agriculture

MDPI

Article

Information Perception Method for Fruit Trees Based on 2D LiDAR Sensor

Yong Wang, Changxing Geng *, Guofeng Zhu, Renyuan Shen, Haiyang Gu and Wanfu Liu

Robotics and Microsystems Centre, Soochow University, Suzhou 215000, China;
20195229071@stu.suda.edu.cn (Y.W.); 20195229074@stu.suda.edu.cn (G.Z.); 20205229012@stu.suda.edu.cn (R.S.); 20215229104@stu.suda.edu.cn (H.G.); 20215229086@stu.suda.edu.cn (W.L.)
* Correspondence: chxgeng@suda.edu.cn; Tel.: +86-186-6229-9889

Abstract: To solve the problem of orchard environmental perception, a 2D LiDAR sensor was used to scan fruit trees on both sides of a test platform to obtain their position. Firstly, the two-dimensional iterative closest point (2D-ICP) algorithm was used to obtain the complete point cloud data of fruit trees on both sides. Then, combining the lightning connection algorithm (LAPO) and the density-based clustering algorithm (DBSCAN), a fruit tree detection method based on density-based lightning connection clustering (LAPO-DBSCAN) was proposed. After obtaining the point cloud data of fruit trees on both sides of the test platform using the 2D-ICP algorithm, the LAPO-DBSCAN algorithm was used to obtain the position of fruit trees. The experimental results show that the positive detection rate was 96.69%, the false detection rate was 3.31%, and the average processing time was 1.14 s, verifying the reliability of the algorithm. Therefore, this algorithm can be used to accurately find the position of fruit trees, meaning that it can be applied to orchard navigation in a later stage.

Keywords: point cloud registration of fruit trees; lightning attachment procedure optimization; density-based spatial clustering of applications with noise; information perception of fruit trees

Citation: Wang, Y.; Geng, C.; Zhu, G.; Shen, R.; Gu, H.; Liu, W. Information Perception Method for Fruit Trees Based on 2D LiDAR Sensor. *Agriculture* **2022**, *12*, 914. https://doi.org/10.3390/agriculture12070914

Academic Editor: Francesco Marinello

Received: 14 April 2022
Accepted: 18 June 2022
Published: 23 June 2022

1. Introduction

To accelerate the development of smart agriculture, agricultural vehicle navigation technology has been developed rapidly. Agricultural machinery autonomous navigation systems based on machine vision, GPS, and LiDAR sensors have emerged [1]. Machine vision is greatly affected by the operating environment and lighting conditions. The application of GPS is affected by satellite signals. A LiDAR sensor can provide a large amount of accurate distance information at a higher frequency, reliably provide the position and depth information of surrounding objects [2], and provide more comprehensive information.

There are many ways to identify fruit trees in orchards. Judging from the existing research results, LiDAR sensors, cameras, or multisensor fusion can be used to detect fruit trees. Since the overall characteristics of trees are obvious, the trunks of fruit trees can be regarded as circles which can be detected by LiDAR sensors [3]. Due to the different installation methods and types of LiDAR sensors used, the data obtained are also different. (1) A LiDAR sensor can be installed vertically to extract the contour information of fruit trees [4,5]. Although this method can obtain the information of the trunks of fruit trees, as the LiDAR sensor is installed vertically, it can only extract the information of one tree at a time. This perception method is usually used to find the specific growth information of a fruit tree, such as fruit trees contour reconstruction. (2) A ground LiDAR sensor can be used to scan the environment to obtain fruit tree information [6,7]. (3) A mobile ground LiDAR sensor has been used to identify Fuji apples [8]. (4) An airborne LiDAR sensor has been used to obtain the scan data of fruit tree trunks [9,10].

Some scholars have also obtained tree information by analyzing LiDAR sensor data found from scanning. Using the same distance between the positions of fruit trees in an orchard, the data points in the arithmetic sequence of the concave points in the LiDAR sensor scan data can be extracted as the trunk points to obtain data [11]. LiDAR sensors can be used to scan woodland environments to obtain woodland data [12,13]. Since the data type of LiDAR sensors can be approximated by a point set, a clustering algorithm can be used to obtain fruit tree trunk information. Two-dimensional LiDAR sensors can be used to scan orchard environments and perform data clustering to extract the arc information of trunks [14,15]. Besides obtaining fruit tree information from clustering, 2D LiDAR sensors can be used to extract the central feature point data of tree trunks using the Euclidean clustering algorithm and the important geometric theorem of three-point collinearity [16]. Three-dimensional LiDAR sensor data are more abundant than 2D LiDAR sensor data, so many people use 3D LiDAR sensors for tree detection [17,18]. Although machine vision is greatly affected by the operating environment and lighting conditions, there have been many studies on the use of cameras for fruit tree inspection in orchards [19]. Due to the complex environment of orchards, a variety of sensor fusion methods can be used for research [20–23].

In previous studies, various sensors have been used to obtain orchard environmental information for orchard intelligent equipment. Usually, the information of fruit trees is used to pave the way for the application of intelligent equipment in orchard navigation.

The main purpose of this article is to obtain the position information of fruit trees using a 2D LiDAR sensor. After obtaining the position information of fruit trees with the algorithm proposed in this paper, it can be used for positioning, fitting navigation lines, and the navigation of orchard intelligent equipment in later stages. For the complex environment of orchards, this environment perception method is studied. Firstly, a fruit tree information acquisition method based on 2D-ICP is proposed. After the iterative registration of the point cloud data of both sides of fruit trees obtained by the 2D LiDAR sensor, the point cloud data of each fruit tree in the orchard are obtained. Then, by improving the LAPO and DBSCAN algorithms, a new method based on LAPO-DBSCAN is used to obtain the position of each fruit tree and realize their detection. Finally, the accuracy of the algorithm is verified by a field test.

2. Materials and Methods

2.1. Experimental Equipment

In this research, a differential test platform with a maximum speed of 1 m/s was built, as shown in Figure 1. The LiDAR sensor scans the surrounding data in real time, and the obtained LiDAR sensor data are transmitted to the industrial computer. The industrial computer runs a self-made software system to analyze the LiDAR sensor data. The LiDAR sensor is Rashen N30103B and it adopts the horizontal installation method, which is located in the front and middle of the orchard transportation robot. The installation height is 0.65 m and the parameters are shown in Table 1.

Figure 1. Test platform.

Table 1. Two-dimensional LiDAR sensor parameters.

2D LiDAR Sensor Specifications	Parameter Index
Detection range (m)	30
Ranging accuracy (mm)	±30
Scanning angle (°)	360
Angle resolution (°)	0.18
Scanning frequency (Hz)	10

2.2. Fruit Tree Information Perception Method

In this section, we introduce the complete method of fruit tree information perception; the specific process is shown in Figure 2. When the test platform runs in the orchard, the surrounding fruit tree information can be obtained by 2D LiDAR sensor scanning, and the point cloud data of the fruit tree will be preprocessed. Preprocessing is used so as to only retain fruit trees on both sides of the test platform for point cloud registration and clustering. The most important is point cloud registration and clustering. Firstly, complete fruit tree data on both sides can be obtained by point cloud registration. Then, the position of each fruit tree can be obtained by a clustering algorithm.

