership functions of PCLeft and PCRight are shown in Figure 12.

$$\begin{aligned} \mu_S(\text{PCLeft}) &= \begin{cases} \left(\frac{S_{c1} - \text{PCLeft}}{S_{c1} - S_{b1}}\right) & S_{b1} < \text{PCLeft} < S_{c1} \\ 1 & 0 \leq \text{PCLeft} \leq S_{b1} \\ 0 & \text{other} \end{cases} \\ \mu_M(\text{PCLeft}) &= \begin{cases} \left(\frac{\text{PCLeft} - m_{a1}}{m_{b1} - m_{a1}}\right) & m_{a1} < \text{PCLeft} < m_{b1} \\ \left(\frac{m_{c1} - \text{PCLeft}}{m_{c1} - m_{b1}}\right) & m_{b1} \leq \text{PCLeft} \leq m_{c1} \\ 0 & \text{other} \end{cases} \\ \mu_L(\text{PCLeft}) &= \begin{cases} \left(\frac{\text{PCLeft} - l_{a1}}{l_{b1} - l_{a1}}\right) & l_{a1} < \text{PCLeft} < l_{b1} \\ 1 & \text{PCLeft} \geq l_{b1} \\ 0 & \text{other} \end{cases} \end{aligned} \tag{27}$$

$$\begin{aligned} \mu_S(\text{PCRight}) &= \begin{cases} \left(\frac{S_{cr} - \text{PCRight}}{S_{cr} - S_{br}}\right) & S_{br} < \text{PCRight} < S_{cr} \\ 1 & 0 \leq \text{PCRight} \leq S_{br} \\ 0 & \text{other} \end{cases} \\ \mu_M(\text{PCRight}) &= \begin{cases} \left(\frac{\text{PCRight} - m_{ar}}{m_{br} - m_{ar}}\right) & m_{ar} < \text{PCRight} < m_{br} \\ \left(\frac{m_{cr} - \text{PCRight}}{m_{cr} - m_{br}}\right) & m_{br} \leq \text{PCRight} \leq m_{cr} \\ 0 & \text{other} \end{cases} \\ \mu_L(\text{PCRight}) &= \begin{cases} \left(\frac{\text{PCRight} - l_{ar}}{l_{br} - l_{ar}}\right) & l_{ar} < \text{PCRight} < l_{br} \\ 1 & \text{PCRight} \geq l_{br} \\ 0 & \text{other} \end{cases} \end{aligned} \tag{28}$$

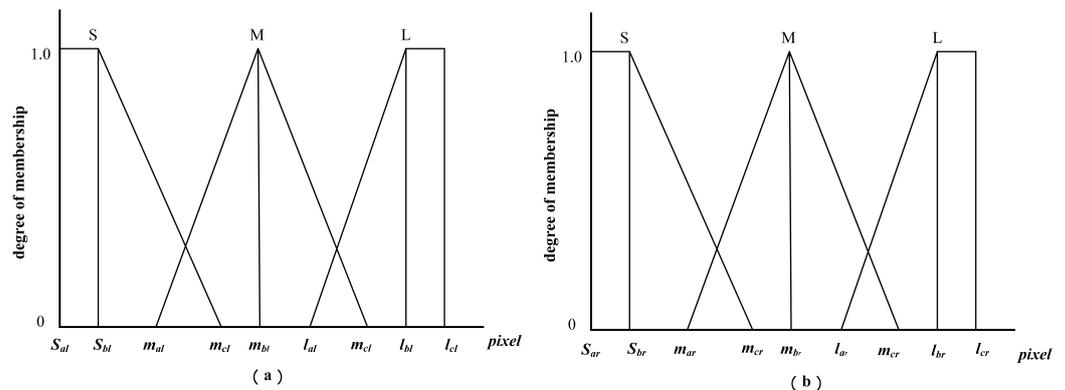


Figure 12. (a) PCLeft membership function; (b) PCRight membership function.

The fuzzy output of linear velocity is a standard value, which is multiplied by a suitable gain K . After fuzzy processing by the weighted average method, the output linear velocity instruction v_{vis} and the output steering angle instruction θ_{vis} can be calculated as below:

$$v_{vis} = K * \left(\frac{\sum_{i=1}^N v_{visi} * \alpha_i(x)}{\sum_{i=1}^N \alpha_i(x)} \right) \tag{29}$$

$$\theta_{vis} = \frac{\sum_{i=1}^N \theta_{visi} * \alpha_i(x)}{\sum_{i=1}^N \alpha_i(x)} \tag{30}$$

where $\alpha_i(x)$ represents the trigger strength.

