ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Intelligent path planning for two-machine collaborative forestry robot based on high-precision tree point cloud segmentation

Hao Sun[1], *Yanhu Zhu*[1], *Buyong Ren*[1], *Sifan Chen*[1] *and Ziqiang Guo*[1,*]

[1] Linxia Power Supply Company, State Grid Corporation of China, Linxia, Gansu, 731800, China

SUMMARY: *In the structural design of forestry robot harvesting machine, there are very few planning studies that can avoid obstacle movement. In this regard, this paper uses the VelodyneHDL-32E sensor carried by the UAV of DJI FC6320 to scan the trees of the four test sample plots,for the purpose of obtaining the tree point cloud data, and through the method of data pre-processing,so that the data set is free of interference information. With reference to the current mainstream point cloud segmentation methods, the PointNet network model was finally selected for high-precision tree point cloud segmentation,the loss function of the PointNet model was designated as the Softmax cross - entropy loss function, and in order to achieve a more desirable point cloud segmentation performance, the optimal parameters of the model were also determined through the calculation of the gradient. Combined with the characteristics of point cloud segmentation, we propose to use ACO-RRT* to complete the intelligent path planning of two-machine collaborative forestry robot based on point cloud segmentation.The path planned by ACO-RRT* algorithm is smoother than the RRT algorithm and RRT* algorithm, the path cost is smaller, and the optimization effect is better, and the specific enhancement is 67.14% and 15.16%, which verifies the performance of two-machine collaborative forestry robot based on point cloud segmentation. The effectiveness of intelligent path planning of two-machine collaborative forestry robot based on point cloud segmentation is verified. This paper can automatically plan the paths of target stumps and non-target stumps so as to avoid collision, which is of great theoretical and practical significance to improve the overall operational efficiency of forestry machinery.*

KEYWORDS: *ACO-RRT* algorithm; PointNet network model; tree point cloud segmentation; path planning; two-machine collaborative manipulator*

1 Introduction

The application of 3D point cloud technology in forestry is becoming increasingly widespread, providing a revolutionary method for accurately measuring and analyzing tree structural parameters [1]. As a key component of ecosystems, the structural characteristics of trees directly affect the carbon stock, biodiversity and ecosystem function of forests. Traditional methods for measuring tree parameters are time-consuming and limited in accuracy, while LiDAR (laser radar) technology [2] can rapidly acquire high-precision 3D point cloud data, which provides a technological basis for the precise analysis of the structure of tree components. The data which come from point clouds that are got through many platforms, include ground laser scanning (TLS), no man aerial vehicle laser scanning (ULS), and airborne laser scanning

*zzaazz112@163.com
https://doi.org/10.65102/is2026617
Published: 30 April, 2026

(ALS) [3 - 5], have shown that they have very big potential in forest checking and ecological monitoring work. Compared with traditional methods, LiDAR technology is able to non-destructively record the detailed 3D structure of trees, this thing provides very precise close values for parameters, which include breast height diameter (DBH), the tree's height, the spread of the crown, and ground level diameter [6, 7]. Studies have shown that point cloud-based tree parameter extraction is 5-10 times more efficient than traditional methods, while significantly improving data completeness and accuracy [8].

Tree point cloud segmentation is a key preprocessing step for tree parameter inversion, and the research mainly relies on traditional computer vision and point cloud processing algorithms to achieve segmentation by analyzing the geometric features and spatial distribution of the point cloud. These research ways can be divided into three main kinds: ways that are based on geometry characteristics, model fitting methods, and region expanding ways. Methods which are based on geometric features can differentiate many different parts of trees through extracting and examining the local geometric properties that the point cloud has.For example, Xu, M and other persons have proposed a method to extract individual trees which is based on strengthened voxel clustering, which demonstrated high computational efficiency and spatial adaptability when dealing with large-scale point cloud data, and especially showed significant advantages in improving the accuracy of initial trunk localization. Zhang, W et al [10] combined normal vector variation with 3D hyperboloid connectivity analysis to effectively eliminate branch and leaf points and enhance the trunk extraction effect, however, when there is a close fit between low vegetation and the trunk, the local typical geometric features of the trunk points are no longer prominent, which leads to the difficulty of accurately distinguishing the trunk points from the non-trunk points. Yang, B et al. [11] established a tree trunk model by cylindrically fitting the sliced point cloud, which has good performance under the assumption that the tree trunk is a regular cylinder, and is especially suitable for forest stands with sparse tree distribution and regular growth patterns; however, in the case of complex spatial structure of trees or irregular growth patterns, the applicability of this method is obviously limited, especially in the case of severe understory vegetation shading. However, in the case of complex spatial structure or irregular growth pattern of trees, especially when the understory is heavily shaded by vegetation, the applicability of this method is obviously limited. Luo,Z and his work group [12] at the beginning carried out a vertical cutting work on the point cloud. To each slice, they produced pictures which included density, height, and local height gradient. After that, therefore, they have proposed a multi-channel representation method for the purpose of encoding the image data of each individual slice.Finally, by the combination of multi-channel features, a multi-branch network is constructed to realize single-tree segmentation in UAV LiDAR point cloud data.These methods are computationally simple and work well for recognizing regularly shaped tree trunks, but the accuracy is often unsatisfactory when dealing with morphologically complex tree species or the intersection of trunks and branches.

Model fitting methods recognize tree components by fitting a predefined geometric model to the point cloud data. For example, Chen, X et al [13] used PointNet to encode features in voxelized point clouds by first dividing the spatial voxel with the average canopy width, training the network to recognize the canopy point clouds in the voxel, and then further refining the canopy boundaries by incorporating the height gradient information.Li, B and his work group used neural network method to separate tree pixels and non-tree pixels, hence obtaining semantic segmentation masks that are for trees. In addition, they bring forward radial bounding volumes (RBVs) which is a method with fixed dimension for expressing three-dimensional tree models, using neural networks to learn how to map the input semantic segmentation masks of the trees to the RBVs, which can be used to further infer the 3D structure of the trees. Fu, L et al [15] first extracted the initial skeleton from a tree point cloud using an octree segmentation

and level set method, and then used cylindrical shape prior constraints (CPC) optimization and estimated branching radii to correct the location of incorrect joints.Hu, S and his work group have made a direction field on the basis of the growth rule of real trees, for restricting the growth direction of tree branches. They have utilized a bottom-up greedy algorithm that is combined together with a priority queue. Starting from the tail part, they step by step constructed one branch-shaped framework, and arranged the leaves on the branch skeleton based on the leaf order principle, thus generating a 3D model of a tree with a high degree of detail and realism. Su, Y et al [17] proposed an adaptive trunk extraction method by combining a height stratification strategy and region growing, which showed strong robustness in complex forest environments.The method is simple in concept and relatively easy to implement, but the selection of seed points and the setting of stopping criterion often rely on empirical parameters, and have limited generalization ability among different scenarios. Liu, L et al [18] In areas with flatter terrain and regular tree distribution, a canopy height model (CHM) based approach can accomplish the segmentation task more efficiently, whereas in environments with complex terrain and diverse tree species, a direct point cloud based segmentation approach usually yields more accurate results.Zhong, L and other persons [19] have proposed a tree check and cut-apart method which is based on the octree structure. The experiment results showed that the stem checking method which depends on the histogram of octree nodes can distinguish tree trunks from other pole-like features such as lamp posts and walking people. This type of method is simple in concept and relatively easy to implement, but the selection of seed points and the setting of stopping criterion often rely on empirical parameters and have limited generalization ability among different scenes.

In addition, region growing methods have been applied in tree structure segmentation.For the giving example, Ma, Z together with other researchers [20] have proposed a two-stage single tree crown (ITC) dividing method for airborne Light Detection and Ranging (LiDAR) point cloud data. The present method is established upon region expansion and morphological segment division, which combines the advantages of area growth features and detailed morphological structure of the tree canopy.Premebida, C and other colleagues have carried out transformation of the point cloud into an aerial perspective view. From this look taken from the air, the picture element points have connection with one order of checkers inside the real environment, then each pixel is populated with RGB values based on the density, average intensity and height of the point cloud in each grid, and finally the target is captured from the bird's eye view by using image detection techniques.Li, S and his work companions [22] made use of the differing normal distribution characteristics of the leaf point cloud and the branch point cloud of single trees. They utilized the conventional difference operator for accomplishing the recognition of the branch point cloud of one individual tree. The purpose was to carry out the segmentation of the branch point cloud, which therefore reduces the degree to which the branch point cloud influences the accuracy of the extraction of leaf area density.Nuijten, R J and co-workers [23] employed Digital Aerial Photogrammetric (DAP) point clouds which were got by Unmanned Aerial Vehicles (UAVs) for the comparison of the individual tree crown segmentation results in deciduous forests among different seasons. The result of research showed that the maximum precision of single-tree division work obtained was 77.10%, which emphasized the effect of seasonal variations on segmentation results, and these studies showed that DAP point cloud also has good ability to segment single trees. These methods can handle noise and anomalies better,but, there exist problems concerning the segmentation effect and its insufficient stability when point cloud density is not uniform.

