Journal of the Indian Society of Remote Sensing (September 2021) 49(9):2163–2172
https://doi.org/10.1007/s12524-021-01358-x

RESEARCH ARTICLE

An Automatic Method for Tree Species Point Cloud Segmentation Based on Deep Learning

Lin Pu[1,2] · Jiabin Xv[1,2] · Fei Deng[1,2,3]

Received: 12 November 2019 / Accepted: 10 March 2021 / Published online: 7 June 2021

Abstract
Tree species segmentation is an essential condition for research forestry and has a large impact on forest resource monitoring, sustainable forest management, and biodiversity research. Recently, the development of hardware and software has been rapidly increasing. Regarding hardware, the active remote sensing system LiDAR can be used to obtain many point clouds and can significantly improve the tree segmentation accuracy compared with traditional optical remote sensing hardware. With respect to software, deep learning theory is effectively utilized to process 3D point clouds, such as extracting the features of data. However, deep learning-based methods are underutilized in tree species point cloud segmentation. Therefore, it is extremely important to combine current technological advantages for this application. In this article, we construct a point cloud processing dataset that comprises substantial tree information and 5 tree species, including willow, fir, bamboo, palm, and rubber. The novel representation of point clouds via a superpoint graph is utilized to pre-process the point clouds in a large outdoor scene. We propose to apply state-of-the-art deep learning frameworks, including PointNet network and graph convolution networks, to process tree species point clouds in complex forest scenes. We also discuss the effectiveness of the method and the situations influenced by different parameters. The experimental results finally verify the effectiveness of the framework in tree species segmentation.

Keywords Deep learning · 3D point clouds · Point cloud segmentation · Tree species segmentation · Graph convolution network

We confirm that this work is original and has not been published elsewhere nor is it currently under consideration for publication elsewhere.

✉ Fei Deng
fdeng@sgg.whu.edu.cn

Lin Pu
pulin_0@whu.edu.cn

1 School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China

2 Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Land and Resources, Shenzhen 518000, China

3 Collaborative Innovation Center for Territorial Sovereignty and Maritime Rights, Wuhan University, Wuhan 430079, China

Introduction

In deep learning, convolutional neural networks (CNNs) are widely used in 2D image processing, and the applications of CNNs have demonstrated abundant achievements. However, in terms of point cloud processing, the processing efficiency is still not ideal because of massive computation and data storage. To date, 3D data have been preprocessed with deep learning methods for most point cloud processing applications. These methods convert point clouds into a large number of 2D images and further process the images by CNNs (Zhang et al., 2018). For example, the points at a range of a certain point are projected into a window image and further processed by CNNs (Hu & Yuan, 2016). Alexandre Boulch et al. proposed SnapNet, which is also based on CNNs for point cloud processing by converting data to snapshots (Boulch et al., 2017). SnapNet handles snapshot data in two types of images: RGB views and depth composite views with geometric features. Each group of snapshots is pixelated by

CNNs, and the predictions are projected backward to each 3D point by an efficient buffering algorithm.

With the advancement of technology, many new 3D data processing methods have been proposed. Voxelization is one of the mainstream methods for point cloud processing; it decomposes point clouds based on an octree by selecting the smallest cube as the root node and recursively subdividing it into 8 sub-voxels. Non-empty voxel can be further divided until there is no non-empty voxel (Guan et al., 2015). The difficulty of 3D data processing caused by the irregularity of the spatial relationship can be reduced with voxelization so that the data can be further processed by deep learning methods. For example, arithmetic SegCloud uses 3D CNNs for training and detection after voxelization (Tchapmi et al., 2017).

VoxNet integrates a supervised 3D CNN that represents the volume occupancy of the mesh. This method has been applied to classify point clouds quickly and accurately (Maturana & Scherer, 2015). In 2016, scholars from Stanford University proposed the PointNet network (Charles et al., 2017), which is an influential deep learning framework for direct point cloud segmentation and classification. In terms of the disorder of point clouds, PointNet adopts the max-pooling layer to solve the problem. In addition, an important part is that there are two T-net networks in PointNet, which are used for rotating coordinates and extracting intermediate features. T-net networks solve the problem of rotation invariance. PointNet performs well in terms of 3D data recognition tasks with the ModelNet dataset and the ShapeNet part dataset. And its utility was shown on the ModelNet dataset and the ShapeNet part dataset (Li et al. 2016a; Wu et al., 2015). The proposal of PointNet inspired many scholars. Many direct point cloud processing methods have been proposed for 3D object detection, point cloud semantic segmentation, and instance segmentation (Guerrero et al., 2017; Qi et al., 2017b; Wang et al., 2018; Yin & Tuzel, 2017).

Through specific practice, PointNet has achieved splendid outcomes. The characteristics of point clouds are represented by the relationship between the points. However, this network processes a single point with a sliding window and integrates all the sampling points with a max-pooling layer so that local information is lost. Furthermore, scholars have proposed the PointNet^{++} network to address deficiencies in local information (Qi et al., 2017a). However, the application scenes are still difficult to expand to large-scale complex outdoor scenes because of the limitation of the scale of the sliding window. Loic Landrieu et al. proposed a representation named "SuperPoint" to represent the set of points, which enables large-scale point clouds to be processed efficiently (Landrieu & Simonovsky, 2018).