It must be emphasized that the use of fuzzy logic control depends on the number of rules that must be evaluated. Moreover, large systems with many rules require very powerful and fast processors for real-time computation. The smaller the rule base, the less

computing power is required [49]. Therefore, unlike pure fuzzy logic controllers encountered in rule expansion problems, the visual pixel-based fuzzy logic control system designed in this paper has only nine if/then rules, and it uses trigonometric membership functions to make their structure simpler. The constructed fuzzy rule basis is shown in Table 3.

Table 3. The constructed fuzzy rule.

Number	PCLeft	PCRight	v_{vis}	$\theta_{vis}/^\circ$
1	Small	Small	0.9	90
2	Small	Medium	0.5	67
3	Small	Large	0.1	45
4	Medium	Small	0.5	112
5	Medium	Medium	0.9	90
6	Medium	Large	0.5	67
7	Large	Small	0.1	135
8	Large	Medium	0.5	112
9	Large	Large	0.9	60

5. Experimental Test Results of Autonomous Robot

5.1. System Composition

To verify the performance of the robot visual navigation and obstacle avoidance control system described in Sections 3 and 4, a Three-Wheeled Differential Mobile Robot (TWD MR) was used [50]. It is mainly composed of a vision camera, laser radar (RPLIDAR S2, measuring radius 30 m, frequency 32 K), a wheeled robot chassis, and ROS software control system. The visual camera is used for visual navigation, collecting environmental image information, data acquisition and environment modeling of obstacle avoidance systems with LiDAR. The wheeled robot chassis includes stepper motors on both sides, universal wheels, and uses differential control to change or maintain course. The ROS control system was used to verify the feasibility of the algorithm and integrated the visual navigation control algorithm based on image features and the obstacle avoidance control algorithm based on LiDAR data.

5.2. Test of Obstacle Avoidance System

Figure 13 shows the obstacle avoidance test scenario for an autonomous robot, which integrates the obstacle avoidance control algorithm developed in this paper to implement the obstacle avoidance function. To increase the complexity of the obstacle during the test, six plastic tubes with a diameter of 20 cm were placed at different positions and angles to act as obstacles. The autonomous robot was tested directly through six obstacles at different speeds of 0.25 m/s, 0.5 m/s, and 1 m/s. Following the theoretical analysis in Section 3, the central circle of the obstacle is enlarged using a modified clustering algorithm. The expanded part is shown in the gray part, and the theoretical obstacle avoidance trajectory of the robot when it passes the obstacle is shown in Figure 14a. Finally, real-time scanning of the surrounding environment information via LiDAR is used to locate and measure the robot's trajectory using the slam-map modeling method on top of the computer's Ubuntu system to obtain the robot's moving position and pose in real time. The results show that the robot can successfully pass through obstacles at three different speeds: 0.25 m/s, 0.5 m/s, and 1 m/s, and achieve obstacle avoidance function at different speeds. The actual obstacle avoidance trajectory of the robot when passing obstacles is shown in Figure 14b. This test preliminarily proves that the autonomous robot (TWD MR) designed in this paper is equipped with LiDAR and vision camera only (detailed parameters are introduced in Section 5.1), and the obstacle avoidance function is realized by using the developed algorithm. It does not require sophisticated GPS navigation and RTK localization modules, reflecting its convenience and low-cost advantages.