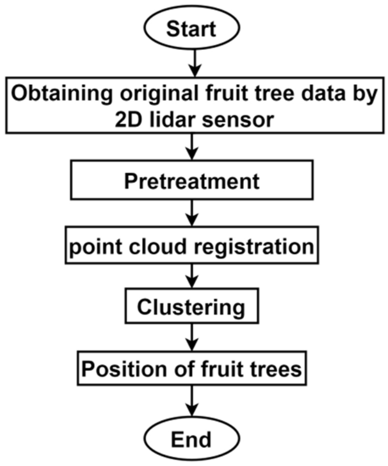


Figure 2. The process of fruit tree perception.

2.3. Fruit Tree Point Cloud Data Collection Method Based on 2D-ICP

With the increase in distance, the amount of LiDAR sensor data collected will become less and less. At the same time, in the process of acquisition, the fruit tree rows on both sides of the non-test platform will be blocked and the information will be incomplete, resulting in a poor iterative effect. Therefore, for the preprocessing of the collected LiDAR point cloud data, only the fruit tree row data on both sides of the test platform will be retained. After obtaining the LiDAR sensor data of fruit trees on both sides of the test platform, the 2D-ICP algorithm can be used for registration. The specific algorithm steps are as follows. The preprocessed target point cloud and source point cloud are $P^k = \left\{P_i^k\right\}$ and $P^{k+1} = \left\{P_i^{k+1}\right\}$, $i = 1, 2, 3 \ldots n$. According to the 2D-ICP algorithm, Equation (1) gives the objective function.

$$Dist(R,T)_{min} = \frac{1}{n}\sum_{i=1}^{n}\left|P_i^k - \left(RP_i^{k+1} + T\right)\right|^2 \tag{1}$$

There are two variables in Equation (1) that can be regularized by considering only the rotation matrix R—that is, the centers of two frame point clouds are $C_{P^k} = \frac{1}{n}\sum_{i=1}^{n} P_i^k$ and $C_{P^{k+1}} = \frac{1}{n}\sum_{i=1}^{n} P_i^{k+1}$, where $P_i^k = P_i^k - C_{P^k}$ and $P_i^{k+1} = P_i^{k+1} - C_{P^{k+1}}$.

Then, Equation (1) becomes Equation (2).

$$Dist(R,T)_{min} = \frac{1}{n}\sum_{i=1}^{n}\left|P_i^k - RP_i^{k+1}\right|^2 \tag{2}$$

Decomposing Equation (2) into Equation (3) gives us:

$$Dist(R,T)_{min} = \frac{1}{n}\left(\sum_{i=1}^{n}\left|P_i^k\right|^2 + \sum_{i=1}^{n}\left|RP_i^{k+1}\right|^2 - 2\sum_{i=1}^{n}\left|P_i^k RP_i^{k+1}\right|^2\right) \tag{3}$$

If the objective function is minimized, Equation (4) is maximized:

$$F(R)_{max} = \frac{1}{n}\sum_{i=1}^{n}\left|P_i^k RP_i^{k+1}\right|^2 \tag{4}$$

Since the fruit tree data are two-dimensional, the rotation matrix R is valued as $\begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$, where θ is the rotation angle between the two frame point clouds, which is substituted into Equation (4) to obtain Equation (5).

$$F(R)_{max} = \frac{1}{n}\sum_{i=1}^{n}\left|\begin{bmatrix} Px_i^k & Py_i^k \end{bmatrix}\begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}\begin{bmatrix} Px_i^{k+1} \\ Py_i^{k+1} \end{bmatrix}\right|^2 \tag{5}$$

The derivation and extreme value of θ are used to deduce Equation (6).

$$\frac{sin\theta}{cos\theta} = sin\theta\frac{1}{n}\sum_{i=1}^{n}\left(\frac{Py_i^k \times Px_i^{k+1} - Px_i^k \times Py_i^{k+1}}{Px_i^k \times Px_i^{k+1} + Py_i^k \times Py_i^{k+1}}\right) \tag{6}$$

After θ is calculated, R can be obtained. T can be obtained by Equation (7), and then iterated until the threshold is satisfied.

$$T = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} C_{pk}^x \\ C_{pk}^y \end{bmatrix} - R\begin{bmatrix} C_{pk+1}^x \\ C_{pk+1}^y \end{bmatrix} \tag{7}$$

2.4. Fruit Tree Position Detection Algorithm

2.4.1. Introduction to LAPO and DBSCAN Algorithms

The LAPO algorithm [24] has four important stages, including the cloud surface penetrating the air phase, the lightning channel moving downward, the upward pilot starting to spread from the ground (or grounded object), and the last fight back stage. The LAPO algorithm has a strong optimization ability in many engineering problems, and no additional parameters need to be set, which can help to avoid subjective factors influencing the results of the algorithm. Due to the influence of randomness and other factors, the standard LAPO algorithm may also fall into a local optimum, which makes it impossible to obtain a better solution every time. There is room for further improvement in its stability. Therefore, this algorithm has also been applied and improved in various clustering algorithms [25,26].

The DBSCAN algorithm [27] is based on a certain distance measurement criterion, which clusters closely related data points based on their criteria into one category. The following two parameters are set before clustering. The first one is *Eps* (the radius of the given object is the neighborhood). The second one is *MinPts* (the minimum number of components that make up a class). The traditional DBSCAN clustering algorithm is affected by *Eps* and *MinPts*. These two parameters are global and fixed so that only the data in the data set that meet the threshold condition can be effectively clustered, meaning that data of other densities may be treated as noise. In addition, the traditional DBSCAN algorithm needs to traverse each data point. When the data scale is large, the algorithm execution efficiency will be low, and the processing time will be long, which is not conducive to the realization of the algorithm. In view of the shortcomings of the traditional DBSCAN algorithm, our predecessors in this area have carried out a considerable amount of research and improved the DBSCAN algorithm [28–30].

2.4.2. Fruit Tree Detection Algorithm Based on LAPO-DBSCAN

Due to the shortcomings of the two algorithms, this paper proposes a fruit tree position detection algorithm based on LAPO-DBSCAN. This algorithm is mainly used to obtain the position of fruit trees. This process includes preparation and detailed steps, and its flow chart is shown in Figure 3.

