

A Method for 3D Virtual Orchard Construction from LiDAR Point Clouds (Postprint)

Authors: Feng Han, Zhang Hao, Wang Zi, Jiang World, Liu Weihong, Zhou Linghui, Yaxiong Wang, Kang Feng, Liu Xingxing, Zheng Yongjun

Date: 2023-02-17T00:00:00+00:00

Abstract

To address issues such as low digitization levels in orchard management and relatively simplistic construction methods, this study proposes a three-dimensional virtual orchard construction method based on laser point clouds. First, a handheld three-dimensional point cloud acquisition device (3D-BOX) combined with the Simultaneous Localization and Mapping-Lidar Odometry and Mapping (SLAM-LOAM) algorithm is employed to obtain orchard point cloud datasets; then statistical filtering algorithms are used to remove outliers and noise points from the point cloud data, and combined with the Cloth Simulation Filtering (CSF) algorithm and the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering algorithm to achieve ground removal and fruit tree clustering segmentation, followed by downsampling using a VoxelGrid filter; finally, the Unity3D engine is utilized to construct a virtual orchard roaming scene, converting real-time GPS (Global Positioning System) data of operating machinery from the WGS-84 coordinate system to the Gauss projection plane coordinate system, and displaying real-time trajectories through LineRenderer to achieve visualization of operating machinery motion trajectory control and operation tracks. To validate the effectiveness of the virtual orchard construction method, orchard construction method tests were conducted in crabapple orchards and mango orchards. The results indicate that the proposed point cloud data processing method achieved clustering segmentation accuracies of 95.3% and 98.2% for crabapple trees and mango trees, respectively; through comparison with the row spacing and plant spacing of actual mango orchards, the virtual mango orchard exhibited an average inter-row error of approximately 3.5% and an average intra-row error of approximately 6.6%. Furthermore, compared with actual orchards, the virtual orchard constructed using Unity3D can effectively reproduce the three-dimensional actual conditions of the orchard, achieving favorable visualization effects, and providing a technical solution for the digital modeling and management of orchards.

Full Text

Three-Dimensional Virtual Orchard Construction Method Based on Laser Point Cloud

FENG Han¹, ZHANG Hao¹, WANG Zi¹, JIANG Shijie¹, LIU Weihong¹, ZHOU Linghui², WANG Yaxiong², KANG Feng², LIU Xingxing¹, ZHENG Yongjun^{1,3*}

¹College of Engineering, China Agricultural University, Beijing 100083, China

²College of Engineering, Beijing Forestry University, Beijing 100083, China

³Modern Agricultural Equipment and Facilities Engineering Research Center, Beijing 100083, China

Abstract: To address the problems of low digitalization level and relatively single construction methods in orchard management, this study proposes a three-dimensional virtual orchard construction method based on laser point cloud. First, a hand-held 3D point cloud acquisition device (3D-BOX) combined with the Simultaneous Localization and Mapping-Lidar Odometry and Mapping (SLAM-LOAM) algorithm was used to obtain the orchard point cloud dataset. Then, statistical filtering algorithm was employed to remove outliers and noise points from the point cloud data, combined with cloth simulation filtering for ground removal and fruit tree clustering segmentation, and further using DBSCAN clustering algorithm. Finally, the Unity3D engine was utilized to construct a virtual orchard roaming scene, converting real-time GPS data of the operating machinery from the WGS-84 coordinate system to the Gauss projection plane coordinate system, and displaying the real-time trajectory through LineRenderer to achieve visualization of machinery motion trajectory control and operation trajectory. To verify the effectiveness of the virtual orchard construction method, tests were carried out in both a crabapple orchard and a mango orchard. The results showed that the proposed point cloud data processing method achieved clustering segmentation accuracies of 95.3% and 98.2% for crabapple and mango trees, respectively. Compared with the actual row and plant spacing in the mango orchard, the average inter-row error of the virtual mango orchard was approximately 3.5%, and the average inter-plant error was approximately 6.6%. The virtual orchard constructed by Unity3D effectively reproduced the actual three-dimensional situation of the orchard and achieved good visualization effects, providing a technical solution for digital modeling and management of orchards.

Keywords: virtual orchards; point cloud processing; Unity3D; trajectory visualization; LiDAR; SLAM-LOAM

1 Introduction

With the increasing trend of orchard area and fruit production year by year, standardized and normalized orchard management has become particularly important. Constructing a three-dimensional virtual orchard that contains a large amount of agricultural production information can not only display real orchard and fruit tree three-dimensional morphology data more stereoscopically and intuitively, but also combine with the Unity3D engine to realize virtual orchard roaming scenes and visualization of orchard machinery working trajectories, which greatly facilitates staff management of orchards and effectively promotes the development of smart orchards and unmanned orchards [1].