The rapid development of deep learning technology provides a new idea for point cloud processing, compared with the traditional method, the tree structure segmentation method based on deep learning has stronger feature expression ability and adaptive. Qi, C R et al [24] first

proposed PointNet++ enhanced the ability to perceive local structures through a layered feature learning strategy, which significantly improved the segmentation accuracy in complex scenarios.The work laid the foundation framework for deep learning of point clouds, but its ability to perceive local geometric structures is still limited when dealing with point clouds of trees.PointNet, as the first direct processing disordered point clouds, a deep neural network, realizes end-to-end point cloud classification and segmentation by processing point sets through multilayer perceptrons and symmetric functions. Subsequently,the PSegNet method which was put forward by Li, D and other people [25] has carried out simplification for the samples in the process of plant structure semantic segmentation. This result was obtained through the employment of the Double Neighborhood Character Acquisition Module (DNFEB) and the Double Granularity Character Integration Unit (DGFFM). Nevertheless, it meets with difficulties in the segmentation of fine details. To speak specifically, it has difficulty when it processes the closely placed or mutually overlapping twigs and leaves that are inside the point cloud. At the same time, Mr. Yang J [26]and his work group, on the other hand, introduced a deep network based on Transformer, fused multispectral features to realize automatic identification of urban tree point clouds, and accomplished segmentation at the single-tree level through morphological operations; however,deep-learning models, in the general case, have the requirement of a big amount of marked data for the process of training. But, the obtaining of large-scale, high-quality marking materials for every single tree is on the one hand time-consuming and on the other hand excessively costly. Zhang, J et al [27] utilized 3D geometric features for initial extraction of tree trunk points, followed by segmentation of tree trunks using the DBSCAN clustering algorithm; however, the clustering-based segmentation method has a simple structure and high computational efficiency, but its segmentation results are sensitive to parameter settings and are susceptible to interference from dense understory shrubs or low vegetation. However, the current tree point cloud segmentation still faces three main challenges.

First of all, the structure which this tree possesses is extremely complicated. One single tree point cloud generally includes tens of thousands of points, thus boundaries exist among the tree trunks, branches and leaves are fuzzy, especially at the bifurcation intersections, which makes it difficult for traditional methods to accurately differentiate between different components. Second, high-quality point cloud annotation requires professional knowledge and a large amount of manpower, and it usually takes 30-60 minutes to complete the annotation of a medium-sized tree, which makes the annotation cost extremely high. Finally, the morphology of different tree species, seasons and growing environments varies greatly, and existing deep learning models lack the ability to adapt to this diversity.

In this paper, Scenic Area B in City C is taken as the study area, and the VelodyneHDL-32E sensor on board the DJI FC6320 UAV is used to scan the seven types of trees in the four experimental sample plots, so as to complete the data collection work.After the point cloud data have been gotten, it is not able to be used right now. On the contrary, extra pre-processing procedures are required in order to ensure the usable degree of the data set. Drawing on several mainstream tree point cloud segmentation methods, the PointNet model was finally adopted to carry out high-precision tree point cloud segmentation,furthermore, the loss function which is used inside the model is the Softmax cross-entropy loss function. In order to maximize the performance of the PointNet model, the optimal parameters of the PointNet model were determined by gradient calculation, aiming at realizing the high-precision tree point cloud segmentation. From the RRT* algorithm, the principle of ant colony algorithm, and the characteristics of point cloud segmentation, it is proposed to use the combined ACO-RRT* algorithm to complete the intelligent path planning work of two-machine collaborative forestry manipulator based on high-precision tree point cloud segmentation, and to verify its validity and reliability through empirical analysis.

2 Intelligent path planning for manipulators based on point cloud segmentation

2.1 Overview of the study area and data acquisition

2.1.1 Overview of the study area

In the research which was done for this paper, the selected study place is located in Scenic Area B of City C. City C, which is affirmed as a national forest city, has a gentle climate that is given the feature of clear four seasons. It also gets an exceedingly large quantity of rainfall and is categorized as a warm and humid subtropical monsoon climate type.City C has an annual average rainfall that changes between 1500 and 2150 millimeters, and its average every year temperature is located at 16.9 Celsius degrees. In the whole year, January is the most cold month, hence July is the most hot. The mean temperature values of these several months are 3.1 degree Celsius and 27.9 degree Celsius separately.Qishan the mountain has total area of 42 square kilometers, its peak height arrives 859 meters. The tree-living population inside this area mainly is made up of seven different kinds of species. Among these plants are included Dawn Redwood, True Fir, Sweet Osmanthus, Maidenhair Tree, Soapberry, Poplar, and Camphor Tree.In this experiment, four kinds of experimental sample plots were selected by us from Scenic Area B, that is the seedling culture base, the temple garden, the mixed forest and the deciduous forest. Experimental experiment sample plots 1, 2 and 4 are placed at the foot of the mountain, while experimental sample plot 3 is located on the hillside on which the hilly land form is not flat. All four of the experimental sample plots are constituted by buildings, shrubs and trees.

2.1.2 Data acquisition

One Velodyne HDL-32E sensing device which was installed on a DJI FC6320 unmanned aerial vehicle (UAV) was utilized by us for conducting experiments. The aim of these experimental works was to collect data from four testing sample regions. The laser scanning mechanism which is inside this system can permit angular changes that are from -30.59° to +10.59°, and it provides a complete 360° horizontal view field.This sensor possesses the capability of generating approximately 700,000 scanned point clouds within every one second, thus it has a measurement accuracy which is plus or minus +/-2 centimeters. Furthermore, this sensor possesses the advantage that it can pass through smoke. Its work environment may cover from -10 degrees Celsius to +60 degrees Celsius, therefore greatly increasing the redundancy of the working conditions.The Velodyne LIDAR system has combined together laser scanning and SLAM technology, for rapidly completing every one scan and producing a high-density point cloud. In the aspect of data collection, the flight speed value is 18 meters each second, the flight height value is 60 meters, and the laser scanning overlapping rate is 40 percent. The finally obtained point cloud masses were preserved in the LAS1.2 format.The average density values of LIDAR point clouds which got from four experiment sample regions, that is the nursery base, the temple garden, the mixed forest, the deciduous forest, were 1511.15 points each square meter, 1002.09 points each square meter, 722.16 points each square meter, 502.17 points each square meter separately.

2.2 Data pre-processing

2.2.1 Removal of noise points

Point cloud data cannot be used directly after acquisition, mainly due to the fact that when LIDAR samples the data, it is easy to mix unreasonable noise points in the real data. These

noise points are abnormal data and cannot be used for subsequent calculations, so they need to be removed.

2.2.2 Point cloud voxelization

Since the four different experimental sample plots studied in this paper contain trees,large manorial buildings, defensive constructions, worship places, and living houses that hold many different kinds of Chinese building arrangement styles, they are not easy to manipulate when classifying point clouds using the PointNet network model. Therefore, we voxelized each of the selected experimental sample plots accordingly, using the average canopy size of each experimental sample plot obtained from the preliminary forest survey to define the size of the voxel, as a way to ensure that each voxel contains as complete a point cloud of trees as possible.

2.2.3 Production of data sets

After every training sample was undergone the process of voxelization, one group of point cloud documents with the form of. The txt files have been generated through our processing. After that, every txt document which belongs to every point cloud was processed by random sampling, thus making them be reduced to 1024 points. After that, in line with the specifications of the PointNet network model, the point clouds within each voxel that made up the training and test sets were converted into the HDF5 file format.

2.3 High-precision tree point cloud segmentation based on PointNet modeling

Referring to related research data and literature, several mainstream tree point cloud segmentation methods are Random Forest (RF), Support Vector Machine (SVM), 2D projection, K Nearest Neighbor Algorithm (KNN), and PointNet model, and ultimately, this paper chooses the PointNet model to carry out high-precision tree point cloud segmentation.