The development of tree species point cloud segmentation research is similar to the development of point cloud processing. For example, some methods process point cloud data based on traditional machine learning methods, such as support vector machines (SVMs) (Xiangyu et al. 2019), and some are based on a deep belief network (DBN) after voxelization (Zou et al., 2017). All of these methods are only used for individual trees and only for tree classification, and few studies take tree point clouds in complex scenes as the segmentation data due to the complex structure between trees.

Summarizing the current relevant research, we find that the application research of deep learning-based point cloud processing in outdoor complex scenes is not comprehensive. The method performs well in large-scale point cloud segmentation processing and achieves a great effect based on SPG, which is verified by the Sementic3D dataset (Hackel et al., 2017). We combined current advanced technical methods and carried out further exploration in complex scene tree species segmentation. This research is of great significance for forestry monitoring, sustainable forest resource management, and biodiversity research.

Method

In this section, we introduce the specific networks and methods of tree species point cloud segmentation utilizing the deep learning theories in this article. First, we preprocess cloud points into superpoints and further obtain the SPG after calculating the geometric partition. In the next step, we embed the superpoints into PointNet to extract the features of point clouds. Finally, we finish contextual point cloud segmentation with graph convolution networks (Simonovsky & Komodakis, 2017; Zheng et al., 2015; Vosselman et al., 2017; Li et al., 2016b). The complete pipeline of tree species point cloud segmentation is shown in Fig. 1.

Geometric Partition

The geometric partition was computed based on 3D geometric features (Demantké et al., 2012), the verticality feature, and the elevation of each point. Point clouds are partitioned automatically according to these five features mentioned above (linearity, planarity, scattering, verticality, and elevation), and each partition is converted to a superpoint.

The features are derived from the local adjacency of individual point of the whole point cloud according to the PCA method. For each single point, we compute the eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$ of the covariance matrix of the adjacent coordinates. The adjacency size is selected that minimizes the entropy E of the vector $(\lambda_1/\Lambda, \lambda_2/\Lambda, \lambda_3/\Lambda)$

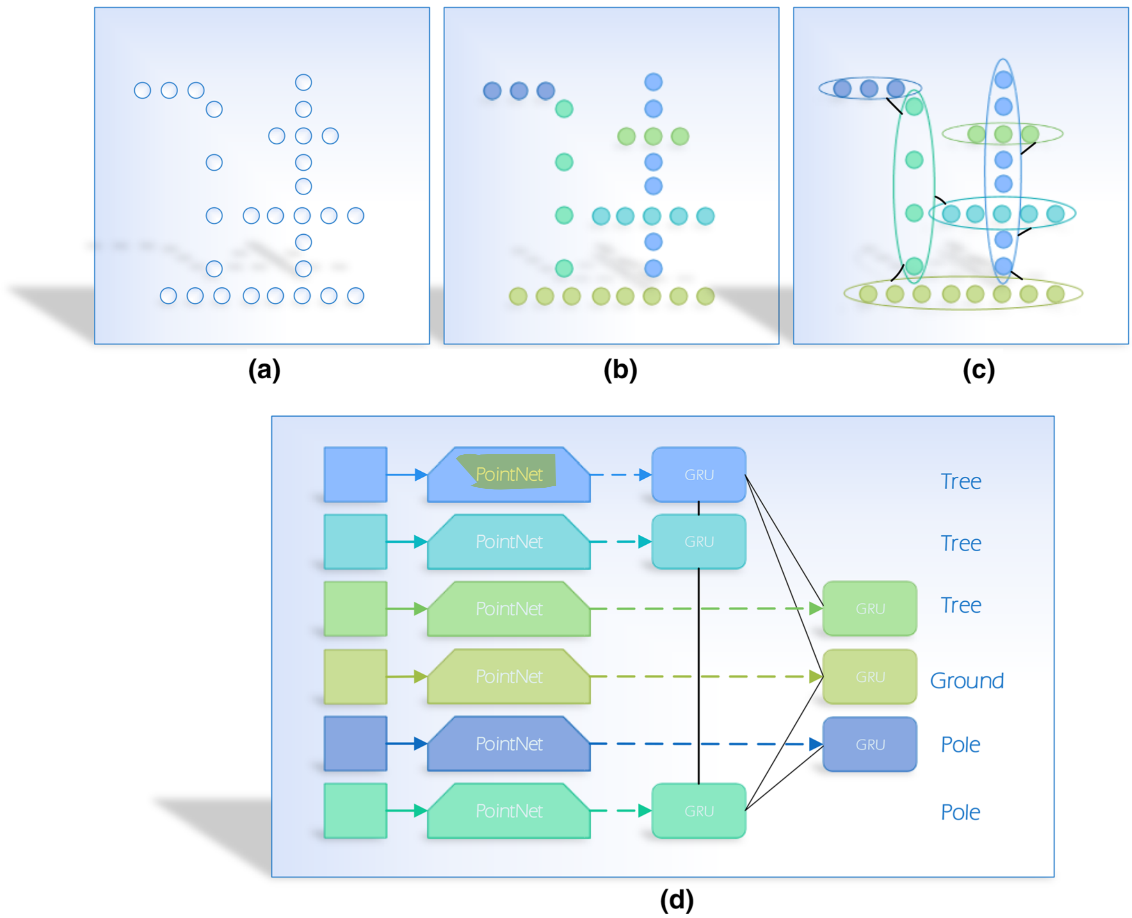

Fig. 1 The pipeline of the SPG framework. **a** Represents the original input data (taking a tree and a pole as examples). **b** Is a hand-crafted point cloud partition, **c** each partition is converted into a superpoint of the SPG, and **d** is the network for superpoint embedding and segmentation based on PointNet and GRU. The black lines represent the superedges

with $\Lambda = \sum_{i=1}^{3} \lambda_i$, in accordance with the optimal adjacency:

$$E = -\sum_{i=1}^{3} \frac{\lambda_i}{\Lambda} \log\left(\frac{\lambda_i}{\Lambda}\right) \tag{1}$$