Due to the complexity of orchard terrain and the randomness of fruit tree growth, reconstructing three-dimensional models of orchards in real environments has become challenging. Currently, vision and Light Detection and Ranging (LiDAR) are mainly used for three-dimensional reconstruction research. However, vision sensors are mostly suitable for constructing single objects or small-scale models, and have disadvantages such as susceptibility to external environmental influences and large construction errors for large-scale three-dimensional model reconstruction of building clusters, forests, and orchards [2-6]. LiDAR, with its advantages of strong penetration, long straight-line propagation distance, and large scanning angle, has been widely used in cultural relics protection, building outline extraction, forest modeling, and other fields [7-9].

For large-scale three-dimensional model construction of orchards, Wang et al. [10] used mobile LiDAR to obtain orchard point cloud maps, combined with a lightweight LiDAR-Inertial Measurement Unit (LiDAR-IMU) state estimator and rotation constraint optimization algorithm to reconstruct orchards. Hu et al. [11] proposed a trunk parameter estimation and trunk model reconstruction method based on multi-station laser point cloud data, which could accurately extract trunk geometric parameters through established models for visualization and biomass estimation of large-scale models such as forests. To improve the accuracy of three-dimensional model construction, Indirbai et al. [12] used LiDAR scanning to obtain point cloud libraries, combined with hierarchical minimum segmentation algorithm and supervoxel clustering algorithm to achieve detection of tree crowns and trunks and segmentation of individual plants, and reconstructed high-precision tree models using laser point cloud libraries. Zhai et al. [13] realized three-dimensional reconstruction of trees based on LiDAR point cloud data through a graph optimization Simultaneous Localization and Mapping (SLAM) method. Zheng and Fu [14] studied theoretical derivatives for batch optimization using a LiDAR batch optimization framework, and integrated the batch optimization framework into the LiDAR Odometry and Mapping (LOAM) framework to reduce cumulative errors in LiDAR mapping. Zhang and Singh [15] proposed a real-time 6-degree-of-freedom 2-axis single-line LiDAR odometry and mapping method to address motion distortion issues in point clouds received at different times that cause point cloud registration errors, but it lacked back-end optimization

functionality.

The above studies show that LiDAR can effectively obtain three-dimensional point cloud data of large-scale features such as orchards, but the accuracy of point cloud data processing still needs improvement. The number and location characteristics of fruit trees in large orchards are also not described in detail. This study takes crabapple orchards and mango orchards as research objects and proposes a three-dimensional virtual orchard construction method based on laser point cloud data. Statistical filtering algorithm, Cloth Simulation Filtering (CSF) algorithm, and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm are used to process orchard point cloud datasets obtained by hand-held 3D point cloud acquisition equipment (3D-BOX) for outlier removal, ground removal, and clustering analysis. Based on downsampling processing and Unity3D engine, virtual orchard construction is completed, and secondary development with operating machinery is implemented to realize motion trajectory control and operation trajectory visualization.

2 Equipment and Methods

2.1 Three-Dimensional Data Acquisition

This study selected two locations as virtual orchard construction objects: one is the crabapple orchard west of the College of Engineering at China Agricultural University, tested on March 1, 2021; the other is the Mango Manor in Tianyang County, Baise City, Guangxi Province, tested on April 8, 2021. The two orchards were scanned using LiDAR to complete the original three-dimensional point cloud data collection.

The hand-held 3D point cloud acquisition equipment (3D-BOX) from Dalian Hangjia Robot Technology Co., Ltd. was used for three-dimensional point cloud data acquisition, as shown in [Figure 1: see original paper]. The equipment consists of data acquisition and client display parts. The data acquisition device comprises a Velodyne VLP-16 LiDAR, an Inertial Measurement Unit (IMU), and a microcomputer; the microcomputer connects the LiDAR and IMU, drives sensor operation, collects three-dimensional point cloud data and performs mapping, while simultaneously interacting with the client through a wireless network.

The Global Navigation Satellite System (GNSS) positioning device LGR-BD982 receiver module provided by Beijing Leigerui Technology Co., Ltd. was selected as the GPS (Global Positioning System) data acquisition device.