2.3.1 Loss Functions for PointNet Models

During this experiment, the loss function employed for the PointNet model throughout the training phase is the Softmax cross - entropy loss function. Below is the definition of the whole loss function for all voxel point clouds:

$$\begin{aligned} Loss &= -\frac{1}{(N+1)}\sum_{l=0}^{N} L_l + weight_{rege}\cdot L_{reg} \\ &= -\frac{1}{N+1}\sum_{l=0}^{N}\sum_{\zeta=0}^{2}\left(indicate_{\varsigma,l}\cdot\log\left(\hat{y}_l^{\,j}\right)\right) + weight_{rege}\cdot L_{reg} \end{aligned} \tag{1}$$

$$indicate_{\varsigma,l} = \begin{cases} [0,0,1] & The\ first\ class\ \zeta = 0 \\ [0,1,0] & The\ \sec ond\ class\ \zeta = 1 \\ [1,0,0] & The\ third\ class\ \zeta = 2 \end{cases} \tag{2}$$

$$\hat{y}_l^{\,j} = soft\ \max\left(\vec{\eta}_l^{\,j}\right) = \frac{e^{\vec{\eta}_l^{\,j}}}{\sum_{l=0}^{2} e^{\vec{\eta}_l^{\,j}}} \tag{3}$$

$$\vec{\eta}_l^{\,j} = \omega^* \cdot p^j \tag{4}$$

$$L_{reg} = l2\_loss\left(I - AA^T\right) \tag{5}$$

where $indicate_{\varsigma,l}$ is the indicator, which is 1 if the current computational category ζ is the same as the category of the sample category l, and 0 otherwise. on the premise that we have divided the experimental sample into 3 categories, i.e., $N = 2$ and $l = \{0,1,2\}$. In the formula, - represents the point multiplication. $\vec{\eta}_l^{\,j}$ represents the vector probability of the point cloud within the j th voxel obtained by the change of network linear weight w, and *Soft* max represents the corresponding *Soft* max nonlinear activation function. A is the T-Net of feature transformations, L_{reg} is used to constrain the feature transformation matrix, and I is the corresponding unit matrix. The $l2_$loss function represents the summation of each element of the matrix after squaring it and dividing by 2. In this experiment, weight_{regge} is set to 0.0001.

In this experiment, the weight values of the layers of the deep convolutional neural network are continuously renewed through the utilization of a random gradient decrease algorithm. One layer may be regarded as a receiving container which generally accepts inputs that have weights. It then carries out processing on these inputs via a sequence of mostly non-linear functions. Following that step, it sends these handled values as output onward to the following layer.When the loss function that appears in the training procedure falls lower than a certain fixed loss threshold, therefore, the training procedure must be brought to a stop. After that, the weight values of each layer of the network are fixed at their current positions, and hence will not receive further adjustment. This outcome brings about a completely trained deep convolution nerve network.

2.3.2 Calculation of the gradient

In the mathematical definition, the gradient is a vector, the so-called vector can also be referred to as a vector, that is, with a certain direction,this indicates that, at a certain fixed point, the direction derivative of one specific function along a given direction can attain its maximum value, in other words, that is, the function at the defined point along the direction of the fastest change and the rate of change is the largest, then the direction is called the direction of the gradient. The rate of change is the magnitude of the mode of that gradient.

Suppose, the binary function $z = f(x, y)$ has first order continuous partial derivatives on the planar region D, then for each point $P(x, y)$ a vector $\left\{\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right\} = f_x(x, y)\bar{i} + f_y(x, y)\bar{j}$, which is then referred to as the gradient of the function $z = f(x, y)$ at the point $P(x, y)$, denoted by $grad(x, y)$ or $\nabla f(x, y)$, and can be expressed by equation (6):

$$grad(x, y) = \nabla f(x, y) = \left\{\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right\} = f_x(x, y)\bar{i} + f_y(x, y)\bar{j} \tag{6}$$

where $\nabla = \frac{\partial}{\partial x}\bar{i} + \frac{\partial}{\partial y}\bar{j}$ is known as the differential or Hamiltonian operator for two-dimensional vectors and has $\nabla f = \frac{\partial f}{\partial x}\bar{i} + \frac{\partial f}{\partial y}\bar{j}$.

On this basis, and assuming that $e = \{\cos\alpha, \cos\beta\}$ is the unit vector of the binary function $z = f(x, y)$ in the direction l, we have:

$$\begin{aligned} \frac{\partial f}{\partial l} &= \frac{\partial f}{\partial x}\cos\alpha + \frac{\partial f}{\partial y}\cos\beta = \left\{\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right\} \cdot \{\cos\alpha, \cos\beta\} \\ &= grad(x, y) \cdot e = |grad(x, y)||e|\cos|grad(x, y), e| \end{aligned} \tag{7}$$

When the direction l coincides with the direction of the computed gradient, then there is $\cos[grad(x, y), e] = 1$, at which point the directional derivative $\frac{\partial f}{\partial l}$ has a maximum and the maximum is the value of the gradient, i.e.:

$$\left| grad(x, y) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \right| \tag{8}$$

Thus, the function has a maximum rate of change along the direction of the gradient at a point, with the maximum value being the mode of that gradient.

Similarly, generalizing this to ternary functions, and assuming that the ternary function $u = f(x, y, z)$ has first-order continuous partial derivatives in the spatial region G, and that the point $P(x, y, z) \in G$, the vector $\left\{\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right\} = \frac{\partial f}{\partial x}\bar{i} + \frac{\partial f}{\partial y}\bar{j} + \frac{\partial f}{\partial z}\bar{k}$ is the gradient of the function $u = f(x, y, z)$ at the point P, denoted as $\mathrm{grad}f(x, y, z)$ or $\nabla f(x, y, z)$, and it can be expressed by Eq. (9):

$$gradf(x, y, z) = \nabla f(x, y, z) = f_x(x, y)\bar{i} + f_y(x, y)\bar{j} + f_z(x, y)\bar{k} \tag{9}$$

where $\nabla = \frac{\partial}{\partial x}\bar{i} + \frac{\partial}{\partial y}\bar{j} + \frac{\partial}{\partial z}\bar{k}$ is referred to as the three-dimensional vector differential operator or Hamiltonian operator and $\nabla f = \frac{\partial f}{\partial x}\bar{i} + \frac{\partial f}{\partial y}\bar{j} + \frac{\partial f}{\partial z}\bar{k}$, and also when this gradient computation solves for the direction coincides with the direction in which the maximum directional derivative is obtained, at that time, the size of its modulus stands for the highest value of the directional derivative. This numerical magnitude can be expressed by utilization of Equation (10):

$$\left| grad(x, y, z) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 + \left(\frac{\partial f}{\partial z}\right)^2} \right| \tag{10}$$

2.3.3 Realization of monoki splitting

The step flow for checking the arithmetic worked out to reach single wood cut division includes the below main working steps. At the beginning, by means of carrying out voxelization on the three-dimensional point cloud, the point cloud of every test sample location is assigned into voxels which have continuous distribution. Then, the PointNet network model is used to identify trees in the finely segmented point cloud of each voxel, and the learned parameters are iteratively updated through the training phase to obtain the optimal parameters, which include the tree structure, the number of sampling points, and the size of the voxel edges, and then the model analyzes these trained parameters,in the testing stage, by this way we can obtain the classification results which are related to the point cloud of each individual voxel. These are the classification outcomes that belong to the point cloud of each individual voxel. Finally, for the voxels recognized as trees, we refine the segmentation of the canopy boundaries based on the height-dependent gradient information, which is not constrained by the voxel boundaries on the point cloud segmentation.

2.3.4 Assessment of indicators

The evaluation metrics for tree point cloud segmentation performance analysis use the overall classification accuracy (OA), which can directly evaluate the proportion of correct classifications. The definition into the formula (11) is shown:

$$Overall\ Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{11}$$

where TP the number of samples that we expect to show positive result is the number that positive samples have., FP it points to the number of samples which are forecast to show positive among the samples which are in fact negative, FN the number of specimens which are expected to give negative results among those specimens which are truly positive, and TN the number of specimens which are expected to show negative results belongs to the negative specimens.

Considering the imbalance of data, the classification accuracy of tree species with fewer test samples has a lower impact on OA, resulting in higher OA may also have the problem of low classification accuracy of a certain type of tree. Here the consistency test index Kappa coefficient is introduced to correct the OA, and when there exists a certain class of trees with low classification accuracy, the Kappa coefficient will be significantly lower, which is defined as shown in Equation (12):

$$p_e = \frac{\sum_{j=1}^{C} T_j \times P_j}{N^2}, Kappa = \frac{accuracy - p_e}{1 - p_e} \tag{12}$$

where T_j is the actual number of samples for category j and P_j is the predicted number of samples for category j.

2.4 Path planning based on point cloud segmentation

Through incorporating the characteristics of point cloud segmentation, this paper integrates the RRT* algorithm together with the ant colony algorithm and hence puts forward a heuristic search algorithm that is named ACO - RRT*,it utilizes the global searching capability that the ant colony algorithm possesses to conduct direction for the local searching work of the RRT*

algorithm. Through this method, it can more quickly find the overall optimal route and therefore finally realize the intelligent route planning for a two-machine cooperation forestry robot.

2.4.1 Principles of RRT* and Ant Colony Algorithms

When the RRT algorithm is used to solve the intelligent path planning problem of two-machine collaborative forestry robotic arm in the tree point cloud segmentation map, it is found that the RRT algorithm still has some drawbacks, such as low node utilization and weak stability of generated paths. In view of the problems of RRT algorithm, RRT* algorithm is mainly improved from two aspects, which are reselecting parent nodes and search tree rewiring.