The linearity, planarity, and scattering are derived by the following terms:

$$\begin{aligned} \text{linearity} &= \frac{\lambda_1 - \lambda_2}{\lambda_1} \\ \text{planrity} &= \frac{\lambda_2 - \lambda_3}{\lambda_1} \\ \text{scattering} &= \frac{\lambda_3}{\lambda_1} \end{aligned} \tag{2}$$

The linearity describes how elongated the adjacency is, the planarity assesses how plane the adjacency is, the high-scattering values are used to represent an isotropic and spherical adjacency. We describe these three point features as geometric features.

The computation of the geometric partition in the SPG framework is an important basis for constructing an SPG. The target of processing is not to index or identify the object point clouds but to classify every point in point clouds into different categories. To this end, we assume that every point in the obtained 3D point clouds belongs to only one category.

In this article, the generalized minimal partition problem is studied by referring to the partition problem of global properties (Guinard & Landrieu, 2017). The basic theoretical formula used is as follows:

$$\underset{g \in R^{d_g}}{\arg\min} \sum_{i \in C} \|g_i - f_i\|^2 + \mu \sum_{(i,j) \in E_{nn}} \omega_{i,j} [g_i - g_j \neq 0] \tag{3}$$

The $[\cdot]$ in formula (3) is an Iverson bracket. For each point $i \in C$, the shape of its local neighborhood is characterized with geometric features $f_i \in d_g$. μ is the regularization factor and influences the coarseness of the partition. $\omega_{i,j}$ is the edge weight, and E_{nn} represents the point cloud adjacency relationship. For the partition, the l_0-cut algorithm is adopted in this paper. This algorithm does not need to select the partition range in advance and can rapidly carry out regional processing (Landrieu & Obozinski, 2017).

According to the pipeline of our framework, first, we compute the geometric partition. Then, each geometric partition is converted into a superpoint. Next, we input the superpoints in the SPG to a PointNet network for superpoint embedding. The final operation is to segment the point clouds based on context with a graph convolution network. Therefore, the results of geometric partition have a great impact on the semantic segmentation due to different superpoint caused by varying partition result.

We will discuss the regularization parameter μ which influences the coarseness of the partition. In formula (3), the former term describes the fidelity, and the latter is the smoothness. The varying regularization parameter μ causes different partition results. Then, when we convert the partition to a superpoint, it will construct varied superpoint graph, which is actually the neural network input.

The Structure of the SPG

The SPG is a new structured representation of point clouds and is defined as an oriented attributed graph G:

$$G = (S, E, F) \tag{4}$$

In formula (4), the nodes of G are the sets of superpoints S, and superedges E represent the adjacency between the superpoints. The superedges are annotated by a set of d_f features, namely $F \in \mathbb{R}^{E \times d_f}$, characterizing the adjacency relationship between superpoints. The SPG has the following characteristics:

1. SPG considers all object parts as a whole rather than classifying a single point or voxel.
2. SPG can describe a more rational relationship between adjacent objects by considering the contextual information for segmentation. For example, cars and trees are higher than the ground in the outdoor scenes. In indoor scenes, the walls are around, and the ceilings are above.
3. The data processed by an SPG can describe the simple structures of some point clouds rather than the isolated points. Therefore, this representation can perform better to show the integrity of the point clouds, and it is several orders of magnitude smaller than the raw point clouds.

We can encode every part of the object by this novel point cloud representation, which can perform better in semantic segmentation by considering the contextual relationship so that we can segment point clouds in an outdoor scene rather than only a small scene.

Superpoint Embedding and Semantic Segmentation Based on Deep Neural Networks

The fundamental aspect of superpoint embedding is the PointNet network, which can turn each superpoint into a fixed dimension vector. It is convenient for contextual semantic segmentation. The structure of the PointNet network for superpoint embedding is shown in Fig. 2.

We segment point clouds based on contextual information by classifying the superpoint's embedding vector and its surroundings in the GCN. The approach builds with gated graph neural networks and edge-conditioned convolutions (ECCs). A GRU is a kind of RNN and is used as the input gate because it can learn to ignore contextual information and pay more attention to special features if the class state is highly certain. An ECC is an operation on graph-structured data that can process the value of continuous attributes with the MLP and dynamically generate filter weights conditioned on the edge labels. The MLP consists of 4 layers (widths 32, 128, 64, 32) with ReLUs. We used Adam in the training, and batch normalization is used in the networks as well as ReLUs (Ioffe & Szegedy, 2015).

Experiments

Dataset

It can be shown from the recent improvements in deep learning that enough datasets for training classifiers are highly required when processing the data in complex real-world scenes. Until now, datasets with rich objects and a large number of marked labels for point cloud segmentation in a complex scene have been absent. For example, the famous Oakland dataset contains fewer than 2 million markers, while the popular benchmark created from New York University only provides point clouds in indoor scenes (Munoz et al., 2009). Both the Sydney city object dataset and the IQmulus & TerraMobilita competition dataset were scanned by a 3D Velodyne LiDAR that was mounted on a car, which measures much fewer points than a static scanner, as well as the benchmarks scanned by airborne sensors (Vallet et al., 2015). The Semantic3D benchmark provides more than 4 billion points in total, which include 8 categories: artificial terrain, natural terrain, high vegetation, low vegetation, architecture, artificial landscape, scanning artificial sculpture, automobile, and additional unmarked points (Hackel et al., 2017).