2.2 Point Cloud Preprocessing

After LiDAR completes three-dimensional scanning of the orchard, the LOAM algorithm is used for three-dimensional orchard point cloud preprocessing. This

preprocessing mainly includes four aspects: feature point extraction, finding corresponding edge lines and local patches, LiDAR motion estimation, and mapping, to construct the original three-dimensional orchard point cloud model. In the original three-dimensional point cloud data model, due to LiDAR accuracy, data acquisition speed and stability, and environmental factors affecting the measured objects, the point cloud data contains a large number of noise points, while occlusion by non-measured objects also produces some outliers, which need to be removed. Additionally, the huge point cloud data affects subsequent visualization processing, so downsampling of the entire point cloud dataset is required to reduce the number of points while ensuring clear expression of fruit tree point cloud structure. In the point cloud data post-processing stage, algorithms for outlier removal, ground removal, clustering, and point cloud downsampling are mainly used. The flowchart of the model construction method in this study is shown in [Figure 2: see original paper].

2.2.1 Feature Point Extraction The feature point extraction process is as follows: feature points are classified by curvature value c . When c is the maximum value in a continuous data point set, the point is set as an edge point; when c is the minimum value in a continuous data point set, the point is set as a plane point. The selection of edge points starts from the maximum curvature c , while plane feature points are the opposite. Assuming the k -th frame point cloud data in LiDAR scanning point cloud dataset $\{L\}$ is p_k , and a certain data point in p_k is i , then the set of neighboring points of data point i in the same scanning frame is S , and the curvature function c at point i is defined as shown in Equation (1).

2.2.2 Finding Corresponding Edge Lines and Local Patches The selection method for edge lines is as follows: assuming edge point $i \in p_k$, find the nearest neighbor point of point i on different scanning lines, then judge these two points based on curvature values. If they are edge points, determine a straight line based on these two points to obtain the edge line. The selection method for corresponding local patches is as follows: assuming plane point $i \in H_k$, the corresponding local patch can be determined by three non-collinear points. Select two nearest neighbor points of i on the same scanning line, then select a nearest neighbor point corresponding to point i on different scanning lines, thereby determining the local patch corresponding to plane point i . The selection of plane points for the nearest neighbor points also requires calculating their curvature values to judge whether they are plane points.

The above t_k and H_k are defined as follows: assuming the start time of the k -th frame scan is t_{k1} , the scan end time is t_{k2} , the current frame point cloud data obtained at the end of the scan is $p\{k\}$. Using the spatial transformation matrix to map the current frame point cloud data to the coordinate system of the point cloud data at time t_{k1} , the transformed point cloud data is $\hat{p}\{k\}$. Using the k -th frame point cloud data p_k obtained at time t_{k2} and the transformed point cloud data $\hat{p}\{k\}$ for registration, rough pose estimation is completed, that is,

point cloud registration at time t_k is completed, thereby completing LiDAR pose estimation.

In the point cloud data p_k obtained from the k -th frame scan, edge points and plane points are extracted to form edge point set E_k and plane point set H_k , respectively. In \mathcal{P}_{k1} , find the edge lines corresponding to edge point set E_k and the local patches corresponding to plane point set H_k . Therefore, this study needs to judge the nearest neighbor points in two adjacent frames of point clouds, and calculate the minimum distance from edge point set E_k points to the corresponding edge lines in \mathcal{P}_k and from plane point set H_k points to the corresponding local patches in H_k . The calculation equations are shown in Equation (2) and Equation (3).

where t_k moment edge point $i \in E_k$; t_{k-1} moment nearest neighbor points j, l determine the edge line; d_{E} is the minimum distance from edge point i to the edge line corresponding to nearest neighbor points j, l at t_{k-1} moment; t_k moment plane point $i \in H_k$; d_H is the distance from plane point i to its corresponding local patch at t_{k-1} moment nearest neighbor points j, l, m .

2.2.3 LiDAR Motion Estimation Assuming LiDAR moves at a constant speed, the pose transformation matrix of LiDAR in the time period $[t_k, t]$ is $T_{L} = [t_x, t_y, t_z, \tau_x, \tau_y, \tau_z]$. For any point i in the k -th frame point cloud data p_k , the LiDAR pose $T_{Lk,i}$ in the time period $[t_k, t]$ can be obtained through interpolation. The coordinate transformation of edge points and plane points in point cloud data p_k is shown in Equation (4).

where $X_{Lk,i}$ is the coordinate of the corresponding edge point or plane point after coordinate transformation; $R_{Lk,i}$ is the rotation matrix; $\tau_{Lk,i}$ is the translation matrix; $\tau = [T_{Lk,i}(4:6)]$, \hat{n} is the skew-symmetric matrix of n ; n is the unit vector of rotation direction, calculated as shown in Equation (5).