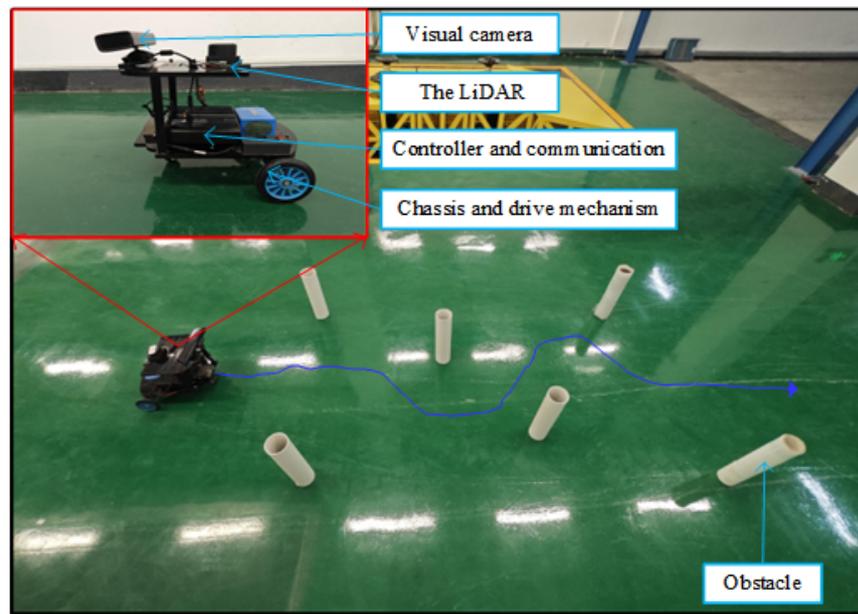


Figure 13. A diagram of the field test of the Three-Wheeled Mobile Differential Robot (TWMDR).

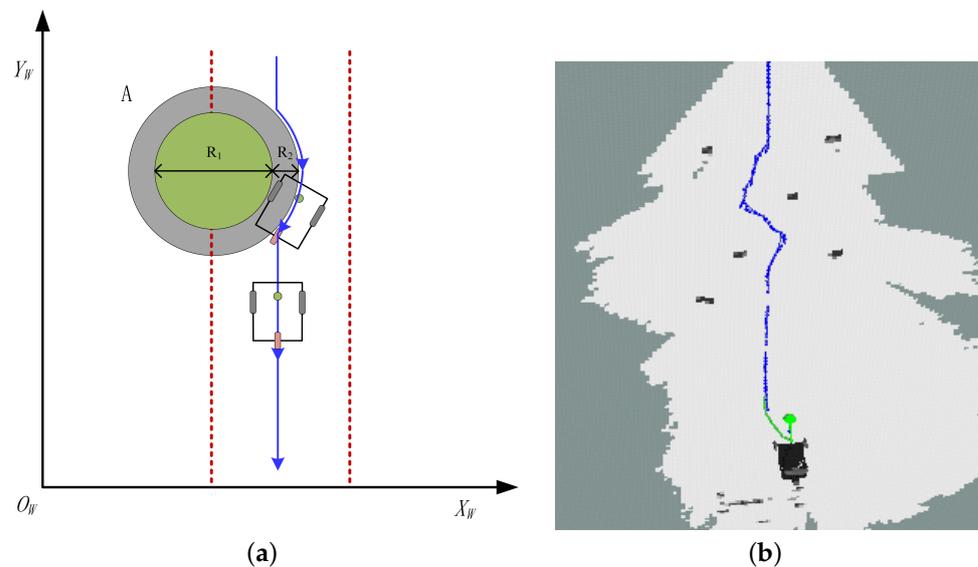


Figure 14. (a) The theoretical obstacle avoidance analysis diagram of robot; (b) the SLAM obstacle avoidance trajectory diagram of robot.

5.3. Test of Visual Guidance System

The experimental test is carried out in the robot (TWDMR), which uses a vision system to control the steering. The visual guidance trajectory and the operational specifications of the robot platform are shown in Table 4. The experimental testing procedure is based on a guided trajectory of first running straight, then turning, and finally running straight again. Figure 15 depicts the real-time variation in the left and right wheel speeds of the robot. Specifically, when the sampling number is 100, the robot starts to turn in the right corner; when the sampling number reaches 150, the right turn ends; when the sampling number reaches 250, the robot starts to turn left; when the sampling number reaches 290, the left turn ends. In the case of linear guidance trajectory, the deviation between the rotation speed of the left and right wheels of the robot platform is within 2.3 r/min.

Table 4. The operating parameters of the robot platform.