In summary, there is no specific tree point cloud benchmark. We constructed a new augmented tree species point cloud dataset for segmentation. The origin dataset

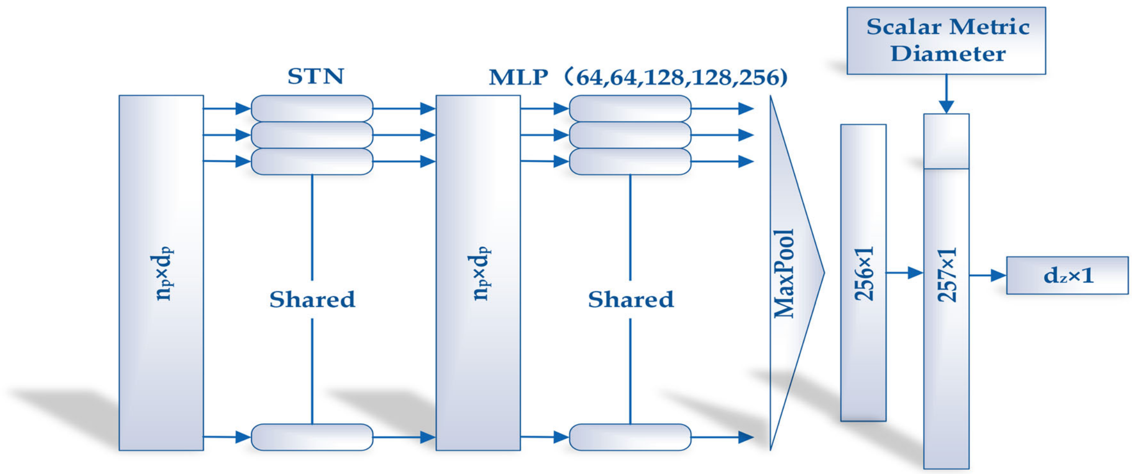

Fig. 2 The structure of the PointNet embedding network, which embeds n_p d_p-dimensional samples of a superpoint into a d_z-dimensional vector. We set n_p = 128, $n_{\mathrm{min}p}$ = 40 and d_z = 32. STN is a network with 3 multilayer perceptrons (MLPs) (64, 64, 128) before max pooling and 3 MLPs (128, 64, 4) after

contains 5 tree species, including willow, fir, bamboo, palm, and rubber, with a total of 4,934,766 points, which were scanned by LiDAR provided by the College of Information Science and Technology, Nanjing Forestry University. Their tree species classification research also inspired us. The tree species classification method in this research was based on the SVM method (Xiangyu et al., 2019). All of the point cloud coordinate information is used in the experiments, including the single category point clouds and the mixed point clouds in a real scene. We labeled each point manually in the whole dataset with different numbers to represent different tree species. Some of the original point clouds for the experiments are shown in Fig. 3.

Point Cloud Segmentation

First, we compute the geometric partition. Then, each geometric partition is converted into a superpoint. Next, we input the superpoints in the SPG to a PointNet network for superpoint embedding. The final operation is to segment the point clouds based on context with a graph convolution network.

We use PyTorch as the deep learning framework and Ubuntu 16.0 as the experimental platform, and we finished the experiments with 32 GB of memory and an NVIDIA RTX 2070 GPU.

The main experimental pipeline of the experiments is as follows:

1. Data pre-processing: We only use point coordinate information in this algorithm, which corresponds to the label of each point. We pre-process input data with voxelization subsampling by computing per-voxel mean positions and observations over a regular 3D grid, which can help increase the computational efficiency. After that, we convert the information and labels into the hdf5 format.
2. Model training: We process data with a deep neural network and adjust the training epoch, batch size, and volume size to output a classifier for prediction.
3. Result output: The point cloud test set is segmented, and we can obtain the final segmentation results. The output files include point cloud geometric partition graphs, SPGs, and segmentation results.

Results and Discussion

The output results of this experiment are shown in Figs. 4 and 5. Individual tree segmentation results are shown in Fig. 4. Different tree species are represented by different colors. The species in order from left to right are 4(a) willow, 4(b) fir, 4(c) bamboo, 4(d) palm, and 4(e) rubber. The tree segmentation results in a large-scale scene are shown in Fig. 5. Figure 5a shows geometric features. In the geometric feature graphs, red represents the linear features, green represents the horizontal features, and blue represents the vertical features. 5(b) is geometric partition results. The geometric partitions are represented with random colors. Each region is turned into a superpoint. 5(c) is an SPG. In SPGs, the superpoints and edges can be shown. Finally, 5(d) shows the segmentation results. Different colors represent different kinds of trees in the segmentation graphs.