Reselecting parent nodes and rewiring are important improvements made by the RRT* algorithm over the classical RRT algorithm. The algorithm makes the path cost of the newly generated nodes in the expansion process of the random tree smaller and generates better paths. At a certain moment t of the ant colony algorithm, the probability of transferring ant k from node i to node j is set as P_{ij}^{k}, and the process is calculated as shown in equation (13). That is:

$$P_{ij}^{k}=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\partial}\cdot[\eta_{ij}(t)]^{\beta}}{\sum\limits_{s\in allow_k}\left([\tau_{is}(t)]^{\partial}\cdot[\eta_{is}(t)]^{\beta}\right)}; & s\in allow_k\\ 0; & s\notin allow_k\end{cases} \tag{13}$$

In Eq. (13), $\eta_{ij}(t)$ represents the size of the heuristic value for the ant to transfer from the current node to the next node; $allow_k\left(k=1,2,.....m\right)$ denotes the set of nodes not selected by the ants; ∂ denotes the degree of influence of the pheromone concentration during the transfer process of the ants, the larger the value, the greater the influence; and β denotes the degree of influence of the heuristic function. The rules which are used for the updating of pheromones in the ant colony algorithm are given in the equations (14) and (15). That is:

$$\tau_{ij}^{k}(t+1)=(1-\rho)\tau_{ij}^{k}(t)+\rho\Delta\tau_{ij} \tag{14}$$

$$\Delta\tau_{ij}^{k}(t)=\frac{K}{\min\left(\left\{length(m)\right\}\right)} \tag{15}$$

Equation (14) represents the value of pheromone concentration at a node of an ant's pathway, where the pheromone concentration volatilization factor is $\rho\left(0<\rho<1\right)$. Equation (15) represents the pheromone increment, where $length\left(m\right)$ denotes the set of path lengths searched by m ants.

2.4.2 ACO-RRT* algorithmic framework

Aiming at the RRT* algorithm in the two-machine collaborative forestry robotic arm intelligent path planning process there are randomness search, path generation is not uniform and other problems, this paper proposes a fusion of ant colony algorithm heuristic search algorithm ACO-RRT*, the improvement of the algorithm mainly has the following two points:

(1) Node expansion. This paper improves the traditional RRT* algorithm to randomly expand the nodes, introduces the ant colony algorithm when expanding the nodes, uses the

pheromone to describe the distance relationship between the node and the end point in order to construct the dynamic pheromone volatility factor,according to the results obtained from the search which is carried out by the ant colony algorithm, it selects the following path node to carry out expansion. The doing of this is for the enhancement of the local search ability of the RRT* algorithm.

(2) Collision detection. In order to adapt to the 3D point cloud, this paper adds a neighborhood lookup before collision detection to access whether the nodes in the octree are occupied or not, after which collision detection is carried out using the enclosing box method.

2.4.3 ACO-RRT* algorithm node extensions

Aiming at the problems of RRT* algorithm in node expansion, the ACO-RRT* algorithm proposed in this paper takes the pheromone in the ant colony algorithm as a kind of evaluation index for expanding nodes, and constructs a heuristic function to guide the growth direction of the tree, so as to obtain a better path node.The main process of node expansion in the ACO-RRT* algorithm is as follows:

(1) Initialization

The start point is added to the search tree as the root node and the pheromone values of all nodes are initialized to k, i.e., each node in the search tree of the RRT* algorithm maintains a pheromone concentration value.

(2) Pheromone updating

In the process of pheromone updating, this paper assumes that the direction of ant movement should converge to the position close to the end point, so according to the cost of the path from node X_n to the end point, $D\mathrm{Cost}(X_n)$, to dynamically determine the weight of the pheromone $phe(X_n)$ reduced weight, i.e., the pheromone volatilization factor ρ, and update the pheromone value of the node based on the pheromone increase Δphe of the node that the ant passes through. At the beginning of each iteration, the path cost $\mathrm{Cost}(X_n)$ from the starting point to the node X_n in the tree is calculated, and then the inverse of the path cost is utilized as the pheromone-increase weight, i.e., the smaller the path cost node is, the larger the pheromone-increase weight it has. The specific formulas for the above process are shown in Eqs. (16) to (20). That is:

$$D\mathrm{Cost}(X_n)=\sqrt{(x_n-x_d)^2+(y_n-y_d)^2+(z_n-z_d)^2} \tag{16}$$

$$\rho=\frac{2}{\lambda\cdot\pi}\cdot\arctan D\mathrm{Cost}(X_n) \tag{17}$$

$$\begin{aligned}\mathrm{cost}(X_n)=&\sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}+\cdots\\&+\sqrt{(x_{n-1}-x_n)^2+(y_{n-1}-y_n)^2+(z_{n-1}-z_n)^2}\cdots\end{aligned} \tag{18}$$

$$\Delta phe(X_n)=\frac{Q}{\mathrm{cost}(X_n)} \tag{19}$$

$$phe(X_n)=(1-\rho)*phe(X_n)+\rho*\Delta phe(X_n) \tag{20}$$

In Eq. (18), λ is the parameter that determines the impact of the path cost from the target node to the end point. In Eq. (20), Q is the pheromone constant, which is the total amount of pheromone released by the ants. Eq. (19) is the pheromone update formula.

(3) Expanding nodes

The probability $P(X_n)$ of each node in the ant colony algorithm to be selected as the next node of the ant is related to its inspired value $h(X_n)$, and the formulas for the inspired value and the probability of being selected are shown in Eqs. (21) and (22). That is:

$$h(X_n) = phe(X_n)^{\partial} \cdot \left(\frac{1}{\cos t(X_{start}, X_n)} \right)^{\beta} \tag{21}$$

$$P(X_n) = \frac{hX_n}{\sum_1^m h(X^i)} (i = 1, 2, \cdots, m) \tag{22}$$

In Eq. (19), the heuristic value $h(X_n)$ is the weighting of the pheromone value and the path cost, and $cost(X_{start}, X_n)$ is the path cost between the start point X_{start} and the node X_n. In Eq. (20), m is the number of optional nodes for the ants' next transfer, and X^i denotes the optional nodes for the ants' next transfer. Through the constant iteration of the ant colony algorithm, the optimal path is obtained to guide the local search of the RRT* algorithm, and the end point of the path is the node to be expanded. Suppose the current expanded node is X_n, its parent node is X_{parent}, the new parent node is X_0, the path cost from node to node is written as $\cos t_{old}$, and the path cost from node to node is written as $\cos t_{new}$, at that time, then the parent node is re-selected, and the main update of the node and the parent node's path cost is performed , specifically as shown in equation (23). That is:

$$\cos t(X_n) = \cos t_{new} \tag{23}$$

After that the path cost of the node's children is updated. For each child node X_m of the node, if the path cost is less than the path cost, i.e., $\cos t(X_m) < \cos t(X_n)$, then the parent of the node can be changed to, and the cost of the node and its children can be updated as shown in equation (24). That is:

$$\cos t(X_m) = \cos t(X_n) + \cos t(X_m, X_n) \tag{24}$$

In Eq. (24), $\cos t(X_m, X_n)$ denotes the cost of the path to. Continuously repeating the pheromone updating and node selection process, the extended node part of the ACO-RRT* algorithm can be accomplished to be able to obtain better path nodes to construct the search tree.

2.4.4 ACO-RRT* Algorithm Collision Detection

In order to adapt to the tree point cloud segmentation map, the ACO-RRT* algorithm proposed in this paper joins the neighborhood lookup before collision detection, searches for the free node in the octree, and then conducts collision detection for the node in order to carry out the

subsequent path planning, and finally calculates the cost of the path to provide a basis for the selection of the path.

Figure 1 shows the schematic diagram of the AABB box, assuming that there exist objects A and objects B, when using the AABB box for collision detection, the smallest vertex of object A is denoted as $(X_{a\min}, Y_{a\min}, Z_{a\min})$, and the largest vertex is denoted as $(X_{a\max}, Y_{a\max}, Z_{a\max})$; similarly, the minimum vertex of the object B is denoted as $(X_{b\min}, Y_{b\min}, Z_{b\min})$ and the maximum vertex as $(X_{b\max}, Y_{b\max}, Z_{b\max})$. When performing collision detection, as long as any one of the constraints in Eq. (25) is satisfied, no collision will occur, otherwise it will occur. That is:

$$\begin{cases} X_{a\min} > X_{b\max} \\ X_{b\min} > X_{a\max} \\ Y_{a\min} > Y_{b\max} \\ Y_{b\min} > Y_{a\max} \\ Z_{a\min} > Z_{b\max} \\ Z_{b\min} > Z_{a\max} \end{cases} \tag{25}$$

The obstacles in both the robotic arm and the tree point cloud segmentation maps are represented using a wraparound box. Assume that the robotic arm is a cube of size K with coordinates (X, Y, Z); the obstacle is a cube of size L with center point (X_0, Y_0, Z_0). The condition that the robotic arm does not collide with the obstacle is shown in equation (26). That is:

$$\max\left\{\left|X - X_0\right|, \left|Y - Y_0\right|, \left|Z - Z_0\right|\right\} > 0.5\left(K + L\right) \tag{26}$$

If there is an overlapping portion, it means that the robotic arm has collided with that point cloud region, and the path of that node is set to infinity, i.e., it is marked as an obstacle, and the point is no longer expanded as a new node. If there is no overlapping part, it means that there is no collision, and the traversal of the child nodes of this node can be continued until all the nodes with possible collision have been traversed.