The error function for edge points and edge lines, plane points and local patches is shown in Equation (6).

The error function f corresponds to feature points, where d represents the distance from the point to its corresponding point. The Jacobian matrix J of f can be calculated, and the LiDAR pose $T_{Lk,i}$ is updated. By using a nonlinear method to make d approach 0, $f(T_{Lk,i}+1)$ is solved. λ can be determined by the least squares method, as shown in Equation (7).

where f is the error function; J is the Jacobian matrix.

2.2.4 LiDAR Mapping The three-dimensional mapping process of the LiDAR LOAM algorithm is performed in the world coordinate system W , and the mapping frequency is an order of magnitude lower than the motion estimation frequency. Assuming the point cloud data converted in the time period $[t_k, t_{k+1}]$ in the LiDAR coordinate system L is p_k , the LiDAR pose estimation is

$T\{Lk\}(t_{k+1})$, and the pose at time t_k is $T_{\{Lk\}}(t_k)$. The mapping at time t_k is Q_k . The mapping algorithm can align Q_k with $Q_k(t_{k+1})$ based on $T\{Lk\}(t_k)$, and convert the LiDAR pose $T_{\{Wk\}}(t_k)$ to $T_{\{Wk\}}(t_{k+1})$ in the world coordinate system. Simultaneously, the point cloud data p_k is converted to the world coordinate system as $Q_k(t_{k+1})$, and $Q_k(t_{k+1})$ is continuously integrated into the map Q_{k+1} .

2.3 Point Cloud Data Post-Processing

After preprocessing of laser point clouds, the orchard tree model can be basically displayed. In the point cloud map after LiDAR scanning, it contains fruit trees, ground point clouds, and a large number of noise points. Although the contour of fruit trees can be highlighted after point cloud preprocessing, a large number of ground point clouds not only increase the amount of calculation but are also very unfavorable for subsequent point cloud map clustering. Therefore, it is necessary to remove such data points. In addition, huge point cloud data will have a certain impact on subsequent visualization processing, so the overall point cloud dataset needs to be downsampled to reduce the number of points while ensuring clear expression of fruit tree point cloud structure. In the point cloud data post-processing stage, algorithms for outlier removal, ground removal, clustering, and point cloud downsampling are mainly used.

2.3.1 Outlier Removal Since the point clouds obtained by 3D-BOX are scattered point clouds, the statistical filter in The Point Cloud Library (PCL) [16, 17] is used to remove outliers. For any point p_i in point cloud set P , this method finds the k nearest points $p_j, p_{j+1}, \dots, p_{j+k-1}$ to that point, calculates the Euclidean distance and averages them, as shown in Equation (8).

All K neighboring points' averages a_1 a a_N statistical results approximately follow a Gaussian distribution. The mean and standard deviation σ of the distances from all points to their k neighboring points determine its shape, denoted as $N(\mu, \sigma^2)$, as shown in Equation (9) and Equation (10).

A distance threshold model for removing noise points is established using the K -nearest neighbor distance statistical method, as shown in Equation (11).

where a represents the multiple of standard deviation.

Traverse the average distance of K neighboring points for all points p_i in the point cloud set, and compare them with the distance threshold respectively. If it is not within the $[\mu - \text{threshold}, \mu + \text{threshold}]$ range, that is, when the distance from point p_i to k neighboring points exceeds the average distance and is greater than a times the standard deviation, it is marked as an outlier and separated from the point cloud dataset. The processing flow is shown in [Figure 3: see original paper].

2.3.2 Ground Point Cloud Removal The original dataset obtained by LiDAR scanning has a large and disordered point cloud. The CSF algorithm

[18, 19] is used for ground point cloud removal operations, retaining non-ground point clouds. Its working principle is to invert the point cloud and determine the final shape of the cloth through cloth simulation filtering algorithm [20], using it as the basis for distinguishing ground points and non-ground points. Cloth simulation filtering algorithm is an improvement based on cloth simulation for laser point clouds. The relationship between grid point position and external force in the improved cloth filtering algorithm is expressed as Equation (12).

where m is the mass of grid points, usually set to 1; Δt is the time step; G is a constant.

The displacement vector d of grid points is calculated as shown in Equation (13).

where p_0 represents the current position of the grid point to be moved; p_i represents the position of neighboring grid points of p_0 ; n represents the unit vector in the vertical direction $n = (0, 0, 1)^T$; b is used to determine whether the grid point is movable. If the grid point is movable, b is 1, otherwise it is 0. The process flow is shown in [Figure 4: see original paper].