In the experiments, we use the unweighted average of the intersection over union (mIoU) of each class and the

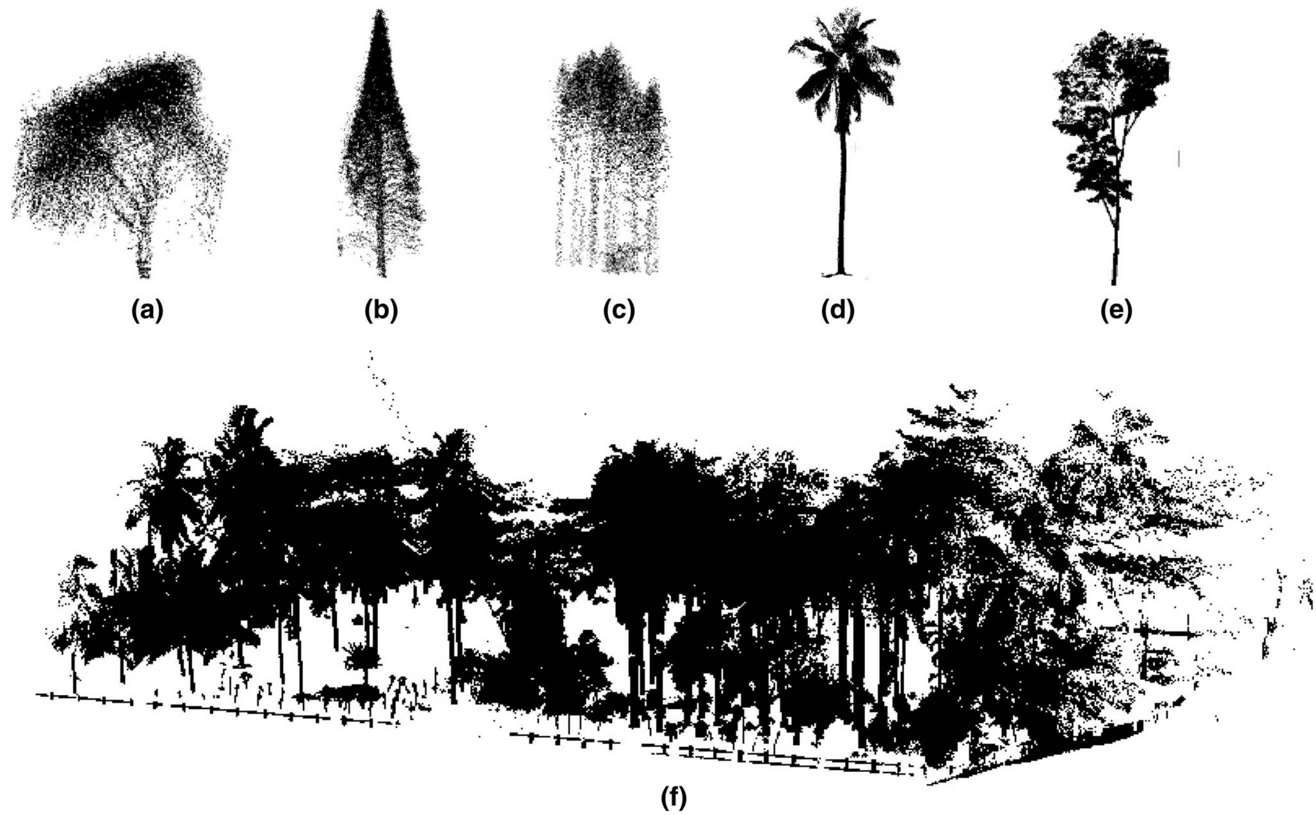

Fig. 3 Part of the dataset. The tree species include **a** willow, **b** fir, **c** bamboo, **d** palm, **e** rubber, **f** shows a part of the grove point cloud data

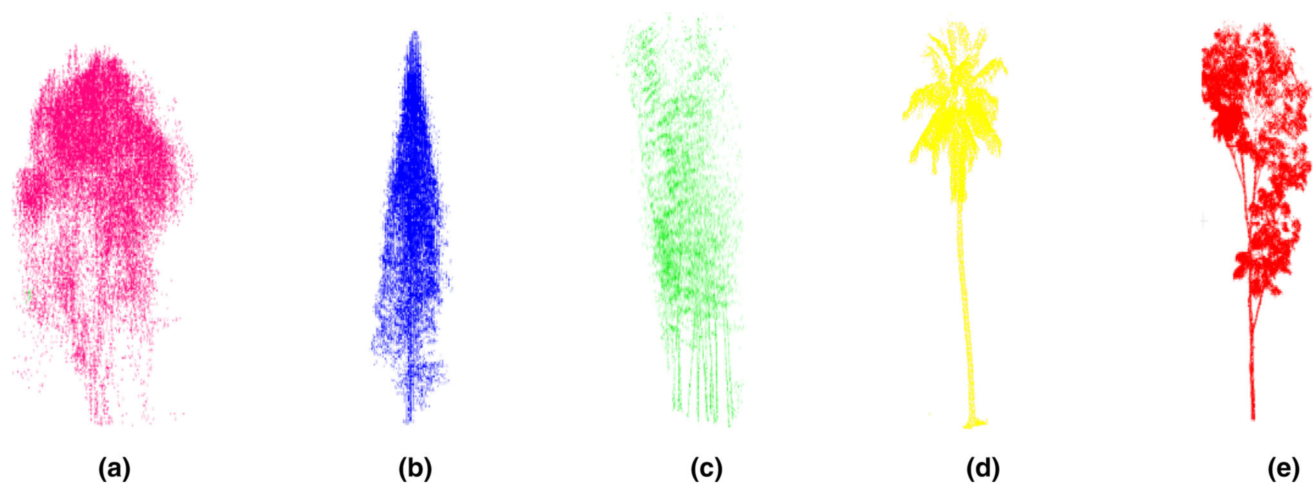

Fig. 4 Individual tree segmentation. The order corresponds to the sample data in Fig. 3. Different tree species are represented by different colors. **a** Willow, **b** fir, **c** bamboo, **d** palm, **e** rubber

overall accuracy (OA) as metrics. We adjusted three parameters (voxel size is the voxelization subsampling size, the regularization strength influences the coarseness of the minimal partition, and batch size means the batch number of point cloud samples in training). We used the hold-out method to distinguish the test set and the training set at a ratio of 3:7 for preprocessing the original data. We further adjust the parameters of networks to explore the influence of parameters on experimental results, and we discuss the selection of the frame parameters.