Sampling Rate ($S \cdot s^{-1}$)	The Length of Robot (/cm)	The Width of Robot (/cm)	The Width of Trajectory (/cm)	The Average Rotational Speed of Robot ($r \cdot \text{min}^{-1}$)
10	30	45	35	20

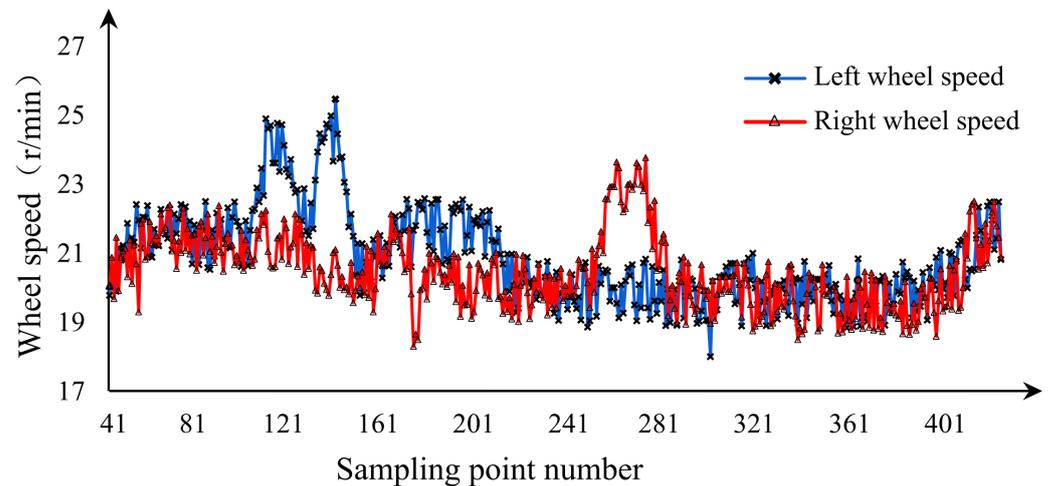
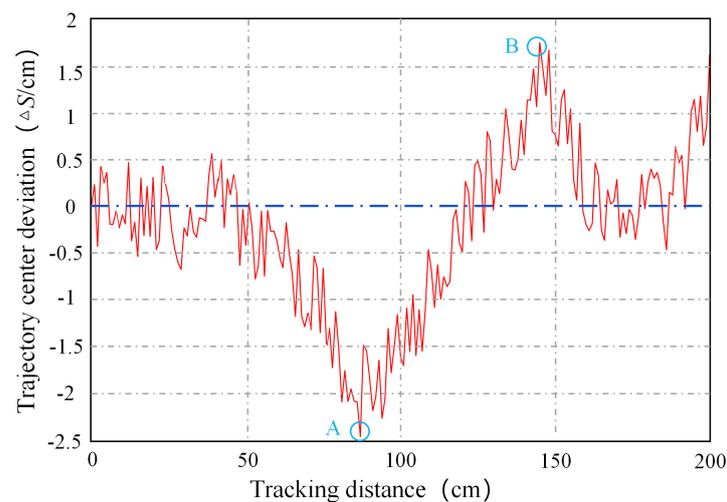
**Figure 15.** Real-time change in robot left and right wheel speed.

Figure 16 shows the test results of the robot trajectory tracking, including the transverse center deviation ΔS and angular deviation $\Delta\psi$, where the negative value represents the left deviation and the positive value is the right deviation. The test results show that the center deviation of the navigation track is within the range of -2.5 cm to 2 cm, the maximum deviation to the left of the track center is shown at point A in Figure 16, and the maximum deviation to the right is shown at point B in Figure 16. Meanwhile, the angle deviation is within the range of -0.5° to 0.4° , the maximum deviation to the left is shown at point D in Figure 17, and the maximum deviation to the right is shown at point C in Figure 17.

**Figure 16.** The deviation of the lateral center of the trajectory.

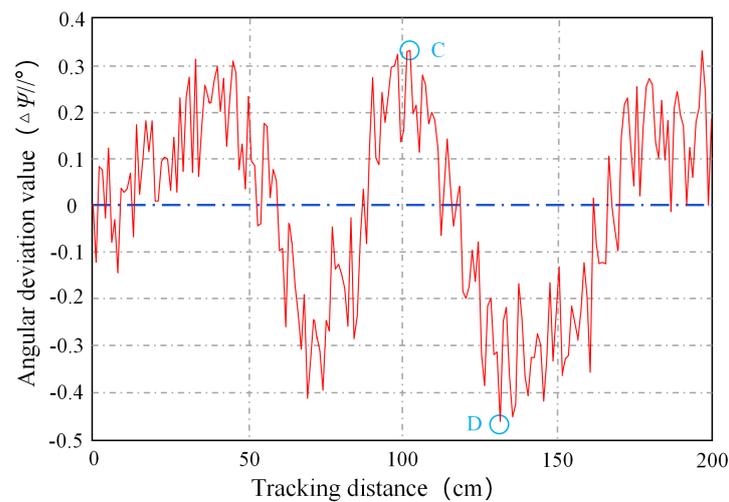


Figure 17. The deviation of trajectory angle.