2.3.3 DBSCAN Clustering Analysis DBSCAN [21] is one of the classic algorithms based on density clustering, which can divide regions with sufficiently high point density into clusters and find clusters of arbitrary shapes in data containing noise points. In this study, two parameters, Eps clustering radius and MinPts minimum number of samples in the core point neighborhood, are used as important prerequisites for region division. Regions with high point density are divided into clusters to form the largest set of points that satisfy density connectivity. The DBSCAN algorithm is used for clustering analysis of orchard trees, and the total number of orchard trees is obtained through clustering.

2.3.4 Point Cloud Downsampling Importing the processed point cloud map into Unity3D engine for secondary development, excessive point cloud numbers will increase memory usage in project development. Therefore, point cloud data needs to be downsampled to reduce point cloud density and file memory usage while maximizing the retention of orchard point cloud feature information. This study uses the VoxelGrid filter for downsampling.

2.4 Virtual Orchard Design

The orchard roaming scene displays the actual orchard situation in a virtual manner. Based on the orchard point cloud map, secondary development is performed in the Unity3D engine [22] to process the orchard point cloud model and develop a three-dimensional virtual orchard. It mainly includes functions such as scene roaming, trajectory display, coordinate interaction, and file management. The software packages used for building the virtual orchard include the Point Cloud Library PCL, Unity3D development engine, point cloud file format

conversion tool MeshLab, operating machinery sprayer model processing tool 3DMax, and scripting language C#.

2.4.1 GPS Data Processing

- (1) Latitude and longitude extraction. This function uses the StreamReader class in C# to achieve line-by-line reading of raw data in the file stream. When reading a line of data, this study uses two data extraction methods: one is to query by special characters and then intercept the string; the other is to query based on the position of the target object in the string and intercept the string at the specified position. The intercepted latitude and longitude strings are converted to double floating-point type. For easy querying, the converted latitude and longitude data are first stored in a two-dimensional array and then added to a dictionary for temporary storage.
- (2) Coordinate conversion. The GPS data collected by the device belongs to the WGS-84 geodetic coordinate system, which is a spherical coordinate system containing latitude, longitude, and altitude. This study will perform trajectory drawing in a two-dimensional plane, so the spherical coordinates need to be processed and the geodetic coordinates converted to Gauss projection plane coordinates through Gauss projection forward calculation. The coordinate solution is shown in Equation (14) and Equation (15).

where all longitudes and latitudes are converted to radians; X corresponds to the meridian arc length of the latitude; B is the latitude of the coordinate point, ($^{\circ}$); N is the radius of curvature in the prime vertical; a is the semi-major axis; e is the first eccentricity of the ellipsoid; L is the longitude of the point, ($^{\circ}$); L_0 is the longitude of the central meridian, ($^{\circ}$).

This study uses 3° zone division in the conversion from WGS-84 coordinate system to Gauss projection coordinate system. After coordinate conversion, the first two digits of the obtained y -value are the zone number. To prevent negative y -values, 500 km is added to the y -value result, which is equivalent to a uniform translation of the y -value, thus having no impact on trajectory display.

2.4.2 Roaming Scene Design Based on the reconstructed three-dimensional virtual orchard, the operating machinery is loaded into the virtual orchard environment to simulate the working conditions of orchard machinery. Realizing the mapping between the real world and the virtual world through operating machinery, functions such as operating machinery working trajectory drawing and operation process simulation are established. In the virtual orchard environment, necessary viewing functions such as translation, rotation, and zoom are added, and Baidu Map's ordinary positioning Application Programming Interface (API) and system time are integrated into the location display bar at the upper right corner of the display interface to achieve real-time presentation

of the orchard roaming scene. Other functions in the virtual orchard include measurement tools, time and location display, and historical data saving, establishing a relatively complete orchard machinery display scene. Based on the Unity3D development platform, the System.IO.Ports class is used to design a GPS data acquisition interaction interface with capabilities for real-time reading of GPS serial port data, data storage, latitude and longitude extraction, and coordinate conversion processing.

Unity3D enables hierarchical development environment, visual editing environment, and other operations, and can use graphics engines such as Direct3D, OpenGL, APIs, and the PhysX physics engine to complete development and debugging functions. Developers can view the three-dimensional virtual orchard roaming status at any time during development.

The texturing of point clouds is mainly implemented in Unity3D. The main process is as follows: match the target point cloud information with image texture information to obtain a point cloud model with real color information; the colored point cloud can directly generate a dense triangular mesh model, and RGB color matching is performed when rendering the triangular mesh model to obtain a triangular mesh model with real colors; for relatively sparse triangular mesh models, correction and texture mapping are performed, and finally a three-dimensional orchard model with realistic texture is obtained.