Selection of the Frame Parameters

We investigated the effects of several parameters in our framework, such as the regularization strength, voxel size for the minimal partition, and training batch size. In

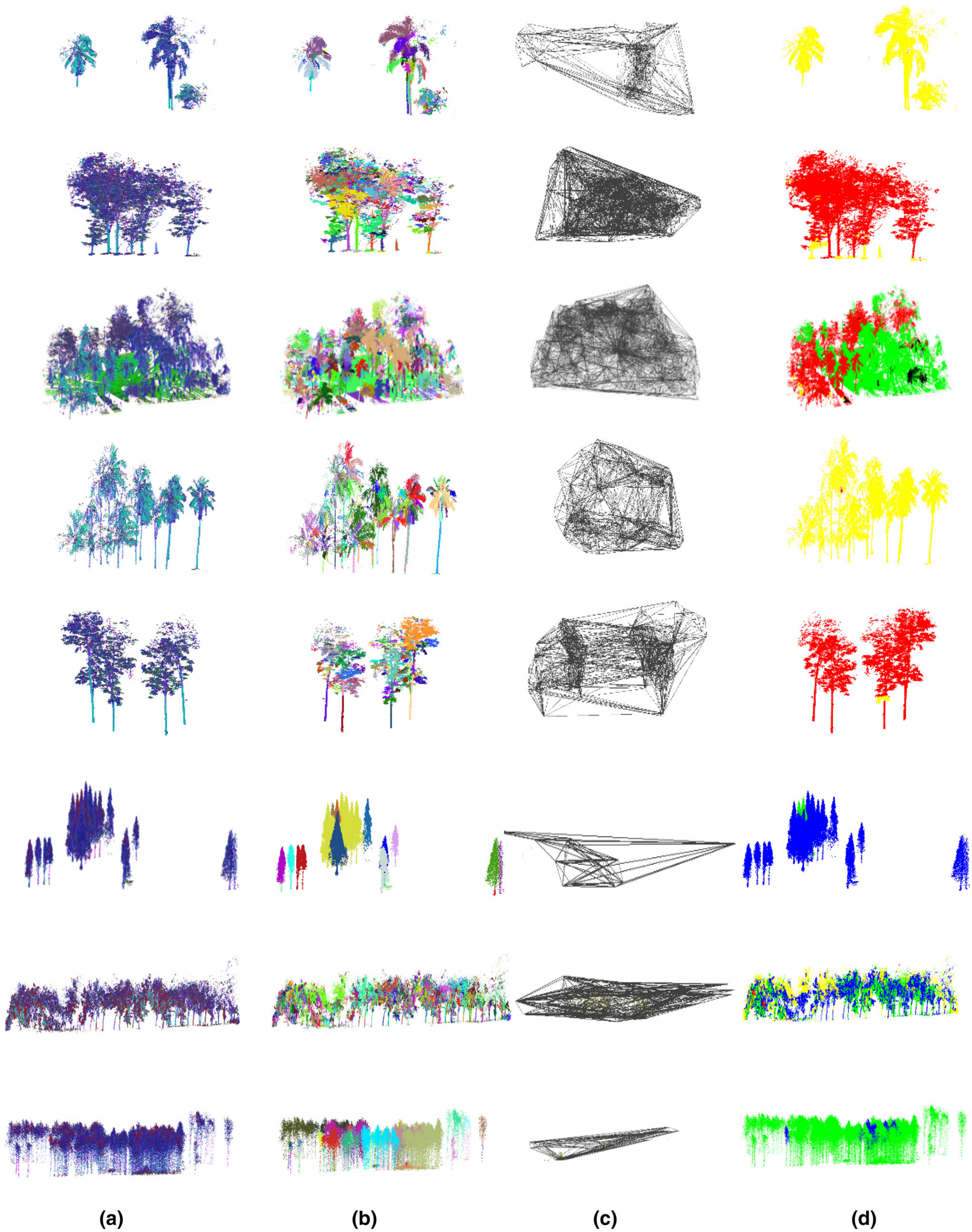

Fig. 5 Part of segmentation experimental results. **a** Geometric features, **b** Geometric partition, **c** SPG, and **d** Segmentation result

training, we set the learning decay rate to 0.7 so that the learning rate gradually decays as the training progresses. Table 1 shows the experimental results in which we fixed the voxel size and batch size and adjusted the regularization strength, from which we can be told that result is better while regularization strength is 0.2.

Relatively speaking, the voxel size parameter is more important; it influences the point cloud voxelization randomly when computing the geometric partition. It helps increasing the computing efficiency and decreasing the memory requirement during learning and inference. Generally, the larger the voxel size is, the greater the loss of point cloud data is, the less computing time and memory needed. Besides, it influences the accuracy of segmentation by acting as a form of geometric and radiometric denoising as well. In the experiments, we can obtain relatively good results when we take a voxel size = 0.04. We think that this value improves the segmentation efficiency while ensuring the preservation of tree point cloud details in the tree species segmentation task. Table 2 shows the experimental results when we adjusted the voxel size number.