6. Conclusions and Future Works

In this paper, we develop and design an agricultural mobile robot platform with visual navigation and obstacle avoidance capabilities to address the problem of low-cost obstacle detection for agricultural unmanned mobility platforms. As an obstacle detection sensor, LiDAR suffers from several drawbacks, such as a small amount of data, incomplete information about the detection environment, and redundant obstacle information during clustering. In this paper, we propose an improved clustering algorithm for obstacle avoidance systems to extract dynamic and static obstacle information and obtain the centers of obstacle scan points. Then, the maximum diameter of the convex polygon of the cluster data are obtained by fusion of the rotating jamming algorithm and the convex hull algorithm, and the obstacle danger zone is established with the cluster center as the center, and the VFH+ obstacle avoidance path planning and heading control method is proposed. Compared to the field linear navigation system built by Higuti et al. [18] in the same context using LiDAR only, our autonomous mobile robot is capable of turning and avoiding obstacles. On the other hand, the color-space analysis and feature analysis of complex orchard environment images were performed to obtain the H component of the HSV color space, which was used to obtain the ideal visual guidance trajectory images by means of mean-filtering and corrosion treatment. Finally, the effectiveness and robustness of the proposed algorithm are verified through the three-wheeled mobile differential robot platform. Compared with the proposed LiDAR–vision–inertial odometer fusion navigation method for agricultural vehicles by Zhao et al. [51], we can achieve obstacle avoidance and navigation functions for agricultural robots by using only LiDAR and camera. It embodies the advantages of simplicity, low cost, and efficiency, and is easier to meet the operational needs of the agricultural environment and farmers.

In this paper, the visual navigation and obstacle avoidance functions of agricultural mobile robots are realized based on the three-wheeled mobile differential robot platform. In contrast to the methods proposed by Ball and Winterhalter et al. [11,23], for agricultural navigation based on GNSS and GPS technology, the stable operation of agricultural robots working in environments rejected by GNSS is achieved. It is expected to shorten the development cycle of agricultural robot navigation systems when GNSS is not working, and has the advantages of low cost, simple operation, and good stability. In our future research work, we will transfer the proposed algorithm and hardware platform to agricultural driverless tractors for field trials and increase the construction of environmental maps for agricultural tractors.

Author Contributions: Conceptualization, C.H. and W.W.; Data curation, C.H.; Formal analysis, W.W. and X.L.; Investigation, X.L.; Project administration, J.L.; Resources, W.W.; Software, W.W.;

Validation, X.L.; Writing—original draft, C.H.; Writing—review and editing, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 62203176, the 2024 Basic and Applied Research Project of Guangzhou Science and Technology Plan under Grant SL2024A04j01318, and China Scholarship Council under Grant 202308440524.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