2.4.3 Trajectory Visualization The processing of GPS data collected by GPS data acquisition equipment is a key link in trajectory drawing, and the processing results will directly affect subsequent trajectory visualization. After the collected GPS data is converted to Gauss projection plane coordinate data, this data is used to achieve trajectory visualization based on the Unity3D platform. The orchard point cloud model is imported into Unity3D with the coordinate origin position as the centroid position. The trajectory drawing process will take the centroid position as the starting point, with longitude data along the Z-axis direction and latitude data along the X-axis direction. When collecting GPS data in practice, the starting position of collection is fixed, so when drawing the trajectory, the offset between the centroid coordinates and the fixed starting point must be calculated first, and then the plane coordinate data is read for drawing. The trajectory drawing part relies on the LineRenderer in Unity3D, which can set properties such as line material, color, and width. This study uses default material and color, with width set to 0.05 m, and the trajectory drawing coordinate system is set to the world coordinate system.

3 Results Analysis and Function Verification

3.1 Algorithm Verification Results Analysis

3.1.1 Point Cloud Preprocessing After performing preliminary preprocessing operations such as feature point extraction, finding corresponding edge lines

and local patches, LiDAR motion estimation, and mapping on the original point cloud data of crabapple orchard and mango orchard scanned by LiDAR, the pre-processed orchard point cloud dataset was obtained. The original orchard point clouds and preprocessed point clouds are shown in [Figure 6: see original paper].

3.1.2 Point Cloud Post-Processing

- (1) Statistical filtering to remove outliers. Statistical filtering based on the PCL point cloud library under Ubuntu system was used to remove outliers from the preprocessed orchard point cloud dataset. The K-nearest neighbor search point threshold r was set to 40, and the standard deviation σ multiple a was set to 0.6. These two values r and a are empirical values. In the crabapple orchard data processing, the dataset had 4,746,821 points before outlier removal filtering, and 4,113,008 points after filtering. In the mango orchard data processing, the dataset had 4,303,845 points before filtering, and 3,548,440 points after filtering. The contour of mango orchard fruit trees has been highlighted, but there are still a few stray points. Due to the large amount of weeds of varying heights in the actual environment, the ground point cloud data scanned by LiDAR is relatively sparse. Statistical filtering removed some ground point clouds, with effects shown in [Figure 7: see original paper], verifying that this statistical filter with threshold parameters can be used in other scenarios.
- (2) CSF ground removal. The CSF algorithm was used for ground point cloud data removal operations in the PCL point cloud library. The removal effect for crabapple orchard is shown in Figure 8: see original paper, with green representing fruit tree point clouds and red representing ground point clouds. The mango orchard test environment has a relatively large slope, and the ground point cloud is sparse after filtering, with removal effects shown in Figure 8: see original paper. After ground point cloud data removal, the filtered tree crown point cloud segmentation features are not obvious.
- (3) Clustering analysis. By separately counting the actual crabapple orchard and mango orchard trees, clustering analysis was performed as follows: set clustering radius Eps to 1, density threshold $MinPts$ to 100; the crabapple orchard has 43 trees, including two non-identical varieties with different tree heights. Clustering also yielded 43 clusters. The actual environment has 57 mango trees; clustering yielded 56 clusters. The positive detection rate and accuracy rate are used to analyze the clustering segmentation progress of orchard laser point clouds, which can be calculated by Equation (16).

where n is the number of correctly segmented trees; N is the number of trees obtained by segmentation; M is the actual number of trees counted manually. The orchard DBSCAN clustering accuracy results are shown in .

Table 1 Orchard DBSCAN clustering accuracy

Orchard type	Actual tree count	Segmented tree count	Correctly segmented trees	Positive detection rate (%)	Accuracy (%)
Crabapple	43	43	41	95.3	95.3
Mango	57	56	55	98.2	96.5

In the crabapple orchard, due to canopy closure between rows and within rows, DBSCAN clustering did not completely cluster each tree, with an accuracy of only 95.3%, but the clustering result was consistent with the actual tree count. For the mango orchard, canopy closure also existed between rows and within rows, and the closure within rows was more severe, so DBSCAN clustering did not completely cluster each tree, with a positive detection rate and accuracy of 98.2%. The clustering segmentation effect still needs improvement. The clustering effect is shown in [Figure 9: see original paper].