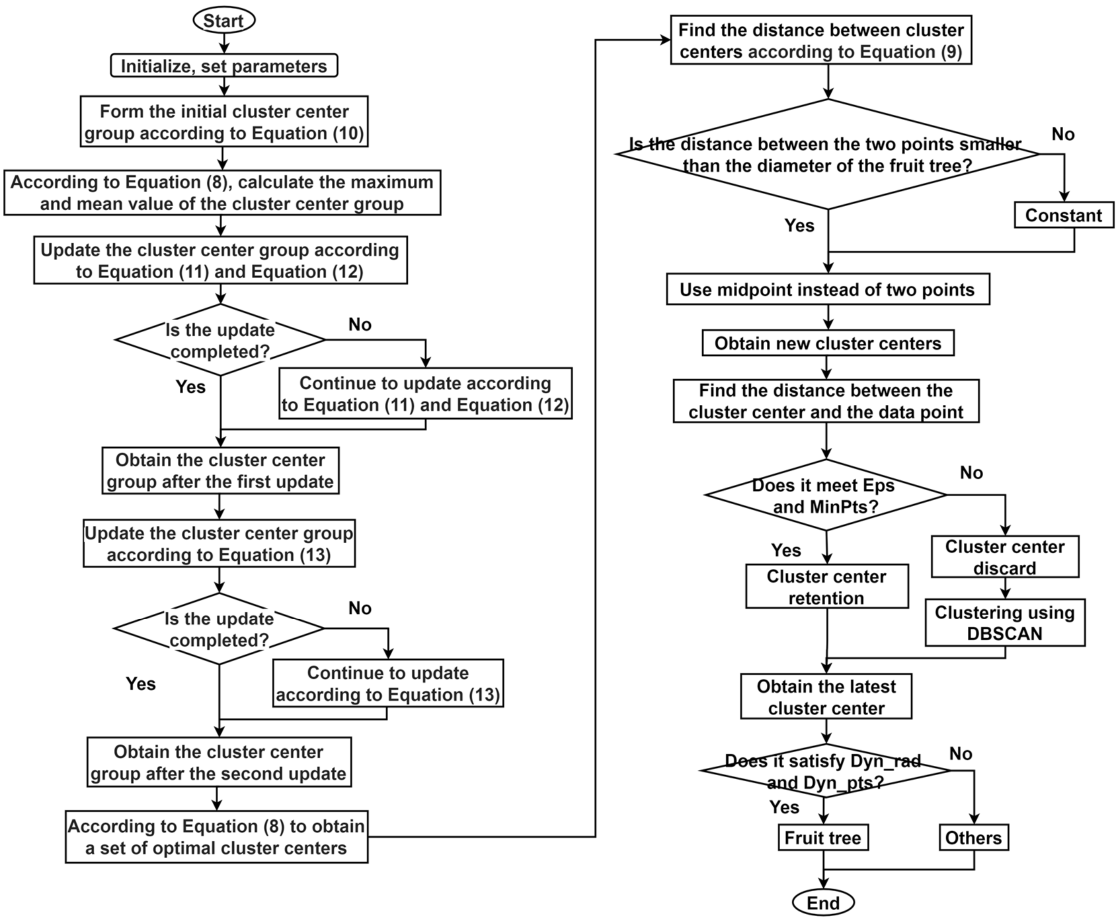


Figure 3. Fruit tree detection based on LAPO-DBSCAN.

A. Preparation.

Parameter settings. 1. Setting the parameters of the LAPO algorithm. (1) n groups of initial clustering centers are used as an initial clustering center group for each group of k clustering centers (k is set randomly). (2) t is the number of iterations. (3) t_{max} is the maximum number of iterations. 2. Setting the DBSCAN algorithm parameters. (1) The traditional DBSCAN algorithm: Eps and $MinPts$ are set according to the actual situation. (2) The dynamic DBSCAN algorithm: the radius range of the dynamic fruit tree is set according to the radius of the fruit tree, which is $Dyn\_rad \in [rad_{min}, rad_{max}]$. According to the LiDAR sensor parameters and distance, different neighborhood density thresholds are obtained as $Dyn\_pts \in [Pts\_num_{min}, Pts\_num_{max}]$.

Setting various functions. 1. Equation (8) is the objective function.

$$F_{min} = \sum_{i=1}^{n}\sum_{j=1}^{k} v_{i,j} \parallel x_i - c_j \parallel^2 \tag{8}$$

where $v_{i,j} = \begin{cases} 1\ if\ arg\ min_j dis(x_i, c_j) \\ 0\ else \end{cases}$ is the confidence function, which means that each datum in the data set can only belong to one class.

2. Equation (9) is the Euclidean distance between two points, where $i \neq j$.

$$dis(X_i, X_j) = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{k}(X_i - X_j)^2} \tag{9}$$

B. Detailed steps.

1. Initializing the LiDAR data and setting the corresponding parameters.

2. Randomly selecting an initial cluster center group (n groups of cluster centers for each group of k) to form a matrix C of n rows and k columns:

$$C = \begin{bmatrix} \{c_{1,1}, c_{1,2}, \cdots c_{1,k-1}, c_{1,k}\} \\ \{c_{2,1}, c_{2,2}, \cdots c_{2,k-1}, c_{2,k}\} \\ \vdots \\ \{c_{n-1,1}, c_{n-1,2}, \cdots c_{n-1,k-1}, c_{n-1,k}\} \\ \{c_{n,1}, c_{n,2}, \cdots c_{n,k-1}, c_{n,k}\} \end{bmatrix}.$$

The cluster center of the row is expressed by Equation (10).

$$C_i = Data_{min} + rand \times (Data_{max} - Data_{min}) \tag{10}$$

where $rand$ is a random number in the range [0, 1]. $Data_{max}$ and $Data_{min}$ are the maximum and minimum values of the radar data, respectively.

3. According to Equation (8), the fitness of cluster center group (C) is calculated. The optimal value, worst value, and average value of the cluster center group are C_{best}, C_{low}, and C_{ave}, respectively. According to the objective function, if $F_{low} > F_{ave}$, we can assign the value of C_{ave} to C_{low}.

4. If the cluster center (C_i) of a certain row is updated, a group of cluster centers (C_j) is randomly selected from the population, where $i \neq j$. If $F_{ave} > F_j$, Equation (11) can be used to iterate.

$$C_i = C_i + rand \times (C_{ave} - rand \times (C_i)) \tag{11}$$

If $F_j > F_{ave}$, Equation (12) can be used to iterate.

$$C_i = C_i - rand \times (C_{ave} - rand \times (C_i)) \tag{12}$$

After the above process is complete, the updated cluster center group (C_{new}) can be obtained.

5. Return to step (3). Updating the cluster center group (C_{new}) and obtaining the optimal value, the worst value and mean value of the cluster center group will be C_{new_best}, C_{new_low}, and C_{new_ave}, respectively.

6. C_{new} can be iterated through Equation (13) to obtain a new cluster center group C_{new}.

$$C_{new} = C_{new} + rand \times S \times (C_{new\_ave} + rand \times (C_{new\_low} - C_{new\_best})) \tag{13}$$

where $S = 1 - \left(\frac{t}{t_{max}}\right) \times exp\left(-\frac{t}{t_{max}}\right)$.

7. Return to step (3). The cluster center group (C_{new}) needs to be updated and the optimal value (C_{new_best}), worst value (C_{new_low}), and mean value (C_{new_ave}) of the cluster center group can be obtained.

8. With step (2) to step (7), the optimal cluster center group (C_{best}) can be obtained and the fitness can be calculated according to the objective function to obtain a set of optimal cluster centers (c_{best}), namely, $\{c_{1\_best}, c_{2\_best}, \cdots c_{k-1\_best}, c_{k\_best}\}$.

9. According to the obtained cluster centers, Equation (9) can be used to find the distance between each cluster center. When the distance between two points is less than the diameter of the fruit tree (the diameter of the fruit tree at the height of the radar installation), the distance between the two points is used. The two cluster centers can be replaced with the midpoint (when encountering bifurcated fruit trees, the midpoint is also used to replace the two points). Otherwise, the cluster center remains unchanged. Finally, the updated cluster center (c_{best}) can be obtained.

10. Equation (9) can be used to calculate the distance from each point (x_i) to the cluster center (c_{best}) in the data set and dividing each data point into each cluster center according to the confidence function, which is $Dist(x_i, c_{n\_best}) = \sqrt{(x_i - c_{n\_best})^2}$.