For tree species segmentation, appropriately increasing batch size can improve efficiency while improving segmentation accuracy. We believe that within a reasonable range limited by GPU memory, a larger training batch size of 8 can better preserve the original features of the point clouds than others. In addition, the network more easily obtains a sharp minimum rather than a flat minimum when the batch size is larger. Therefore, we can obtain better results when the batch size is 8 in our experiments (Table 3).

Through the experimental results, we find that the adjustment of the parameters can also improve the mis-segmentation situation. In summary, we obtain the best result in the five tree species experiments: The mIoU is 81.49% and the OA is 92.45%, while the voxel size is 0.04, the training batch size is 8 and the regularization strength is 0.2.

Effectiveness of the Framework for Target Segmentation

Since the model used in this paper takes point cloud data as the original input, it can extract as many data features as possible. The T-net network proposed in PointNet can better solve the problem of rotation invariance and disorder of the point cloud data. Constructing an SPG for point cloud data processing provides a novel and feasible way for us to directly process and utilize point cloud data in large-scale scenes, enabling us to effectively segment and process them. For point cloud partitioning, some scholars have proposed a point cloud automatic partitioning method based on deep metric learning, which can achieve better results in the semantic segmentation of point clouds (Landrieu & Boussaha, 2019). By comparing the experimental results, we find that when the data of the training set and the test set belong to the same environment, the processing result is better due to the certain degree of overlap and similarity between the data. However, for non-identical environments, the processing results are relatively inferior. Moreover, the model has better prediction accuracy for single-species tree point clouds, and the prediction performance of mixed point clouds of multiple trees in large-scale scenes is not accurate enough. Experiments show that this method can accurately segment the point clouds of a single tree species and can effectively identify and retrieve trees whose features have been learned in a large-scale scene. Therefore, the method can be effectively applied to the retrieval and identification of a single tree species in a large-scale scene. For the fine segmentation of each tree species in a large-scale scene, further research is needed on the dataset, frame structure, and parameter selection.

Table 1 The test result when the voxel size = 0.02 and the batch size = 1

Regularization strength	OA (%)	mIoU (%)	Willow (%)	Fir (%)	Bamboo (%)	Palm (%)	Rubber (%)
0.2	70.76	48.08	20.86	30.01	80.05	51.35	58.13
0.4	70.00	45.54	12.22	33.36	70.49	52.77	58.87
0.6	70.11	47.34	40.93	15.47	74.43	51.27	54.57

Table 2 The test result when the regularization strength = 0.2 and the batch size = 1

Voxel size	OA (%)	mIoU (%)	Willow (%)	Fir (%)	Bamboo (%)	Palm (%)	Rubber (%)
0.02	70.76	48.08	20.86	30.01	80.05	51.35	58.13
0.03	67.16	49.03	48.61	36.00	69.86	44.40	49.29
0.04	66.39	49.98	59.05	36.85	65.24	47.97	40.80

Table 3 The test result when the voxel size = 0.04 and the regularization strength = 0.2

Batch size	OA (%)	mIoU (%)	Willow (%)	Fir (%)	Bamboo (%)	Palm (%)	Rubber (%)
1	66.39	49.98	59.05	36.85	65.24	47.97	40.80
2	81.02	68.24	83.32	51.20	76.33	61.39	68.98
4	87.70	72.06	74.46	44.36	77.04	78.50	85.95
8	92.45	81.49	71.79	71.38	87.70	85.83	90.74

In addition, this experiment only uses the coordinate information of the point clouds. We believe that in combination with the radiant intensity, colour, pre-processing information of the point clouds, point cloud data of more different tree species, and the true labels of points in large-scale scenes, the method can effectively identify the tree species or regional trees that have been learned.

Conclusion

This paper takes tree species point clouds as the research object, uses an independently constructed point cloud dataset, considers the actual comprehensive utility, and uses a deep neural network-based method to segment tree species point clouds in real outdoor scenes. The experiments verify the effectiveness of this method for the automatic segmentation of tree species point clouds. By considering the different combinations of parameters, we obtain mIoU of 81.49% and an OA of 92.45% for point cloud segmentation of five tree species with a regularization strength of 0.2, a training batch size of 8, and a voxel size of 0.04. Through experiments, we find that the choice of the parameter combination is more important than an individual parameter. We think that this method can be applied to the extraction and recognition of tree species in outdoor complex scenes and can be extended to the segmentation of other ground objects. This method provides a valuable reference for forestry research, outdoor real-world 3D reconstruction and many kinds of ground object research by training with massive real-world tree species point cloud data. Due to the influence of seasonality, the surrounding environment, different distributions of data, overlapping objects, and complex structures between trees, it is still difficult to accurately segment tree species point clouds in large-scale complex scenes. Therefore, in the future, we will pay more attention to constructing a dataset that has different and more balanced distribution tree species data. We will try to combine the advanced technique and the dataset to segment more different outdoor tree species point cloud.

Acknowledgements We would like to thank the College of Information Science and Technology, Nanjing Forestry University for providing the data.

Author Contributions FD contributed to conceptualization, funding acquisition, resources, project administration, and supervision; LP were involved in formal analysis, investigation, validation, visualization, writing—original draft; LP, JX contributed to methodology; LP, JX, FD were involved in writing—review and editing.

Funding This research was funded by the Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Land and Resources (No. KF-2018-03-025).

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Boulch, A., Guerry, J., Saux, B. L., & Audebert, N. (2017). SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers & Graphics, 71,* S0097849317301942.