- (4) Downsampling processing. While ensuring basic consistency of fruit tree point cloud features, downsampling processing was performed with voxel size set to $0.05 \times 0.05 \times 0.05$. The crabapple orchard point cloud count was 2,316,788 before downsampling and 610,076 after downsampling. The mango orchard point cloud count was 2,216,632 before downsampling and 855,130 after downsampling. The downsampled point cloud data was then merged and displayed with the pre-downsampling point cloud data, with red points representing pre-downsampling data points and green points representing post-downsampling data points. The post-downsampling effect is shown in [Figure 10: see original paper].
- (5) Accuracy detection. Due to certain slopes and uneven ground in the orchard, with many weeds on the ground, there will be errors in fruit tree height after orchard reconstruction. Therefore, this study mainly detects reconstruction accuracy by measuring the plant spacing and row spacing of fruit trees. Since the crabapple orchard contains two non-identical variety trees, there will be large errors in measurement positions. In the actual mango orchard environment, four positions were selected for measurement: the starting position, middle position, end position of the first row, and the scanning end position. The measurement data is shown in [Figure 12: see original paper], and the two are compared. The results show that the average inter-row error is about 3.5%, and the average inter-plant error is about 6.6%. Due to severe canopy closure within rows, the error is larger, with maximum error around 10%. By comparing the accuracy of actual mango trees with modeled plant spacing and row spacing, as shown in [Figure 13: see original paper], the plant spacing accuracy is greater than 88.8%, and row spacing accuracy is greater than 93.6%.

3.1.3 Three-Dimensional Orchard Construction Efficiency The three-dimensional orchard model reconstruction in this study was performed on a CPU

i7-9750H with 8 GB RAM under Ubuntu 18.04 system. The three-dimensional orchard construction process generates a large number of point clouds, and algorithm running efficiency gradually decreases as the number of point clouds increases. The point cloud preprocessing process mainly uses LiDAR combined with the LOAM algorithm, which consumes relatively little time. Therefore, this study focuses on comparing the point cloud quantities and running times of the crabapple orchard and mango orchard during post-processing, as shown in . By accumulating the duration of the above three time periods, the point cloud post-processing time for both the crabapple orchard and mango orchard is within 30 seconds, demonstrating high processing efficiency.

Table 2 Run schedule of the algorithms

Point cloud processing stage	Crabapple orchard	Mango orchard
	Point count	Time (s)
Preprocessed point cloud	4,746,821	-
Statistical filtering	4,113,008	12.3
CSF cloth filtering	2,316,788	8.7
DBSCAN clustering	670,076	5.4

3.2 Virtual Orchard System Function Test Verification

To verify the effective operation of the functions possessed by the virtual orchard system, tests were conducted in the Mango Manor in Tianyang County, Baise City, Guangxi Province and the crabapple orchard at China Agricultural University. The verification mainly focused on orchard scene roaming and visualization of orchard machinery operation trajectories. By logging into the interface to enter the main interface of the three-dimensional virtual orchard system, users can select functional scenarios according to their needs. Clicking the corresponding button enters the corresponding scene. Clicking the trajectory file selection button allows trajectory drawing. The trajectory file data here is converted Gauss projection plane coordinate data. By testing the functions involved in the virtual orchard system and demonstrating the functions possessed by the system, as shown in [Figure 14: see original paper], the visualization effects of operating machinery and operation trajectories in both crabapple orchard and mango orchard are good. The test results show that all functions of the virtual orchard system can run well.

4 Conclusion

This study proposes a three-dimensional virtual orchard construction method based on LiDAR and verifies the method through crabapple orchard and mango orchard. The main conclusions are as follows:

- (1) The constructed orchard model has high accuracy. This study selected the row spacing and plant spacing of mango orchard trees in the actual environment for measurement and compared them with the reconstructed results. The average inter-row error is about 3.5%, and the average inter-plant error is about 6.6%.
- (2) The proposed method has universality. This study used the DBSCAN algorithm for clustering analysis of crabapple orchard and mango orchard, achieving clustering accuracies of 95.3% and 98.2%, respectively. Moreover, the point cloud post-processing time for both orchards is within 30 seconds.
- (3) The three-dimensional virtual orchard developed based on the Unity3D engine can run all added functions smoothly. Through testing the functions possessed by the virtual orchard system, the three-dimensional virtual orchard constructed in this study can meet the requirements of operating machinery trajectory drawing and can simulate historical operation processes. It possesses functions such as comprehensive three-dimensional display of orchard scenes, distance measurement, and location display.
- (4) The DBSCAN clustering algorithm in this study needs improvement in clustering segmentation effects under severe canopy closure conditions. Subsequent work can optimize the clustering segmentation algorithm to improve segmentation accuracy, and integrate clustering segmentation as a function in the virtual orchard to achieve clustering analysis in the virtual orchard.