11. According to the actual situation, when the *Eps* and *MinPts* conditions are met, the cluster center is retained; otherwise, the cluster center is discarded.

12. After discarding some of the cluster centers that do not meet the criteria, according to the DBSCAN algorithm (still must meet the *Eps* and *MinPts* conditions), clustering is performed to obtain the corresponding cluster, and the mean value of the corresponding cluster is used to represent the cluster center (c_i).

13. When the distance between two cluster centers is less than the diameter of the fruit tree (the diameter of the fruit tree at the radar installation position), return to step (9). Otherwise, the final cluster center (c_{final}) should be obtained to form a cluster.

14. According to the final cluster center (c_{final}), the following two conditions need to be met to determine whether something is a fruit tree. Firstly, if the fruit tree dynamic radius (*Dyn_rad*), which is $rad_{min} \leq Dist(X_i, C_{n\_best}) \leq rad_{max}$, is satisfied, an object is a fruit tree. At the same time, if it meets the dynamic neighborhood density threshold ($Dyn\_pts \in [Pts\_num_{min}, Pts\_num_{max}]$), the object is a fruit tree.

15. Finally, the position of each fruit tree can be detected.

2.4.3. Algorithm Improvement

The algorithm solves the problem of local optimal solutions in the LAPO algorithm and parameter globality in the DBSCAN algorithm. In the LAPO algorithm, *n* initialized cluster centers are randomly selected, as shown in step (2). As the selected *n* cluster centers are random, there will be local optimal solutions in the calculation process for LAPO (from step (3) to step (8)). In the actual test, the local optimal solution may appear in the following situations. The results of the algorithm identify a set of clustering centers, but there is obvious deviation (the obvious deviation here refers to the situation where the clustering center is not on the data point and cannot correspond to the relevant data point) and two clustering centers are together. At the same time, the LAPO algorithm also has the problem of missing perception. To solve the above problems, the LAPO algorithm is improved. When the two clustering centers are together, the two clustering centers can be combined into one by taking the midpoint in step (9). In the cases of obvious deviation and missing detection, the final cluster center can be obtained by the DBSCAN algorithm (from step (10) to step (12)), but fruit trees may still not be detected. As the DBSCAN algorithm is affected by *Eps* and *MinPts*, in the step (14) *Eps* and *MinPts* are used to distinguish fruit trees by the dynamic threshold method. Through three-layer detection, the positive detection rate of fruit tree detection can be greatly improved.

2.4.4. Simulation Data Verification

Both the LAPO and DBSCAN algorithms have shortcomings, so they need to be improved to adapt them to more scenes. Using the 2D LiDAR sensor to scan the contours of fruit trees at different heights, the point cloud data will include two kinds of point cloud data, which are the point cloud data of the main tree trunk and the point cloud data of the canopy. Usually, the more data there are, the better the algorithm will be. To verify that the LAPO-DBSCAN algorithm used in this paper is better than the LAPO and DBSCAN algorithms, simulation data similar to the trunk of fruit trees are used for verification according to the point cloud data of the trunks of fruit trees scanned by 2D LiDAR sensors, as shown in Figure 4.

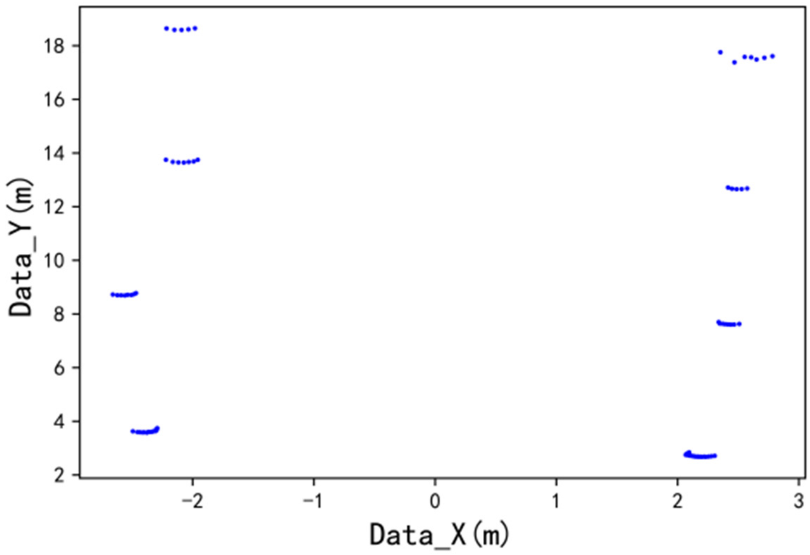


Figure 4. Simulation data.

To prove that the method based on the LAPO-DBSCAN algorithm is able to detect more characteristic information than the method based on the LAPO algorithm, this paper uses simulated fruit tree data to test the above two methods for one hundred frames. The results obtained from the method based on the LAPO-DBSCAN algorithm and the method based on the LAPO algorithm are shown in Table 2. The simulation data used in this paper are similar to the trunk of fruit trees. Scholars [30] have used the improved DBSCAN algorithm to detect the trunk of fruit trees, and the accuracy can reach 95.5%. Compared with previous algorithms, this algorithm increases the accuracy by 3.92%. Therefore, the algorithm used in this paper will no longer be compared with the DBSCAN algorithm for detecting the trunk of fruit trees.

Table 2. Actual scene test results.

Algorithm Type	Times	Results (%)		Average Handling Time (s)
LAPO	100	Positive detection rate False detection rate	97.00% 3.00%	0.41
LAPO-DBSCAN	100	Positive detection rate False detection rate	99.42% 0.58%	0.07

In Table 2, the positive detection rate of the LAPO-DBSCAN algorithm is better than that of the LAPO algorithm, and the detection result is 2.42% higher. In terms of the false detection rate, the LAPO-DBSCAN algorithm is better than the LAPO algorithm, and the difference between the detection results is 2.42%. In terms of the average processing time, the LAPO-DBSCAN algorithm consumes 82.92% less time than the LAPO algorithm. Therefore, the simulation results show that the LAPO-DBSCAN algorithm is superior to the LAPO algorithm and has a better detection effect.

3. Results

3.1. Experimental Scene

Due to the different planting mode and row spacing of each fruit tree, the data obtained by 2D LiDAR sensor scans of different fruit trees are also different, which will affect the accuracy and stability of the algorithm. The test site used in this paper was selected from the orchard of Nijiawan water field in Xiangcheng District of Suzhou, as shown in Figure 1. The distance from the ground to the main trunk of the fruit tree selected in this paper was about 0.5 m, and the area above 0.5 m was the canopy. According to the installation height of the 2D LiDAR sensor, the collected point cloud data were all the point cloud data of fruit tree crowns. As shown in Table 3, data on two rows of fruit trees used in the experiment were obtained.

Table 3. Fruit tree data.

Fruit Tree Information							
The outline length of the fruit tree on the left (m)	3.83	5.03	4.08	2.82	2.67	2.98	3.92
The outline length of the fruit tree on the right (m)	2.83	4.40	4.24	2.98	3.61	3.30	4.46
Distance between left and right fruit trees (m)				4			
Distance between adjacent fruit trees on the same side (m)				3			

3.2. Algorithm Verification

In the previous chapter, we introduced the method of fruit tree information perception. Next, the point cloud registration based on the 2D-ICP algorithm and the fruit tree position detection based on the LAPO-DBSCAN algorithm are tested.