- Albiero, D.; Garcia, A.P.; Umez, C.K.; de Paulo, R.L. Swarm robots in mechanized agricultural operations: A review about challenges for research. *Comput. Electron. Agric.* **2022**, *193*, 106608. [\[CrossRef\]](#)
- Jin, Y.; Liu, J.; Xu, Z.; Yuan, S.; Li, P.; Wang, J. Development status and trend of agricultural robot technology. *Int. J. Agric. Biol. Eng.* **2021**, *14*, 1–19. [\[CrossRef\]](#)
- Li, J.; Li, J.; Zhao, X.; Su, X.; Wu, W. Lightweight detection networks for tea bud on complex agricultural environment via improved YOLO v4. *Comput. Electron. Agric.* **2023**, *211*, 107955. [\[CrossRef\]](#)
- Sparrow, R.; Howard, M. Robots in agriculture: Prospects, impacts, ethics, and policy. *Precis. Agric.* **2021**, *22*, 818–833. [\[CrossRef\]](#)
- Ju, C.; Kim, J.; Seol, J.; Son, H.I. A review on multirobot systems in agriculture. *Comput. Electron. Agric.* **2022**, *202*, 107336. [\[CrossRef\]](#)
- Bechar, A.; Vigneault, C. Agricultural robots for field operations: Concepts and components. *Biosyst. Eng.* **2016**, *149*, 94–111. [\[CrossRef\]](#)
- Liu, X.; Wang, J.; Li, J. URTSegNet: A real-time segmentation network of unstructured road at night based on thermal infrared images for autonomous robot system. *Control. Eng. Pract.* **2023**, *137*, 105560. [\[CrossRef\]](#)
- Dutta, A.; Roy, S.; Kreidl, O.P.; Bölöni, L. Multi-robot information gathering for precision agriculture: Current state, scope, and challenges. *IEEE Access* **2021**, *9*, 161416–161430. [\[CrossRef\]](#)
- Malavazi, F.B.; Guyonneau, R.; Fasquel, J.B.; Lagrange, S.; Mercier, F. LiDAR-only based navigation algorithm for an autonomous agricultural robot. *Comput. Electron. Agric.* **2018**, *154*, 71–79. [\[CrossRef\]](#)
- Rouveure, R.; Faure, P.; Monod, M.O. PELICAN: Panoramic millimeter-wave radar for perception in mobile robotics applications, Part 1: Principles of FMCW radar and of 2D image construction. *Robot. Auton. Syst.* **2016**, *81*, 1–16. [\[CrossRef\]](#)
- Ball, D.; Upcroft, B.; Wyeth, G.; Corke, P.; English, A.; Ross, P.; Patten, T.; Fitch, R.; Sukkarieh, S.; Bate, A. Vision-based obstacle detection and navigation for an agricultural robot. *J. Field Robot.* **2016**, *33*, 1107–1130. [\[CrossRef\]](#)
- Santos, L.C.; Santos, F.N.; Valente, A.; Sobreira, H.; Sarmiento, J.; Petry, M. Collision avoidance considering iterative Bézier based approach for steep slope terrains. *IEEE Access* **2022**, *10*, 25005–25015. [\[CrossRef\]](#)
- Monarca, D.; Rossi, P.; Alemanno, R.; Cossio, F.; Nepa, P.; Motroni, A.; Gabbriellini, R.; Pirozzi, M.; Console, C.; Cecchini, M. Autonomous Vehicles Management in Agriculture with Bluetooth Low Energy (BLE) and Passive Radio Frequency Identification (RFID) for Obstacle Avoidance. *Sustainability* **2022**, *14*, 9393. [\[CrossRef\]](#)
- Blok, P.; Suh, H.; van Boheemen, K.; Kim, H.J.; Gookhwan, K. Autonomous in-row navigation of an orchard robot with a 2D LIDAR scanner and particle filter with a laser-beam model. *J. Inst. Control. Robot. Syst.* **2018**, *24*, 726–735. [\[CrossRef\]](#)
- Gao, X.; Li, J.; Fan, L.; Zhou, Q.; Yin, K.; Wang, J.; Song, C.; Huang, L.; Wang, Z. Review of wheeled mobile robots' navigation problems and application prospects in agriculture. *IEEE Access* **2018**, *6*, 49248–49268. [\[CrossRef\]](#)
- Wang, T.; Chen, B.; Zhang, Z.; Li, H.; Zhang, M. Applications of machine vision in agricultural robot navigation: A review. *Comput. Electron. Agric.* **2022**, *198*, 107085. [\[CrossRef\]](#)
- Durand-Petiteville, A.; Le Flecher, E.; Cadenat, V.; Sentenac, T.; Vougioukas, S. Tree detection with low-cost three-dimensional sensors for autonomous navigation in orchards. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3876–3883. [\[CrossRef\]](#)
- Higuti, V.A.; Velasquez, A.E.; Magalhaes, D.V.; Becker, M.; Chowdhary, G. Under canopy light detection and ranging-based autonomous navigation. *J. Field Robot.* **2019**, *36*, 547–567. [\[CrossRef\]](#)
- Guyonneau, R.; Mercier, F.; Oliveira, G.F. LiDAR-Only Crop Navigation for Symmetrical Robot. *Sensors* **2022**, *22*, 8918. [\[CrossRef\]](#)
- Nehme, H.; Aubry, C.; Solatges, T.; Savatier, X.; Rossi, R.; Boutteau, R. Lidar-based structure tracking for agricultural robots: Application to autonomous navigation in vineyards. *J. Intell. Robot. Syst.* **2021**, *103*, 61. [\[CrossRef\]](#)
- Hiremath, S.A.; Van Der Heijden, G.W.; Van Evert, F.K.; Stein, A.; Ter Braak, C.J. Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter. *Comput. Electron. Agric.* **2014**, *100*, 41–50. [\[CrossRef\]](#)
- Zhao, R.M.; Zhu, Z.; Chen, J.N.; Yu, T.J.; Ma, J.J.; Fan, G.S.; Wu, M.; Huang, P.C. Rapid development methodology of agricultural robot navigation system working in GNSS-denied environment. *Adv. Manuf.* **2023**, *11*, 601–617. [\[CrossRef\]](#)
- Winterhalter, W.; Fleckenstein, F.; Dornhege, C.; Burgard, W. Localization for precision navigation in agricultural fields—Beyond crop row following. *J. Field Robot.* **2021**, *38*, 429–451. [\[CrossRef\]](#)
- Chen, W.; Sun, J.; Li, W.; Zhao, D. A real-time multi-constraints obstacle avoidance method using LiDAR. *J. Intell. Fuzzy Syst.* **2020**, *39*, 119–131. [\[CrossRef\]](#)