References

- [1] ZHENG Y, CHEN B, LYU H, et al. Research progress of orchard plant protection mechanization technology and equipment in China[J]. Transactions of the CSAE, 2020, 36(20): 110-124.
- [2] NI Z, BURKS T F. Three-dimensional dense reconstruction of plant or tree canopy based on stereo vision[J]. Agricultural Engineering International: CIGR Journal, 2018, 20(2): 248-262.
- [3] BERVEGLIERI A, TOMMASELLI A, LIANG X, et al. Vertical optical scanning with panoramic vision for tree trunk reconstruction[J]. Sensors, 2017, 17(12): ID 2938.
- [4] ZHENG L, MAI C, LIAO W, et al. 3D point cloud registration of apple trees based on Kinect camera[J]. Transactions of the CSAM, 2016, 47(5): 9-14.
- [5] BERVEGLIERI A, TOMMASELLI A, LIANG X, et al. Vertical optical scanning with panoramic vision for tree trunk reconstruction[J]. Sensors, 2017, 17(12): ID 2938.

- [6] NEWCOMBE R A, FITZGIBBON A, IZADI S, et al. KinectFusion: Real-time dense surface mapping and tracking[C]// 10th IEEE International Symposium on Mixed and Augmented Reality. Piscataway, New York, USA: IEEE, 2011: 127-136.
- [7] LUO Q, LI G, YANG Y, et al. Application of ground 3D laser scanning technology in the 3D reconstruction of cultural relics: A case study of the Golden Temple[J]. Urban Survey, 2020(3): 112-116.
- [8] YANG J, WANG Z, ZHU K, et al. High-precision three-dimensional reconstruction of complex sports venues under the integrated control of laser scanning and unmanned aerial vehicles[J]. Bulletin of Surveying and Mapping, 2020(12): 21-26.
- [9] WANG G, LI Q, YANG X, et al. Modeling of forest scene based on airborne lidar point cloud[J]. Journal of Beijing University of Civil Engineering and Architecture, 2021, 37(2): 39-46.
- [10] WANG K, ZHOU J, ZHANG W H, et al. Mobile LiDAR scanning system combined with canopy morphology extracting methods for tree crown parameters evaluation in orchards[J]. Sensors, 2021, 21(2): ID 414.
- [11] HU C, PAN S, ZHANG H, et al. Trunk model establishment and parameter estimation for a single tree using multistation terrestrial laser scanning[J]. IEEE Access, 2020, 8: 102263-102277.
- [12] INDIRABAI I, NAIR M V H, JAISHANKER R N, et al. Terrestrial laser scanner based 3D reconstruction of trees and retrieval of leaf area index in a forest environment[J]. Ecological Informatics, 2019, 53: ID 100986.
- [13] ZHAI X, SHAO J, ZHANG W, et al. 3D reconstruction of trees based on mobile LiDAR point clouds[J]. China Agriculture Information, 2019, 31(5): 84-89.
- [14] ZHENG L, FU Z. BALM: Bundle adjustment for LiDAR mapping[J/OL]. arXiv 2020: 8.
- [15] ZHANG J, SINGH S. LOAM: Lidar odometry and mapping in real-time. in robotics[C]// Robotics: Science and Systems Conference(RSS). New York City, USA: RSS, 2014.
- [16] RUSU R B, MARTON Z C, BLODOW N, et al. Towards 3D point cloud based object maps for household environments[J]. Robotics And Autonomous Systems, 2008, 56(11): 927-941.
- [17] LUO F. Application of statistical denoising algorithm of PCL library point cloud[J]. Modern Computer (Professional Edition), 2016(26): 63-66.
- [18] ZHANG W, QI J, WAN P, et al. An easy-to-use airborne LiDAR data filtering method based on cloth simulation[J]. Remote Sensing, 2016, 8(6): ID 501.

- [19] PROVOT X. Deformation constraints in a mass-spring model to describe rigid cloth behaviour[C]// Graphics interface. Mississauga, Canada: Canadian Information Processing Society, 1995: 147.
- [20] XIAO B. Research on the integration of land and water terrain data of laser and multi-beam point cloud[D]. Taian: Shandong Agricultural University, 2018.
- [21] ESTER M, KRIEGEL HP, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// 2nd International Conference on Knowledge Discovery and Data Mining. Broadway, New York, USA: ACM Digital Library, 1996: 226-231.
- [22] HU F. Real-time data-driven digital shop floor research based on Unity 3D[D]. Changsha: Hunan University, 2018.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv –Machine translation. Verify with original.