3.2.1. Experiment on Fruit Tree Information Acquisition

Firstly, the test platform is controlled to drive slowly from the beginning of the fruit tree to the end of the fruit tree to collect the point cloud data of the fruit trees. Then, all the initial point cloud data of the fruit trees are preprocessed, and the final result only retains the nearest point cloud data of the fruit tree on both sides of the test platform. The original point cloud data, as shown in Figure 5, show that the data in the untreated orchard have many interference points, such as the data collected from non-bilateral fruit trees and the "zero-points" generated by the 2D LiDAR sensor at a certain angle. During the preprocessing, we keep the data in the red box in Figure 5. The preprocessed data are shown in Figure 6.

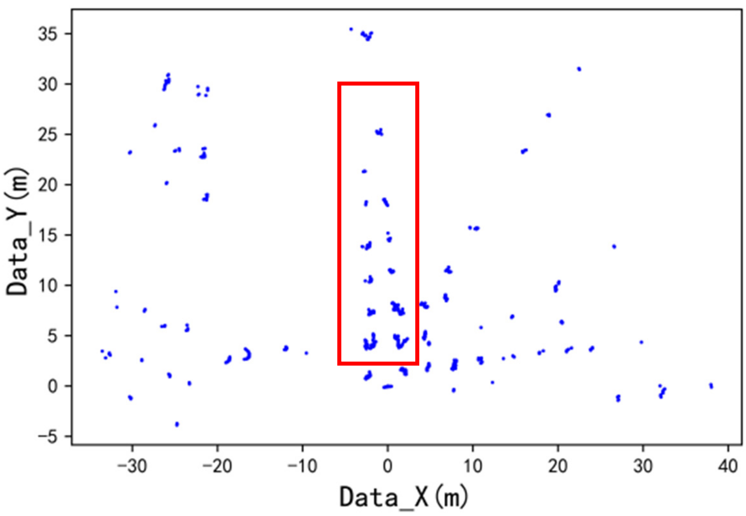


Figure 5. A frame of original data.

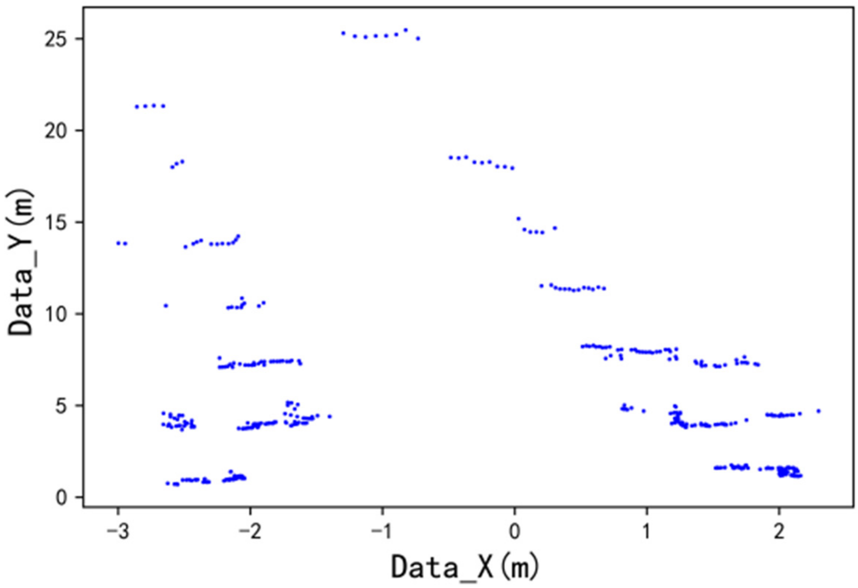


Figure 6. A frame of preprocessed data.

After preprocessing, registration is performed based on the 2D-ICP algorithm to obtain the completed fruit tree row point cloud data. This paper takes the registration of two frame point cloud data as an example to illustrate the registration process of fruit tree point cloud data. The registration process is shown in Figure 7. In Figure 7a–e, the iterative registration processes of two point clouds are shown. The R and T from Figure 7a–e are calculated using Equations (6) and (7). The R and T are $\begin{bmatrix} 0.999948 & -0.010145 \\ 0.010145 & 0.999948 \end{bmatrix}$ and $\begin{bmatrix} 0.027403 & 0.139726 \end{bmatrix}$ in Figure 7a. The R and T are $\begin{bmatrix} 0.999992 & -0.003975 \\ 0.003975 & 0.999992 \end{bmatrix}$ and $\begin{bmatrix} 0.010213 & 0.062397 \end{bmatrix}$ in Figure 7b. The R and T are $\begin{bmatrix} 0.999997 & -0.002030 \\ 0.002030 & 0.999997 \end{bmatrix}$ and $\begin{bmatrix} 0.007544 & 0.030963 \end{bmatrix}$ in Figure 7c. The R and T are $\begin{bmatrix} 0.999999 & -0.000753 \\ 0.000753 & 0.9999996 \end{bmatrix}$ and $\begin{bmatrix} 0.001915 & 0.018437 \end{bmatrix}$ in Figure 7d. The R and T are $\begin{bmatrix} 0.999999 & -0.000435 \\ 0.000435 & 0.999999 \end{bmatrix}$ and $\begin{bmatrix} 0.002288 & 0.000870 \end{bmatrix}$ in Figure 7e. Figure 7f is the result of using a two-frame point cloud registration as the next target point cloud. All the preprocessed point cloud data of fruit trees on both sides of the test platform can be iteratively registered by the 2D-ICP algorithm to obtain complete the point cloud data of fruit trees, including the point cloud information of each fruit tree position, as shown in Figure 8. The fruit trees on both sides of the test platform in Figure 8a show the point cloud data in Figure 8b.

When using the 2D-ICP algorithm to construct point cloud data of fruit trees on both sides of the test platform, we need to consider two problems. The first question is how to obtain as much complete point cloud data as possible. The second problem is how to prevent the oscillation of the collected point cloud data due to the uneven ground of the orchard during movement. Therefore, on the one hand, we move the test platform as slowly as possible, so the vibration amplitude of the vehicle is not large in the process of moving. On the other hand, we choose a 2D LiDAR sensor with a high frequency, as shown in Table 1, to obtain more fruit point cloud data within a short period of time. Through these two measures, the adverse effects caused by the vibration of the test platform can be compensated for to a certain extent, and the obtained fruit tree point cloud information can be enriched, which is conducive to obtaining better point cloud registration results.