25. Lenac, K.; Kitanov, A.; Cupec, R.; Petrović, I. Fast planar surface 3D SLAM using LIDAR. *Robot. Auton. Syst.* **2017**, *92*, 197–220. [[CrossRef](#)]
26. Park, J.; Cho, N. Collision Avoidance of Hexacopter UAV Based on LiDAR Data in Dynamic Environment. *Remote Sens.* **2020**, *12*, 975. [[CrossRef](#)]
27. Li, J.; Wang, J.; Peng, H.; Hu, Y.; Su, H. Fuzzy-Torque Approximation-Enhanced Sliding Mode Control for Lateral Stability of Mobile Robot. *IEEE Trans. Syst. Man Cybern.* **2022**, *52*, 2491–2500.
28. Wang, D.; Chen, X.; Liu, J.; Liu, Z.; Zheng, F.; Zhao, L.; Li, J.; Mi, X. Fast Positioning Model and Systematic Error Calibration of Chang'E-3 Obstacle Avoidance Lidar for Soft Landing. *Sensors* **2022**, *22*, 7366. [[CrossRef](#)]
29. Kim, J.S.; Lee, D.H.; Kim, D.W.; Park, H.; Paik, K.J.; Kim, S. A numerical and experimental study on the obstacle collision avoidance system using a 2D LiDAR sensor for an autonomous surface vehicle. *Ocean Eng.* **2022**, *257*, 111508. [[CrossRef](#)]
30. Asvadi, A.; Premebida, C.; Peixoto, P.; Nunes, U. 3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes. *Robot. Auton. Syst.* **2016**, *83*, 299–311. [[CrossRef](#)]
31. Baek, J.; Noh, G.; Seo, J. Robotic Camera Calibration to Maintain Consistent Precision of 3D Trackers. *Int. J. Precis. Eng. Manuf.* **2021**, *22*, 1853–1860. [[CrossRef](#)]
32. Li, J.; Dai, Y.; Su, X.; Wu, W. Efficient Dual-Branch Bottleneck Networks of Semantic Segmentation Based on CCD Camera. *Remote Sens.* **2022**, *14*, 3925. [[CrossRef](#)]
33. Dong, H.; Weng, C.Y.; Guo, C.; Yu, H.; Chen, I.M. Real-time avoidance strategy of dynamic obstacles via half model-free detection and tracking with 2d lidar for mobile robots. *IEEE/ASME Trans. Mechatron.* **2020**, *26*, 2215–2225. [[CrossRef](#)]
34. Choi, Y.; Jimenez, H.; Mavris, D.N. Two-layer obstacle collision avoidance with machine learning for more energy-efficient unmanned aircraft trajectories. *Robot. Auton. Syst.* **2017**, *98*, 158–173. [[CrossRef](#)]
35. Gao, F.; Li, C.; Zhang, B. A dynamic clustering algorithm for LiDAR obstacle detection of autonomous driving system. *IEEE Sens. J.* **2021**, *21*, 25922–25930. [[CrossRef](#)]
36. Jin, X.Z.; Zhao, Y.X.; Wang, H.; Zhao, Z.; Dong, X.P. Adaptive fault-tolerant control of mobile robots with actuator faults and unknown parameters. *IET Control. Theory Appl.* **2019**, *13*, 1665–1672. [[CrossRef](#)]
37. Zhang, H.D.; Liu, S.B.; Lei, Q.J.; He, Y.; Yang, Y.; Bai, Y. Robot programming by demonstration: A novel system for robot trajectory programming based on robot operating system. *Adv. Manuf.* **2020**, *8*, 216–229. [[CrossRef](#)]