The process of expanding nodes and collision detection is continuously carried out in the ACO-RRT* algorithm to search for a better path without collision from the starting point to the end point.

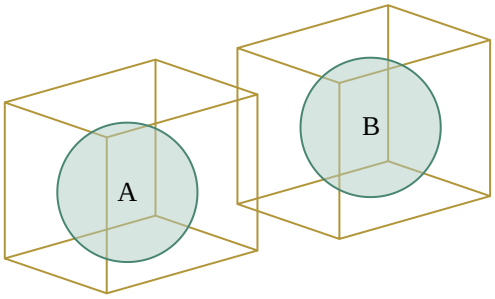


Figure 1: Schematic diagram of AABB enclosure box

3 Analysis of empirical studies

3.1 Empirical analysis of high-precision tree point cloud segmentation

Within this research paper, we initially assess the efficacy of the loss function within the PointNet model via an in - depth analysis of the PointNet model's training process. Subsequently, under the theoretical guidance of evaluation indexes, we carry out the analysis of the influence of different tree structures, the influence of the number of sampling points, and the influence of the size of voxel edges, and then determine the optimal parameters of the PointNet model. Based on this, a comparative assessment of the overall performance of various tree point cloud segmentation techniques is conducted to illustrate the superiority of the PointNet model in high - accuracy tree point cloud segmentation.

3.1.1 PointNet model training analysis

The steps of PointNet model training are as follows: firstly, the training set of tree segmentation point cloud is input into the network, and the prediction vectors of the network structure are obtained by computing them according to the network structure. Calculate the prediction vector and the segmentation label content by loss function to get the loss value,make use of the gradient descending approach to carry out modification on the parameters of the network. This process is repeated until the loss function converges. Among them, the gradient descent method with small batches divides the whole dataset into multiple batches equally to facilitate parallel operation of the computer, improve the accuracy of the descent direction and reduce the training shock, and the size of the batch is determined by the number of batches. Only one batch of data is input for optimization each time during the training process, and all batches of the entire dataset are called 1 round after the training is completed.When we make use of the gradient descent algorithm to carry out optimization work, the gradient item α is being multiplied by one learning rate. In the case that the learning rate is excessively tiny, the convergence procedure will be extremely slow. On the opposite side, when the learning rate has too big a value, it is able to cause the loss function to produce vibration.Within this research paper, the training process of the PointNet model is carried out. At the beginning stage, the value of learning rate is set as 0.0001, and the value of batch size is fixed at 16, and an epoch of 200, and the performance of the validation set is evaluated every 1 round. The best set is selected among the results of multiple training rounds,Figure 2 shows the moving track of the loss function's value that is drawn against the quantity of iteration times. The training set has a more pronounced increase at the position where the number of iterations is around 20 and 105, and continues to decrease within its respective number of iterations, showing an overall decreasing trend. The test set loss values are somewhat oscillating compared to the training set, and its overall trend is more consistent with the training set.

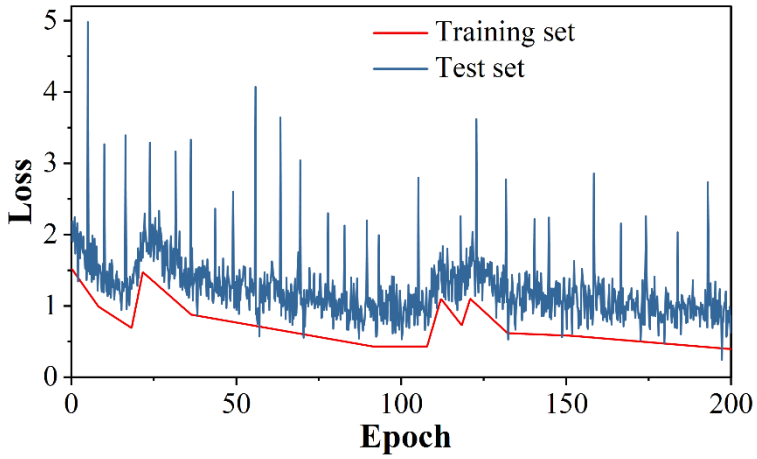


Figure 2: The value of the loss function varies with the number of iterations

Figure 3 shows that with the change of iteration number, the accuracy changing trends of the training set and the test set are presented. Through analyzing the data which is shown in the figure, we can clearly see that the peak accuracy of the model's test set is obtained at 197 iteration times, hence it reaches an accuracy value of 0.863.

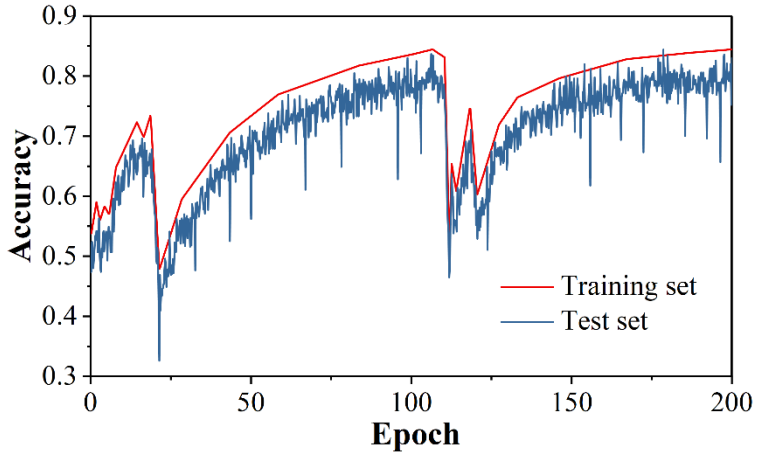


Figure 3: The curve of accuracy rate varying with the number of iterations

3.1.2 Impact of different tree structures

The PointNet model proposed in this paper needs to take into account the structure of the crown, the main trunk as well as the whole tree, and in order to verify the contribution of each structure to the final point cloud segmentation results, Table 1 demonstrates the ablation experiments under different structural branches, where A1 to A7 denote, respectively, the sequoia, the fir, the incense tree, the ginkgo, the sapindus, the poplar, and the camphor tree. It can be seen that learning only the trunk structure gives the worst results because tree trunks are poorly differentiated in most cases, and considering both the trunk and the crown slightly outperforms the results of including only the overall structure, due to the fact that the former prompts the network to focus more on and merge multiple localized regions. The best results were achieved with the combination of holistic and localized approaches, which maintained values above 0.8, demonstrating the influence of crown, trunk, and whole tree structure on the point cloud segmentation performance of the PointNet model.

Table 1: Ablation experiments under different structural branches

Main trunk	Tree crown	Whole	A1	A2	A3	A4	A5	A6	A7	OA	Kappa
√			0.6418	0.6017	0.6411	0.6695	0.6748	0.6268	0.6535	0.6305	0.6232
	√		0.6645	0.6064	0.6704	0.7131	0.6937	0.7669	0.7662	0.7153	0.624
		√	0.7076	0.6067	0.7788	0.7341	0.8235	0.7826	0.7786	0.7442	0.7389
√	√		0.7961	0.6186	0.7913	0.7587	0.8792	0.8345	0.8083	0.79	0.8074
√	√	√	0.8192	0.7471	0.8691	0.8673	0.8865	0.8696	0.8613	0.8158	0.8254

3.1.3 Impact of the number of sampling points

The preceding section delved into the impact of tree structural parameters on the PointNet model. Subsequently, the next section conducted an analysis of the effect of the quantity of sampling points on the PointNet model. The influence of the number of sampling points is presented in Figure 4, which also depicts the effects of the sampling point quantity on the point cloud segmentation outcomes of the PointNet model.When the quantity of sampling points amounts to 512, the accuracy of point cloud segmentation for the PointNet model remains within the range of 0.5 to 0.6. In contrast, when the number of sampling points reaches 1024, the point cloud segmentation accuracy of the PointNet model lies between 0.7 and 0.8. Moreover, when the number of sampling points exceeds 1024, the point cloud segmentation accuracy of the PointNet model stays at a level of 0.7 to 0.8.Once the quantity of sampling points surpasses 1024, the accuracy rate of point cloud segmentation for the PointNet model demonstrates a downward tendency, with its value spanning from 0.6 to 0.7.