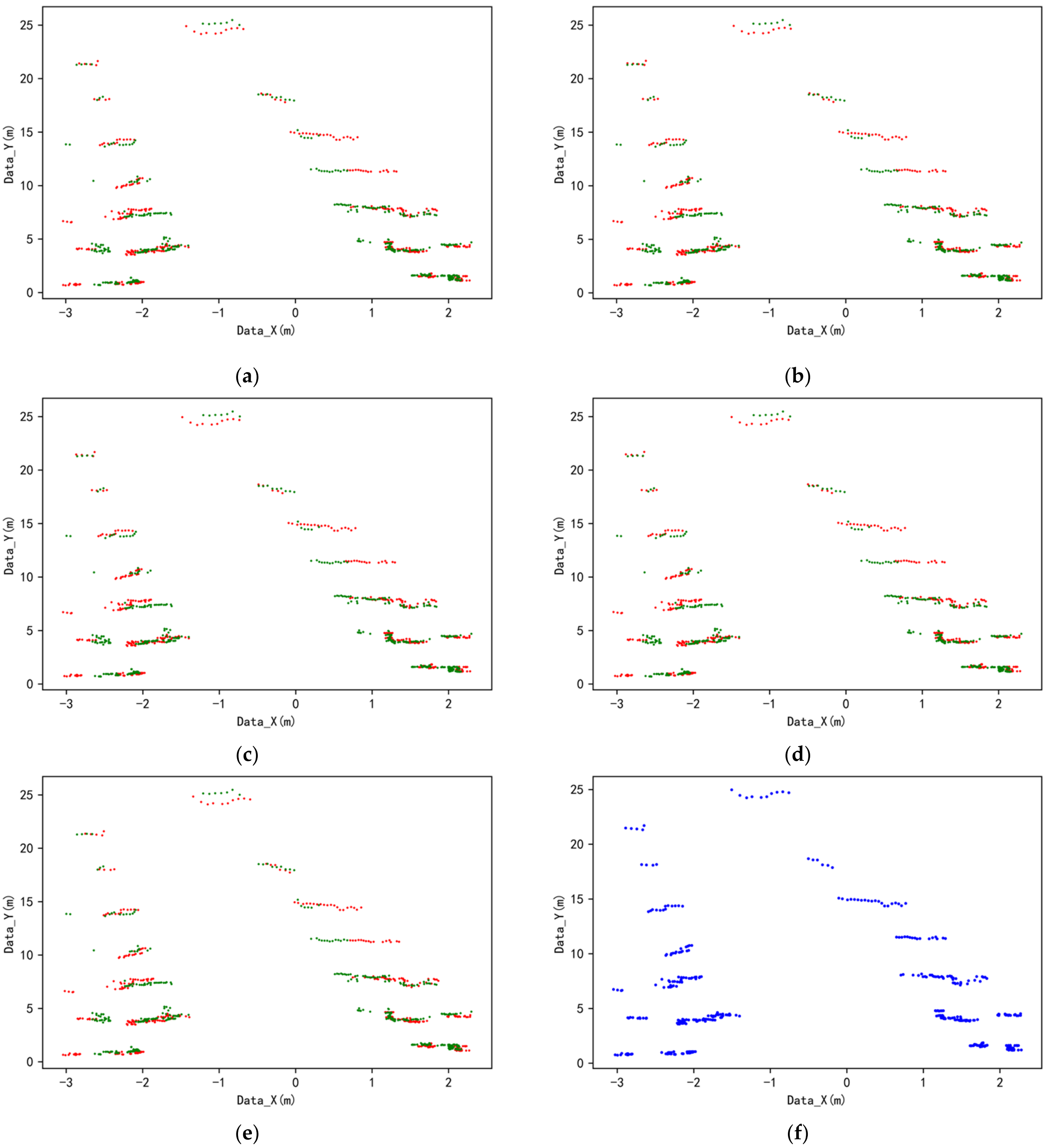
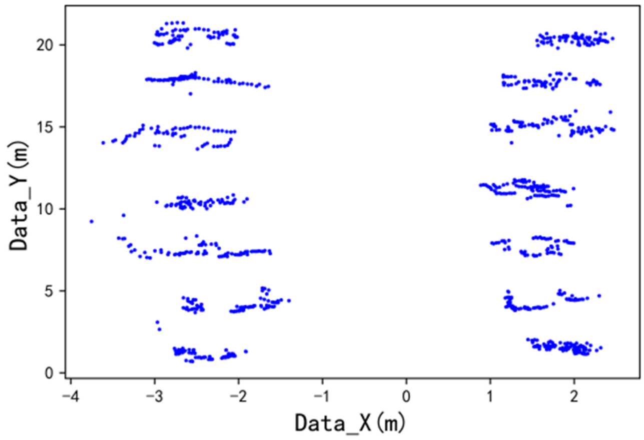


Figure 7. The 2D-ICP algorithm registration process. (**a**) First point cloud registration; (**b**) second point cloud registration; (**c**) third point cloud registration; (**d**) fourth point cloud registration; (**e**) last point cloud registration; (**f**) results after two-frame point cloud alignment.

(**a**)



(**b**)

Figure 8. Complete fruit tree row point cloud data. (**a**) Fruit trees on both sides of the test platform in the actual scene; (**b**) point cloud data corresponding to fruit trees on both sides of the test platform.

3.2.2. Experiment on Fruit Tree Position Acquisition

After obtaining the point cloud data of the fruit trees on both sides of the test platform, we need to obtain the position of the fruit trees. The first task is to set relevant parameters. The number of initial cluster centers are set to four groups with six cluster centers in each group, and the maximum number of iterations is 10. Although the above parameters are set randomly, they should also be designed according to the actual situation. The value of the parameters should not be too large or too small. In the traditional DBSCAN algorithm, *Eps* is set to 0.58 m and *MinPts* is set to 3. *Eps* depends on the average radius of fruit trees on both sides. In the dynamic DBSCAN algorithm, *Dyn_rad* depends on the minimum radius and maximum radius of fruit trees on both sides—namely, $Dyn\_rad \in [0.42,\ 0.80]$. *Dyn_pts* depends on the quotient of the number of point clouds per frame of the 2D LiDAR sensor and the number of cluster centers in each group, namely, $Dyn\_pts \in [3,\ 333]$. The radius of fruit trees depends on their contours, as shown in Table 3. We regard the outline of the fruit tree as a circle and use the formula of the circumference of a circle to calculate the radius of the fruit tree.

The implementation process based on the LAPO-DBSCAN algorithm is as follows. According to the LAPO-DBSCAN algorithm, a set of clustering centers are obtained from step (3) to step (8). In three situations of the missing detection of some fruit trees, two clustering centers appear on one fruit tree and the clustering center obviously deviates from the fruit tree data. Here, we take two clustering centers together as examples, as shown in Figure 9a. Then, we obtain the complete cluster center through step (9) to step (13). Finally, the test results of fruit trees are obtained through step (14), as shown in Figure 9b, where the blue point is the data point and the red "×" is the clustering center. The corresponding cluster center coordinates are shown in Table 4. According to Figure 8b and Table 4, this algorithm can accurately detect the position of fruit trees.

Table 4. Two-dimensional LiDAR sensor parameters.