Charles, R. Q., Hao, S., Mo, K., & Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE conference on computer vision and pattern recognition (CVPR), 2017* (pp. 77–85)

Demantké, J., Mallet, C., David, N., & Vallet, B. (2012). Dimensionality based scale selection in 3D lidar point clouds. *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 3812*(5), 97–102.

Guan, H., Yu, Y., Zheng, J., Li, J., & Zhang, Q. (2015). Deep learning-based tree classification using mobile LiDAR data. *Remote Sensing Letters, 6*(11), 864–873.

Guerrero, P., Kleiman, Y., Ovsjanikov, M., & Mitra, N. J. (2017). PCPNET: Learning local shape properties from raw point clouds. *Computer Graphics Forum, 37*(2), 75–85.

Guinard, S., & Landrieu, L. (2017). Weakly supervised segmentation-aided classification of urban scenes from 3D lidar point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-1/W1,* 151–157. https://doi.org/10.5194/isprs-archives-XLII-1-W1-151-2017

Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., & Pollefeys, M. (2017). Semantic3d.net: A new large-scale point cloud classification benchmark . *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1/W1,* 91–98. https://doi.org/10.5194/isprs-annals-IV-1-W1-91-2017

Hu, X., & Yuan, Y. (2016). Deep-learning-based classification for DTM extraction from ALS point cloud. *Remote Sensing, 8*(9), 730.

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. Paper presented in International Conference on Machine Learning, Lille, pp. 448–456.

Landrieu, L., & Boussaha, M. (2019). *Point cloud oversegmentation with graph-structured deep metric learning*. Paper presented at the 2019 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Landrieu, L., & Obozinski, G. (2017). Cut pursuit: Fast algorithms to learn piecewise constant functions. *Siam Journal on Imaging Sciences*, *10*(4), 1724–1766.

Landrieu, L., & Simonovsky, M. (2018). *Large-scale point cloud semantic segmentation with superpoint graphs.* Paper presented at the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Li, Y., Guibas, L., Kim, V. G., Ceylan, D., Shen, I. C., Yan, M., et al. (2016a). A scalable active framework for region annotation in 3D shape collections. *Acm Transactions on Graphics, 35*(6), 1–12.

Li, Y., Tarlow, D., Brockschmidt, M., & Zemel, R. (2016b). *Gated graph sequence neural networks*. Paper presented in 4th International conference on learning representations (ICLR).

Maturana, D., & Scherer, S. (2015). VoxNet: A 3D convolutional neural network for real-time object recognition. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015,* 922–928.

Munoz, D., Bagnell, J. A., Vandapel, N., & Hebert, M. (2009). Contextual classification with functional Max-Margin Markov Networks. In *2009 IEEE conference on computer vision and pattern recognition, 20–25 June 2009 2009* (pp. 975–982). doi:https://doi.org/10.1109/CVPR.2009.5206590.

Qi, C. R., Li, Y., Hao, S., & Guibas, L. J. (2017a). *PointNet++: Deep hierarchical feature learning on point sets in a metric space.* Paper presented at the 2017 Conference on Neural Information Processing Systems(NIPS).

Qi, C. R., Wei, L., Wu, C., Hao, S., & Guibas, L. J. (2017b). *Frustum PointNets for 3D object detection from RGB-D data.* Paper presented at the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Simonovsky, M., & Komodakis, N. (2017). Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *2017 IEEE conference on computer vision and pattern recognition (CVPR), 21–26 July 2017 2017* (pp. 29–38). doi:https://doi.org/10.1109/CVPR.2017.11.

Tchapmi, L. P., Choy, C. B., Armeni, I., Gwak, J., & Savarese, S. (2017). SEGCloud: Semantic segmentation of 3D point clouds. *arXiv e-prints.*

Vallet, B., Brédif, M., Serna, A., Marcotegui, B., & Paparoditis, N. (2015). TerraMobilita/iQmulus urban point cloud analysis benchmark. *Computers & Graphics, 49*(C), 126–133.

Vosselman, G., Coenen, M., & Rottensteiner, F. (2017). Contextual segment-based classification of airborne laser scanner data. *Isprs Journal of Photogrammetry & Remote Sensing, 128,* 354–371.

Wang, W., Yu, R., Huang, Q., & Neumann, U. (2018). *SGPN: Similarity group proposal network for 3D point cloud instance segmentation.* Paper presented at the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., et al. (2015). *3D ShapeNets: A deep representation for volumetric shapes.* Paper presented at the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Xiangyu, C., Ting, Y., Lianfeng, X., & Yingan, L. (2019). Classification of tree species using LiDAR point cloud data. *Laser & Optoelectronics Progress, 56*(12), 203–214.

Yin, Z., & Tuzel, O. (2017). *VoxelNet: End-to-end learning for point cloud based 3D object detection.* Paper presented at the 2018 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR),

Zhang, L., Li, Z., Li, A., & Liu, F. (2018). Large-scale urban point cloud labeling and reconstruction. *ISPRS Journal of Photogrammetry & Remote Sensing, 138,* 86–100.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., et al. (2015). Conditional random fields as recurrent neural networks. In *2015 IEEE international conference on computer vision (ICCV), 7–13 Dec. 2015 2015* (pp. 1529–1537). doi:https://doi.org/10.1109/ICCV.2015.179.

Zou, X., Ming, C., Cheng, W., Yan, X., & Li, J. (2017). Tree classification in complex forest point clouds based on deep learning. *IEEE Geoscience & Remote Sensing Letters, 14*(12), 2360–2364.