38. Raikwar, S.; Fehrmann, J.; Herlitzius, T. Navigation and control development for a four-wheel-steered mobile orchard robot using model-based design. *Comput. Electron. Agric.* **2022**, *202*, 107410. [[CrossRef](#)]
39. Gheisarnejad, M.; Khooban, M.H. Supervised Control Strategy in Trajectory Tracking for a Wheeled Mobile Robot. *IET Collab. Intell. Manuf.* **2019**, *1*, 3–9. [[CrossRef](#)]
40. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
41. Yang, C.; Chen, C.; Wang, N.; Ju, Z.; Fu, J.; Wang, M. Biologically inspired motion modeling and neural control for robot learning from demonstrations. *IEEE Trans. Cogn. Dev. Syst.* **2018**, *11*, 281–291.
42. Wu, J.; Wang, H.; Zhang, M.; Yu, Y. On obstacle avoidance path planning in unknown 3D environments: A fluid-based framework. *ISA Trans.* **2021**, *111*, 249–264. [[CrossRef](#)] [[PubMed](#)]
43. Liu, C.; Tomizuka, M. Real time trajectory optimization for nonlinear robotic systems: Relaxation and convexification. *Syst. Control. Lett.* **2017**, *108*, 56–63. [[CrossRef](#)]
44. Zhang, Q.; Chen, M.S.; Li, B. A visual navigation algorithm for paddy field weeding robot based on image understanding. *Comput. Electron. Agric.* **2017**, *143*, 66–78. [[CrossRef](#)]
45. Liu, Q.; Yang, F.; Pu, Y.; Zhang, M.; Pan, G. Segmentation of farmland obstacle images based on intuitionistic fuzzy divergence. *J. Intell. Fuzzy Syst.* **2016**, *31*, 163–172. [[CrossRef](#)]
46. Liu, M.; Ren, D.; Sun, H.; Yang, S.X.; Shao, P. Orchard Areas Segmentation in Remote Sensing Images via Class Feature Aggregate Discriminator. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [[CrossRef](#)]
47. Mills, L.; Flemmer, R.; Flemmer, C.; Bakker, H. Prediction of kiwifruit orchard characteristics from satellite images. *Precis. Agric.* **2019**, *20*, 911–925. [[CrossRef](#)]
48. Liu, S.; Wang, X.; Li, S.; Chen, X.; Zhang, X. Obstacle avoidance for orchard vehicle trinocular vision system based on coupling of geometric constraint and virtual force field method. *Expert Syst. Appl.* **2022**, *190*, 116216. [[CrossRef](#)]
49. Begnini, M.; Bertol, D.W.; Martins, N.A. A robust adaptive fuzzy variable structure tracking control for the wheeled mobile robot: Simulation and experimental results. *Control. Eng. Pract.* **2017**, *64*, 27–43. [[CrossRef](#)]
50. Zhang, F.; Zhang, W.; Luo, X.; Zhang, Z.; Lu, Y.; Wang, B. Developing an IoT-enabled cloud management platform for agricultural machinery equipped with automatic navigation systems. *Agriculture* **2022**, *12*, 310. [[CrossRef](#)]
51. Zhao, Z.; Zhang, Y.; Long, L.; Lu, Z.; Shi, J. Efficient and adaptive lidar visual inertial odometry for agricultural unmanned ground vehicle. *Int. J. Adv. Robot. Syst.* **2022**, *19*, 17298806221094925. [[CrossRef](#)]

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