The Coordinates of the Left Fruit Tree (m)	The Coordinates of the Right Fruit Tree (m)
(−2.34, 0.95)	(1.86, 1.61)
(−2.27, 4.08)	(1.94, 4.51)
(−2.23, 7.59)	(1.37, 7.18)
(−2.63, 10.57)	(1.28, 11.16)
(−2.71, 14.23)	(1.88, 15.32)
(−2.11, 17.84)	(1.76, 18.05)
(−2.55, 20.29)	(2.21, 20.68)

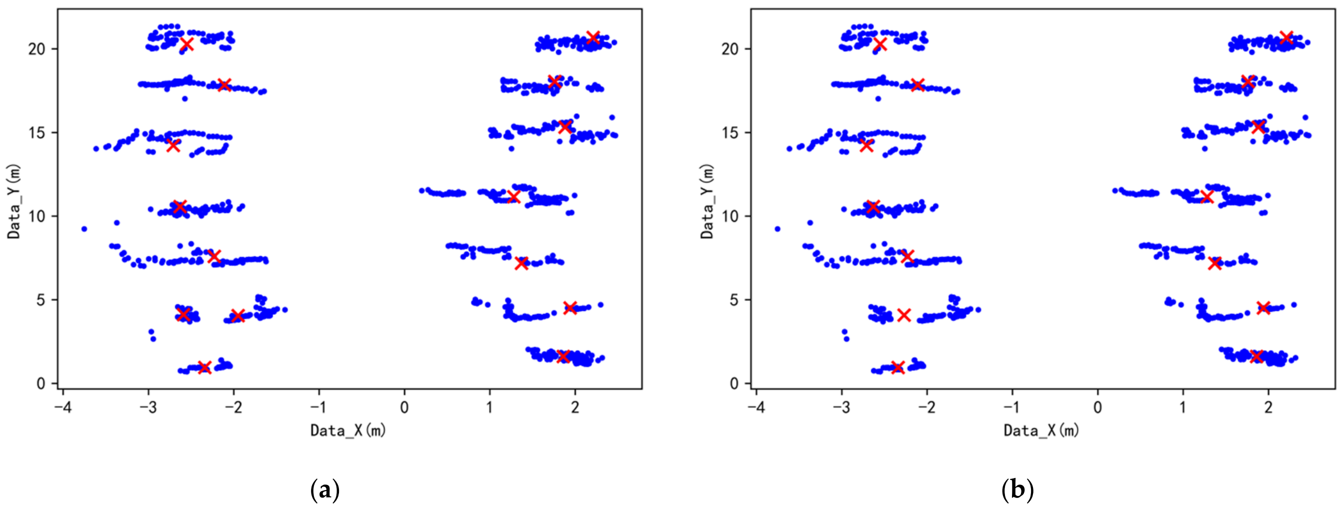


Figure 9. Algorithm detection results: (**a**) two cluster centers on a fruit tree; (**b**) results processed using this algorithm.

In Figure 8, the point cloud information of fruit trees on both sides was stored by the 2D-ICP algorithm. Therefore, the point cloud data were directly tested 100 times by the LAPO-DBSCAN algorithm, as shown in Table 5. One hundred iterations of the LAPO-DBSCAN algorithm are shown in Figure 10.

Table 5. Actual scene test results.

Algorithm Type	Times	Results (%)		Average Handling Time (s)
LAPO-DBSCAN	100	Positive detection rate False detection rate	96.69% 3.31%	1.14

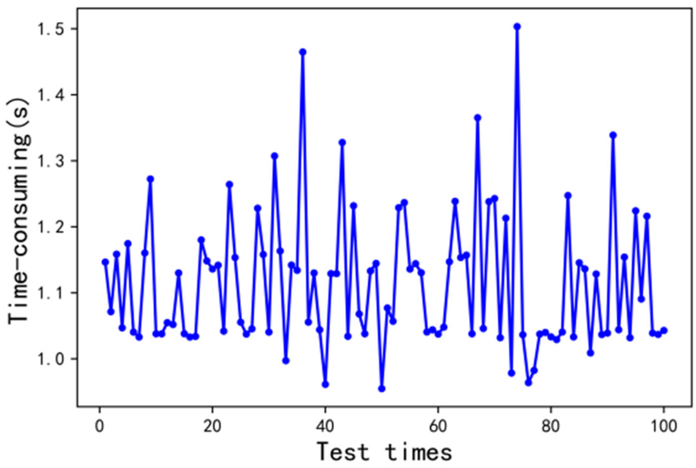


Figure 10. Actual scene, 100 iterations.

In Table 5, the positive detection rate of the algorithm is 96.69%, the false detection rate is 3.31%, the accuracy is maintained at more than 95%, and the average processing time is 1.14 s, which meets the accuracy requirements for the actual scene. Therefore, this algorithm can be used for the detection of fruit trees in orchards. This has certain practical significance for future navigation in orchards.

4. Discussion

In this research, the contour information of surrounding fruit trees was collected by a 2D LiDAR sensor mounted on an experimental platform, and the point cloud registration of fruit trees on both sides of the transportation robot was completed by the 2D-ICP algorithm. Then, the point cloud data were analyzed using the LAPO-DBSCAN clustering method to obtain the coordinate position points of each fruit tree. The most important thing is to propose a fruit tree position detection algorithm based on LAPO-DBSCAN. This algorithm has obvious advantages over those proposed in previous studies. Compared with K-means clustering, this algorithm does not need to set the number of clustering centers to be detected in advance, which makes it more convenient. Compared with the DBSCAN, this algorithm is more adaptable and can classify fruit trees more accurately. Compared with LAPO, this algorithm takes less time and is more accurate. Comparing the results of the simulation data (Table 2) with the results of the actual scene (Table 5), it can be seen that with the increase in environmental characteristics (from the detection of fruit tree trunks to the detection of fruit tree crowns), although the accuracy of the algorithm is reduced and the processing time is prolonged, the accuracy remains above 95% and the average processing time is 1.14 s, which generally meets the accuracy requirements of actual scenes. Therefore, this method can be used for the detection of fruit trees in orchards.

5. Conclusions

In this paper, a fruit tree position information perception method based on a 2D LiDAR sensor was proposed and verified on an experimental platform. According to the actual detection effect, the positive detection rate of the algorithm could reach 96.69%, the false detection rate was as low as 3.31%, and the average processing time was 1.14 s, indicating that the algorithm can be used in fruit tree detection to obtain the position of fruit trees. Although the algorithm has a good perception effect, there are also shortcomings. In the process of the experiment, because of the limitations of the 2D LiDAR sensor itself, the fruit tree information obtained was limited. When the algorithm is used for verification, there will be false detection and missed detection. From the detection of fruit tree trunks to the detection of fruit tree crowns, the amount of point cloud data for fruit trees will increase, resulting in a decrease in the positive detection rate of the algorithm. However, overall, the algorithm can still meet the requirements for the detection of fruit trees. In the future, the positions of fruit trees obtained by this algorithm could play a role in orchard navigation.