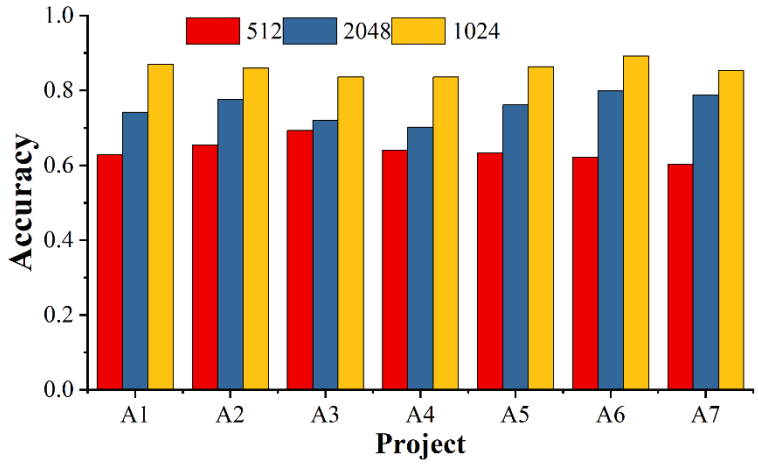


Figure 4: The influence of the number of sampling points

3.1.4 Effect of voxel side length size

In this section, the effect of voxel downsampling on model accuracy is analyzed by setting the voxel block side length size to 2cm, 4cm, and 8cm voxels, respectively, and the effect of voxel side length size is shown in Figure 5. Based on the numerical performance in the figure, it can be seen that when the voxel side length size is set to 2cm and 4cm, the model's point cloud segmentation accuracy for sequoia, fir, incense˛tree, ginkgo, Sapindus, poplar, and camphor is 0.6~0.8, and when the voxel side length size is set to 8cm, the model's point cloud segmentation accuracy for sequoia, fir, incense˛tree, ginkgo, Sapindus, poplar, and camphor is 0.6~0.8, and the model's voxel side length size is set to 8cm. , poplar, and camphor point cloud segmentation accuracy rate is 0.8~0.9, showing the impact of the voxel edge length dimension on the PointNet model.

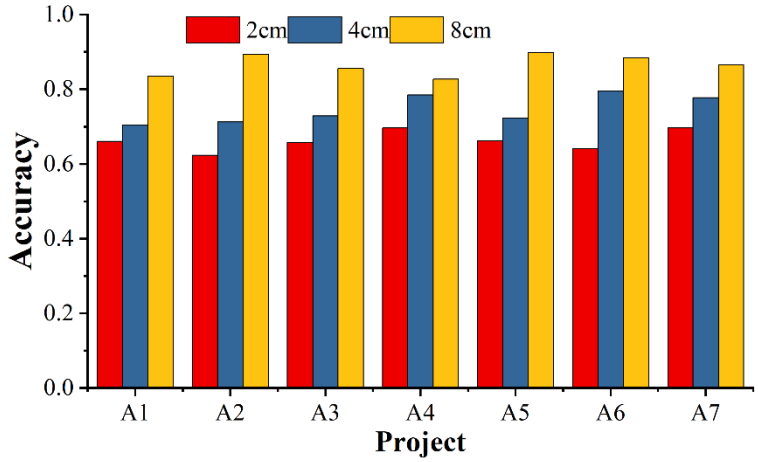


Figure 5: The influence of voxel side length size

3.1.5 Analysis of the overall model performance

Through the above - mentioned analysis of the impact of various tree structures, the effect of voxel side - length dimensions, and the influence of the quantity of sampling points, the ideal parameters of the PointNet model are established. To demonstrate the superiority of the PointNet model in high - accuracy tree point cloud segmentation, an assessment of the model's overall performance is conducted.At present, there are some widely existing tree point cloud segmentation technologies, they are random forest (RF), support vector machine (SVM), two-dimensional projection, and K nearest neighbor algorithm (KNN). Table 2 shows the working effect of various models on tree point cloud segmentation work. Based on the data performance in the table, it can be seen that the accuracy of PointNet model for point cloud segmentation of sequoia, fir, incense tree, ginkgo, sapindus, poplar, and camphor tree is greater than 0.8, and also the Kappa coefficient is greater than 0.8, which is much better than the other four tree point cloud segmentation methods,exhibiting the high - accuracy point cloud segmentation of trees by the PointNet model.Prioritization.

Table 2: The segmentation performance of tree point clouds in different models

Method	A1	A2	A3	A4	A5	A6	A7	OA	Kappa
RF	0.6205	0.61	0.6455	0.7386	0.6213	0.6416	0.6794	0.6987	0.6235
SVM	0.7265	0.632	0.6627	0.8248	0.6542	0.6533	0.7054	0.7114	0.7346
2D projection	0.7497	0.6583	0.7607	0.8548	0.6822	0.7953	0.7325	0.7961	0.8112
KNN	0.7703	0.6847	0.8714	0.8577	0.8054	0.8507	0.7423	0.8128	0.8767
PointNet	0.8894	0.8981	0.8976	0.8664	0.8444	0.8618	0.7563	0.8532	0.8821

3.2 Empirical analysis of intelligent path planning for robotic arms

Through the above analysis, it is known that the effectiveness of high-precision tree point cloud segmentation based on PointNet model, on this basis, this subsection, in order to verify the reliability and superiority of intelligent path planning of two-machine collaborative forestry robotic arm based on point cloud segmentation (ACO-RRT*), the robotic arm platform which is considered to be a prime point is placed in an obstacle-filled environment to conduct the path planning experiments and the optimization performance of the ACO-RRT algorithm , RRT algorithm and RRT* algorithm are compared and analyzed for optimization performance.

3.2.1 Environmental settings

The robotic arm intelligent path planning environment is set up as shown in Fig. 6, the obstacles in this topic are set up as black circles with different radii, the red lines mark the branches of the exploring tree, the green lines represent the branches exploring outward, the red coordinates D(0,0) and E(15.87,14.92) represent the starting point and the end point of this search, respectively, and the black triangles marked near the end point indicate the random sampling points.

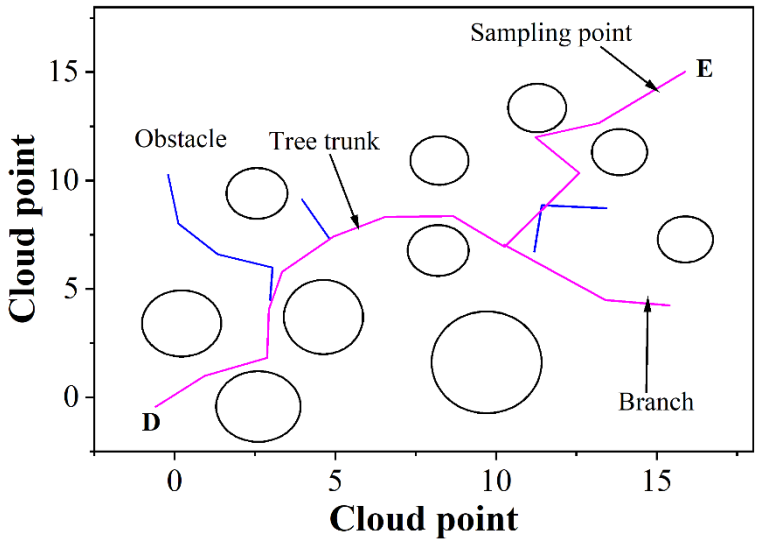
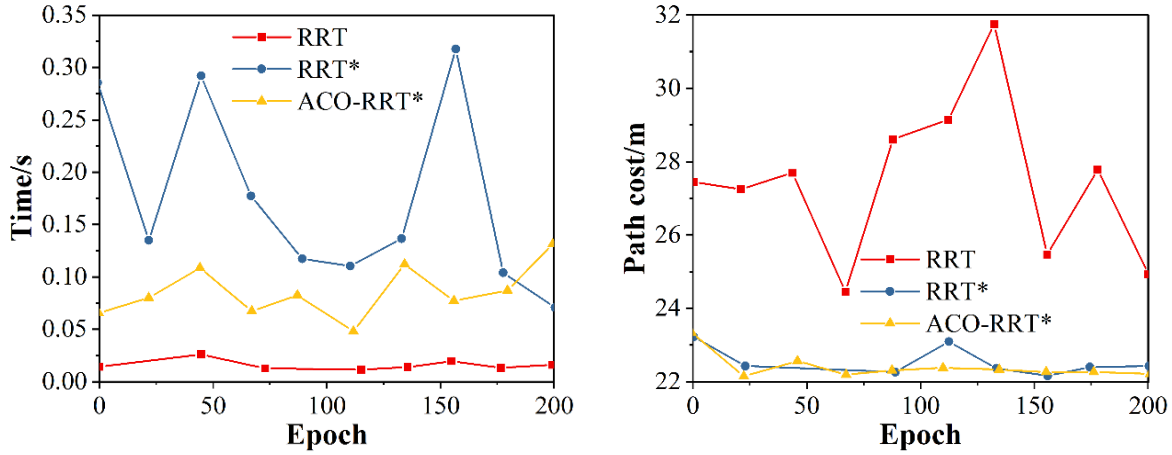


Figure 6: Environment Settings for intelligent path planning of robotic arms

3.2.2 Analysis of simulation results

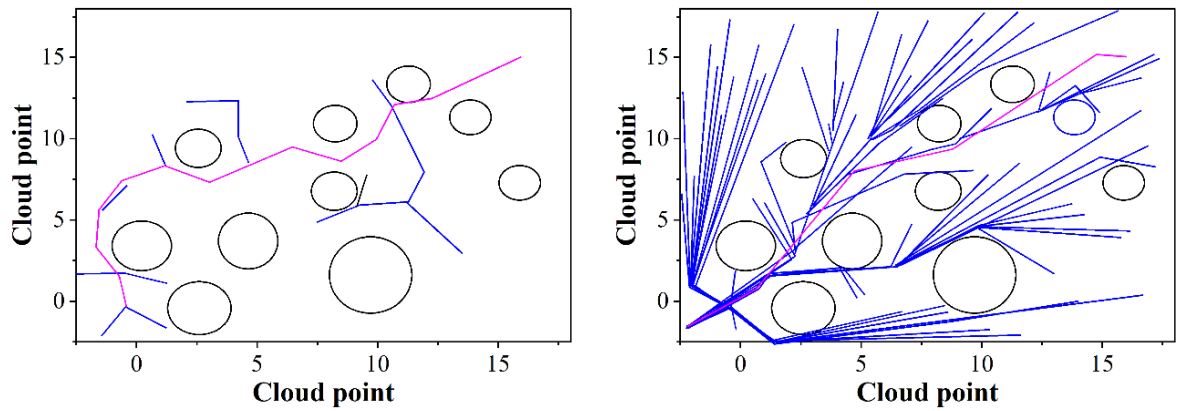
Through the derivation of the formula and theoretical analysis, it is concluded that the number of iterations has a great impact on the performance of the algorithm.For evaluating the degree of the influence which the number of iteration exercises upon the algorithm, this paper analyzes the relationship between the number of iterations and the first path time-consuming and the cost of the optimal path through experiments, and the results of the first path time-consuming and the cost of the optimal path are shown in Fig. 7, in which (a) ~ (b) are the first path, the optimal path, respectively. In order to maintain the accuracy of the experimental data, the experiments were carried out 200 times for each iteration of data for each algorithm for comprehensive analysis. By analyzing the data in Fig. 7(a), it is concluded that the RRT algorithm searches the first path 11.09 times and 5.04 times faster than RRT* and ACO-RRT* without the influence of the steps of choosing parent nodes, rewiring and ACO initialization. Even though the ACO-RRT algorithm has problems such as ACO initialization and probability distribution selection, there is a 1.04 times improvement with respect to the RRT* algorithm. Figure 7(b) can be concluded that although the RRT algorithm can find a path very quickly, there is a big gap between the superiority of the path relative to the other two algorithms. By comparing the cost function values of the data curves, the ACO-RRT* algorithm improves 22.69% relative to the RRT algorithm, which is comparable to the RRT* algorithm, which has a better optimization performance.The ACO-RRT* algorithm maintains the optimal path quality while maintaining a very high solution speed. Through comprehensive analysis, the simulation experiment shows high comprehensive strength of the three algorithms when the number of iterations is 80, no matter in searching for the first path or obtaining the optimal path.



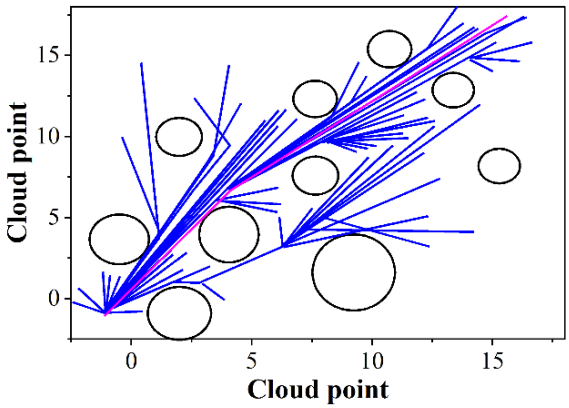
(a) The first path (b) Optimal path

Figure 7: The result of the time consumption of the first path and the optimal path cost

The simulation curves of the same obstacle environment are obtained according to the parameter settings, and the simulation planning of the three algorithms under 80 iterations is shown in Fig. 8, where (a) ~ (c) are the RRT algorithm, the RRT* algorithm, and the ACO-RRT* algorithm, respectively.The optimal paths of the ACO-RRT* algorithm have been strengthened relative to the RRT* algorithm, and there is a great improvement relative to the RRT algorithm. By observing the branches of the exploration tree, the sampling points of the RRT algorithm suffer from the problems of sparse distribution, scattered exploration range and uncorrectable search path. It can be clearly observed that RRT* has a wide range of search points, but more sample points are consumed in exploring useless areas, which limits the overall search efficiency.The ACO-RRT* algorithm constantly corrects the searched paths by weighing the exploration utility so that all the branches will have a tendency to grow towards the end. Closer inspection also reveals that a small number of sampling points are distributed in unexplored regions, which ensures that the samples are always sampled in an infinite domain, and even in the most extreme environments, the variance of the distribution of all ants is greater than 0. This achieves a complete sampling of the state space, which makes the solution results always approach the optimal solution, demonstrating the effectiveness of ACO-RRT*.



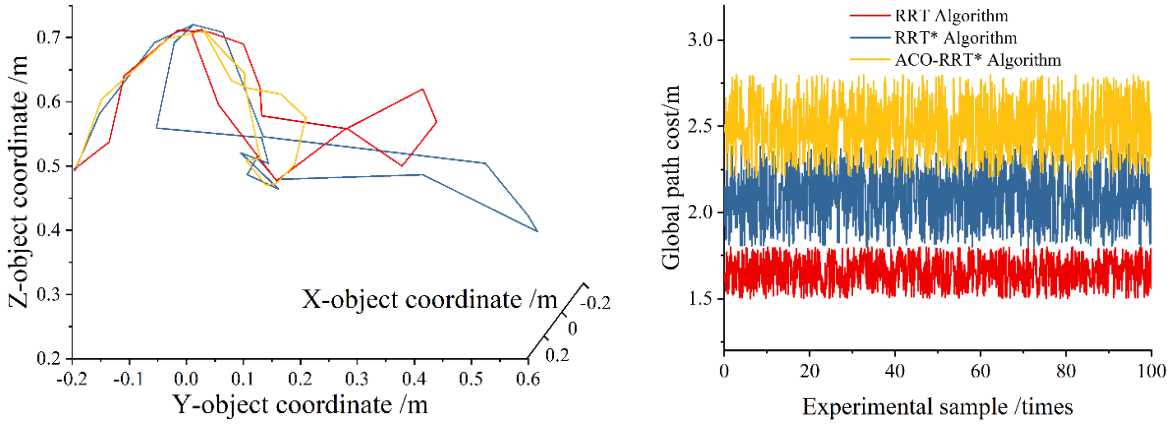
(a) RRT algorithm (b) RRT* algorithm



(c) ACO-RRT* algorithm

Figure 8: Simulation planning of three algorithms after 80 iterations

The results of the comparative analysis of the three algorithms path planning cost data are shown in Fig. 9, where (a) ~ (b) are the paths planned by the three algorithms and the distribution of the path cost of repeated experiments, respectively.For the more clear showing of the route-planning result of the mechanical arm under the three algorithms, the corresponding movement track points of these three algorithms are placed in the point cloud space. Observing the path trajectories of the three algorithms in Fig. 9(a), the paths planned by the ACO-RRT* algorithm are smoother than the RRT algorithm and the RRT* algorithm, the path cost is smaller, and the optimization effect is better. In the same experimental environment of the three algorithms for 100 times repeated experiments, the optimal path cost experimental data obtained for the comparative analysis shown in Fig. 9 (b), it is concluded that the average cost of the ACO-RRT* algorithm relative to the RRT algorithm and RRT* algorithm have 67.14% and 15.16% improvement, respectively. The path optimization performance of the ACO-RRT algorithm is fully proved to be greatly improved compared with the other two algorithms through the comparison experiments of path planning of two-machine collaborative forestry robotic arms.