Author Contributions: Conceptualization, Y.W. and C.G.; Data curation, Y.W., G.Z. and H.G.; Formal analysis, R.S. and W.L.; Investigation, C.G.; Methodology, Y.W.; Software, Y.W.; Validation, G.Z., R.S. and H.G.; Writing—original draft, Y.W.; Writing—review and editing, C.G. and G.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (No. 2016YFD0700600).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Dworak, V.; Selbeck, J.; Ehlert, D. Ranging sensors for vehicle-based measurement of crop stand and orchard parameters: A review. *Trans. ASABE* **2011**, *54*, 1497–1510. [CrossRef]
2. Qiu, Q.; Han, J. 2.5-dimensional angle potential field algorithm for the real-time autonomous navigation of outdoor mobile robots. *Sci. China Inf. Sci.* **2011**, *54*, 2100. [CrossRef]
3. Lehtomäki, M.; Jaakkola, A.; Hyyppä, J.; Kukko, A.; Kaartinen, H. Detection of Vertical Pole-Like Objects in a Road Environment Using Vehicle-Based Laser Scanning Data. *Remote Sens.* **2010**, *2*, 641–664. [CrossRef]
4. Bargoti, S.; Underwood, J.P.; Nieto, J.I.; Sukkarieh, S. A Pipeline for Trunk Localisation Using LiDAR in Trellis Structured Orchards. In *Field and Service Robotics*; Springer International Publishing: Cham, Switzerland, 2015; Volume 105.
5. Underwood, J.P.; Jagbrant, G.; Nieto, J.I.; Sukkarieh, S. Lidar-Based Tree Recognition and Platform Localization in Orchards. *J. Field Robot.* **2015**, *32*, 1056–1074. [CrossRef]

6. Cabo, C.; Ordóñez, C.; López-Sánchez, C.A.; Armesto, J. Automatic dendrometry: Tree detection, tree height and diameter estimation using terrestrial laser scanning. *Int. J. Appl. Earth Obs.* **2018**, *69*, 164. [CrossRef]
7. De Conto, T.; Olofsson, K.; Görgens, E.B.; Rodriguez, L.C.E.; Almeida, G. Performance of stem denoising and stem modelling algorithms on single tree point clouds from terrestrial laser scanning. *Comput. Electron. Agric.* **2017**, *143*, 165–176. [CrossRef]
8. Gené-Mola, J.; Gregorio, E.; Guevara, J.; Auat, F.; Sanz-Cortiella, R.; Escolà, A.; Llorens, J.; Morros, J.-R.; Ruiz-Hidalgo, J.; Vilaplana, V. Fruit detection in an apple orchard using a mobile terrestrial laser scanner. *Biosyst. Eng.* **2019**, *187*, 171–184. [CrossRef]
9. Antonarakis, A.S.; Richards, K.S.; Brasington, J.; Bithell, M.; Muller, E. Retrieval of vegetative fluid resistance terms for rigid stems using airborne lidar. *J. Geogr. Res. Biogeosci.* **2015**, *113*, 96. [CrossRef]
10. Liu, L.; Lim, S.; Shen, X.; Yebra, M. A hybrid method for segmenting individual trees from airborne lidar data. *Comput. Electron. Agric.* **2019**, *163*, 104871. [CrossRef]
11. Liu, P.; Chen, J.; Zhang, M. Automatic control system of orchard tractor based on laser navigation. *Trans. CSAE* **2011**, *27*, 196–199.
12. Wang, D.; Liu, J.; Wang, J. Identification and classification of scanned target in forest based on hierarchical cluster. *Trans. CSAE* **2011**, *27*, 173–177.
13. Pierzchała, M.; Giguère, P.; Astrup, R. Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM. *Comput. Electron. Agric.* **2018**, *145*, 217–225. [CrossRef]
14. Zhou, J.; Hu, C. Inter-row Localization Method for Agricultural Robot Working in Close Planting Orchard. *Trans. CSAE* **2015**, *46*, 22–28.
15. Zhang, C.; Yong, L.; Chen, Y.; Zhang, S.; Ge, L.; Wang, S.; Li, W. A Rubber-Tapping Robot Forest Navigation and Information Collection System Based on 2D LiDAR and a Gyroscope. *Sensors* **2019**, *19*, 2136. [CrossRef] [PubMed]
16. Zhang, S.; Guo, C.; Gao, Z.; Sugirbay, A.; Chen, J.; Chen, Y. Research on 2D Laser Automatic Navigation Control for Standardized Orchard. *Appl. Sci.* **2020**, *10*, 2763. [CrossRef]
17. Monnier, F.; Vallet, B.; Soheilian, B. Trees detection from laser point clouds acquired in dense urban areas by a mobile mapping system. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *3*, 245–250. [CrossRef]
18. Chakraborty, M.; Khot, L.R.; Sankaran, S.; Jacoby, P.W. Evaluation of mobile 3D light detection and ranging based canopy mapping system for tree fruit crops. *Comput. Electron. Agric.* **2019**, *158*, 284–293. [CrossRef]
19. Zhang, X.; Li, X.; Zhang, B.; Zhou, J.; Tian, G.; Xiong, Y.; Gu, B. Automated robust crop-row detection in maize fields based on position clustering algorithm and shortest path method. *Comput. Electron. Agric.* **2018**, *154*, 165–175. [CrossRef]
20. Underwood, J.P.; Hung, C.; Whelan, B.; Sukkarieh, S. Mapping almond orchard canopy volume, flowers, fruit and yield using lidar and vision sensors. *Comput. Electron. Agric.* **2016**, *130*, 83–96. [CrossRef]
21. Narváez, F.J.Y.; del Pedregal, J.S.; Prieto, P.A.; Torres-Torriti, M.; Cheein, F.A.A. LiDAR and thermal images fusion for ground-based 3D characterisation of fruit trees. *Biosyst. Eng.* **2016**, *151*, 479–494. [CrossRef]
22. Astolfi, P.; Gabrielli, A.; Bascetta, L.; Matteucci, M. Vineyard Autonomous Navigation in the Echord++ GRAPE Experiment. *IFAC-PapersOnLine* **2018**, *51*, 704–709. [CrossRef]
23. Chen, X.; Wang, S.A.; Zhang, B.; Luo, L. Multi-feature fusion tree trunk detection and orchard mobile robot localization using camera/ultrasonic sensors. *Comput. Electron. Agric.* **2018**, *147*, 71–108. [CrossRef]
24. Nematollahi, A.F.; Rahiminejad, A.; Vahidi, B. A Novel Physical Based Meta-Heuristic Optimization Method Known as Lightning Attachment Procedure Optimization. *Appl. Soft Comput.* **2017**, *59*, 596–621. [CrossRef]
25. Zheng, T.; Luo, W. An Enhanced Lightning Attachment Procedure Optimization with Quasi-Opposition-Based Learning and Dimensional Search Strategies. *Comput. Intell. Neurosci.* **2019**, *2019*, 1589303. [CrossRef] [PubMed]
26. Wei, L.; Yang, S.; Ye, Z.W.; Huang, Q.; Huang, Y.K. An Image Segmentation Method Based on Two-Dimensional Entropy and Chaotic Lighting Attachment Procedure Optimization Algorithm. *Int. J. Pattern Recognit. Artif. Intell.* **2020**, *34*, 2054030.
27. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*; AAAI Press: Portland, OR, USA, 1996; pp. 226–231.
28. Guo, C.L.; Liu, G.; Zhang, W.J.; Feng, J. Apple tree canopy leaf spatial location automated extraction based on point cloud data. *Comput. Electron. Agric.* **2019**, *166*, 104975. [CrossRef]
29. Liu, X.; Hu, C.H.; Li, P.P. Automatic segmentation of overlapped poplar seedling leaves combining Mask R-CNN and DBSCAN. *Comput. Electron. Agric.* **2020**, *178*, 105753. [CrossRef]
30. Niu, R.X.; Zhang, X.Y.; Wang, J.; Zhu, H.; Huang, J.; Chen, Z.W. Orchard Trunk Detection Algorithm for Agricultural Robot Based on Laser Radar. *Trans. CSAE* **2020**, *51*, 21–27.