(a) The paths planned by three algorithms (b) Path cost distribution

Figure 9: Comparison of path planning cost data of three algorithms

4 Conclusion

In this paper, the landscape area is taken as this research area, and the VelodyneHDL-32E sensor

carried by the UAV of DJI FC6320 is used to acquire the tree point cloud data, and the data set is constructed through data preprocessing. Referring to the current mainstream tree point cloud segmentation methods such as Random Forest (RF), Support Vector Machine (SVM), 2D projection, K Nearest Neighbor Algorithm (KNN), and PointNet model, the PointNet model was finally adopted to carry out high-precision tree point cloud segmentation. On this basis, the ACO-RRT algorithm is used to construct a two-machine collaborative forestry robot intelligent path planning scheme based on high-precision tree point cloud segmentation, and the scheme is empirically analyzed.

(1) Compared with the other four tree point cloud segmentation methods, the accuracy of PointNet model for point cloud segmentation of sequoia, fir, incense tree, ginkgo, Sapindus, poplar, and camphor tree is greater than 0.8, which indicates that the PointNet model has a higher priority in high-precision tree point cloud segmentation.

(2) In the intelligent path planning process of two-machine collaborative forestry robot, compared with RRT algorithm and RRT* algorithm, the average cost of ACO-RRT* algorithm is improved by 67.14% and 15.16%, which comprehensively verifies the reliability of ACO-RRT algorithm in the intelligent path planning of two-machine collaborative forestry robot based on high-precision tree point cloud segmentation.

Funding

This work was supported by the Linxia Power Supply Company, State Grid Corporation of China (under grant B727142400XL).

About the Author

Hao Sun, male, bachelor's degree. He is currently an associate engineer at State Grid Linxia Power Supply Company. His research interests include power system protection and control, relay setting optimization, and fault location algorithms in distribution networks.

Yanhu Zhu, male, bachelor's degree. His current research interests include smart grid monitoring systems, application of IoT in power equipment management, and data-driven fault prediction techniques.

Buyong Ren, male, bachelor's degree. He has been working with State Grid Linxia Power Supply Company, where he serves as an associate senior engineer. His current research focuses on emergency repair strategies, fault diagnosis in distribution networks, and improving power supply continuity.

Sifan Chen, male, master's degree. He is currently an engineer at State Grid Linxia Power Supply Company. His research interests include power grid reconstruction after natural disasters, smart distribution systems, and power system planning.

Ziqiang Guo, male, master's degree, currently working as a senior engineer at State Grid Linxia Power Supply Company. Her research focuses on integration of renewable energy sources, grid-connected technologies for photovoltaic systems, and power quality analysis in distributed generation networks.

References

[1] Kim, J. S., Sung, S. M., Back, K. S., & Lee, Y. S. (2024). Accuracy Assessment of Advanced Laser Scanner Technologies for Forest Survey Based on Three-Dimensional Point Cloud Data. Sustainability, 16(23), 10636.

[2] Lokugam Hewage, C. N., Laefer, D. F., Vo, A. V., Le-Khac, N. A., & Bertolotto, M. (2022). Scalability and performance of LiDAR point cloud data management systems: A state-of-the-art review. Remote Sensing, 14(20), 5277.

[3] Wu, C., Yuan, Y., Tang, Y., & Tian, B. (2021). Application of terrestrial laser scanning (TLS) in the architecture, engineering and construction (AEC) industry. Sensors, 22(1), 265.

[4] Pilarska, M., Ostrowski, W., Bakuła, K., Górski, K., & Kurczyński, Z. (2016). The potential of light laser scanners developed for unmanned aerial vehicles-the review and accuracy. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42.

[5] Dai, W., Guan, Q., Cai, S., Liu, R., Chen, R., Liu, Q., ... & Dong, Z. (2022). A comparison of the performances of Unmanned-Aerial-Vehicle (UAV) and terrestrial laser scanning for forest plot canopy cover estimation in Pinus massoniana forests. Remote Sensing, 14(5), 1188.

[6] Burt, A., Disney, M., & Calders, K. (2019). Extracting individual trees from lidar point clouds using treeseg. Methods in Ecology and Evolution, 10(3), 438-445.

[7] Itakura, K., Miyatani, S., & Hosoi, F. (2021). Estimating tree structural parameters via automatic tree segmentation from LiDAR point cloud data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 15, 555-564.

[8] Kim, D. H., Ko, C. U., Kim, D. G., Kang, J. T., Park, J. M., & Cho, H. J. (2023). Automated segmentation of individual tree structures using deep learning over LiDAR point cloud data. Forests, 14(6), 1159.

[9] Xu, M., Zhong, X., & Zhong, R. (2025). A method for automatic extraction and individual segmentation of urban street trees from laser point clouds. Optics & Laser Technology, 180, 111431.

[10] Zhang, W., Wan, P., Wang, T., Cai, S., Chen, Y., Jin, X., & Yan, G. (2019). A novel approach for the detection of standing tree stems from plot-level terrestrial laser scanning data. Remote sensing, 11(2), 211.

[11] Yang, B., Dai, W., Dong, Z., & Liu, Y. (2016). Automatic forest mapping at individual tree levels from terrestrial laser scanning point clouds with a hierarchical minimum cut method. Remote sensing, 8(5), 372.

[12] Luo, Z., Zhang, Z., Li, W., Chen, Y., Wang, C., Nurunnabi, A. A. M., & Li, J. (2021). Detection of individual trees in UAV LiDAR point clouds using a deep learning framework based on multichannel representation. IEEE Transactions on Geoscience and Remote Sensing, 60, 1-15.

[13] Chen, X., Jiang, K., Zhu, Y., Wang, X., & Yun, T. (2021). Individual tree crown segmentation directly from UAV-borne LiDAR data using the PointNet of deep learning. Forests, 12(2), 131.

[14] Li, B., Kałużny, J., Klein, J., Michels, D. L., Pałubicki, W., Benes, B., & Pirk, S. (2021).

Learning to reconstruct botanical trees from single images. ACM Transactions on Graphics (TOG), 40(6), 1-15.

[15] Fu, L., Liu, J., Zhou, J., Zhang, M., & Lin, Y. (2020). Tree skeletonization for raw point cloud exploiting cylindrical shape prior. Ieee Access, 8, 27327-27341.

[16] Hu, S., Li, Z., Zhang, Z., He, D., & Wimmer, M. (2017). Efficient tree modeling from airborne LiDAR point clouds. Computers & Graphics, 67, 1-13.

[17] Su, Y., Ma, Q., & Guo, Q. (2017). Fine-resolution forest tree height estimation across the Sierra Nevada through the integration of spaceborne LiDAR, airborne LiDAR, and optical imagery. International Journal of Digital Earth, 10(3), 307-323.

[18] Liu, L., Lim, S., Shen, X., & Yebra, M. (2019). A hybrid method for segmenting individual trees from airborne lidar data. Computers and electronics in agriculture, 163, 104871.

[19] Zhong, L., Cheng, L., Xu, H., Wu, Y., Chen, Y., & Li, M. (2016). Segmentation of individual trees from TLS and MLS data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10(2), 774-787.

[20] Ma, Z., Pang, Y., Wang, D., Liang, X., Chen, B., Lu, H., ... & Koch, B. (2020). Individual tree crown segmentation of a larch plantation using airborne laser scanning data based on region growing and canopy morphology features. Remote Sensing, 12(7), 1078.

[21] Premebida, C., Carreira, J., Batista, J., & Nunes, U. (2014, September). Pedestrian detection combining RGB and dense LIDAR data. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 4112-4117). IEEE.

[22] Li, S., Dai, L., Wang, H., Wang, Y., He, Z., & Lin, S. (2017). Estimating leaf area density of individual trees using the point cloud segmentation of terrestrial LiDAR data and a voxel-based model. Remote Sensing, 9(11), 1202.

[23] Nuijten, R. J., Coops, N. C., Goodbody, T. R., & Pelletier, G. (2019). Examining the multi-seasonal consistency of individual tree segmentation on deciduous stands using digital aerial photogrammetry (DAP) and unmanned aerial systems (UAS). Remote Sensing, 11(7), 739.

[24] Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems, 30.

[25] Li, D., Li, J., Xiang, S., & Pan, A. (2022). PSegNet: Simultaneous semantic and instance segmentation for point clouds of plants. Plant Phenomics.

[26] Yang, J., Gan, R., Luo, B., Wang, A., Shi, S., & Du, L. (2024). An improved method for individual tree segmentation in complex urban scenes based on using multispectral LiDAR by deep learning. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 17, 6561-6576.

[27] Zhang, J., Wang, J., Dong, P., Ma, W., Liu, Y., Liu, Q., & Zhang, Z. (2022). Tree stem

extraction from TLS point-cloud data of natural forests based on geometric features and DBSCAN. Geocarto International, 37(25), 10392-10406.