ISSN 1060-992X, Optical Memory and Neural Networks, 2023, Vol. 32, Suppl. 2, pp. S270–S276. © Allerton Press, Inc., 2023.

Individual Tree Segmentation Quality Evaluation Using Deep Learning Models LiDAR Based

I. A. Grishin[a], *, T. Y. Krutov[a], **, A. I. Kanev[a], *, and V. I. Terekhov[a], ******

[a] Bauman Moscow State Technical University, Moscow, 105005 Russia

**e-mail: ilia-grishin@mail.ru*

***e-mail: krutov@bmstu.ru*

****e-mail: aikanev@bmstu.ru*

*****e-mail: terekchow@bmstu.ru*

Received August 15, 2023; revised September 1, 2023; accepted September 5, 2023

Abstract—The study of the forest structure makes it possible to solve many important problems of forest inventory. LiDAR scanning is one of the most widely used methods for obtaining information about a forest area today. To calculate the structural parameters of plantations, a reliable segmentation of the initial data is required, the quality of segmentation can be difficult to assess in conditions of large volumes of forest areas. For this purpose, in this work, a system of correctness and quality of segmentation was developed using deep learning models. Segmentation was carried out on a forest area with a high planting density, using a phased segmentation of layers using the DBSCAN method with preliminary detection of planting coordinates and partitioning the plot using a Voronoi diagram. The correctness model was trained and tested on the extracted data of individual trees on the PointNet ++ and CurveNet neural networks, and good model accuracies were obtained in 89 and 88%, respectively, and are proposed to use the quality assessment of clustering methods, as well as improve the quality of LiDAR data segmentation on separate point clouds of forest plantations by detecting frequently occurring segmentation defects.

Keywords: LiDAR, deep learning, forest inventory, segmentation, machine learning, artificial intelligence

DOI: 10.3103/S1060992X23060061

1. INTRODUCTION

1.1. Problem and Application

The study of the forest structure allows solving many important problems, among them: material assessment of the amount of different types of wood, assessment of the ecological and climatic balance, determination of the biological diversity of forest inhabitants, food supplies and water quality. In addition, forests and plants are sources of medicines. In Russia, about 230 species of medicinal plants are used for medical purposes. More than half of them are used in the pharmaceutical industry, and about 90 more species are supplied to pharmacies after preliminary drying and grinding [1].

Remote sensing of a forest using 3D LiDAR laser scanning systems is today one of the most widely used methods for obtaining information about a forest area. Due to the design features and principle of operation, LiDAR allows you to get detailed images with high spatial resolution. With the help of the obtained images, it is possible to establish the physical parameters of an individual tree, including: tree height, trunk diameter at breast height, crown volume, as well as structural and quantitative parameters of a tree plot, coordinates and number of plantings. In the future, it is possible to calculate the biological parameters of the tree: the volume of phytomass, the age and health of the tree, as well as to find out the species of the plantation. However, LiDAR data processing is a non-trivial task; in manual mode, it is necessary to select a separate tree from the entire volume of captured data, the operation must be carried out for each tree. For small forest polygons, manual segmentation is acceptable, but with an increase in the area under study, this method of segmentation becomes labor-intensive and not justified.

Carrying out automated segmentation of forest areas and assessing its quality is one of the stages of the transition to an automated system for assessing taxation parameters of the forest.

The main task of the study is to form an assessment of the quality of the use of artificial neural networks for the verification of manual or automated segmentation of unordered sets of three-dimensional tree points.

1.2. Current Research

Today, many research groups are engaged in the analysis of the forest area. Also, analysis and monitoring of agricultural plantations is carried out using remote sensing [2, 3]. It is noted in [4] that the number of publications in the problems of tree segmentation and classification using LiDAR data has increased significantly and amounts to 80% of the total number of publications.

Many works are aimed at studying ways to classify trees, so in [5] an approach is considered using three deep learning models based on multilayer perceptron (MLP)—PointNet, PointNet++, PointMLP for classifying the species of individual trees. Various data collected by unmanned aerial vehicles (UAVs) are often used. Thus, the classification of rocks, identification and delineation of crowns using hyperspectral data are proposed in [6]. Dense point cloud data often act as additional data to hyperspectral ones [7]. The study [8] offers a comparative analysis of photogrammetric data of different quality and laser scanner data obtained from UAVs. The preferred data quality was obtained using laser scanning, although the study noted that high quality photogrammetric data from a high-quality camera is sufficient to detect individual trees and determine their height. Work [9] uses airborne laser scanning data as the basis for extracting information about tree species. A study [10] compares the quality of localization of trees with different leaf cover, and it was found that UAV photography makes it difficult to detect tree trunks in conditions of full leaf cover.

Training classification models requires a labeled dataset of individual trees. In the case of images, the pixels of the received data are used, for example, for controlled clustering [11]. In the three-dimensional case, point cloud processing is used, for example, work [12] proposes automating the task of tree segmentation for LiDAR data. The paper [13] presents an algorithm for mapping street trees. The study [14] uses virtual reality technology to simplify the process of labeling scans, which are used to train neural networks involved in solving problems of semantic segmentation, lidar odometry and mapping.

Obtaining taxation information of plantings is also a required task when carrying out forest inventory work, so the study [15] proposes a solution to this problem using LiDAR data of individual trees. To calculate the structural parameters of plantings, reliable segmentation of the initial data is required, the quality of segmentation can be difficult to assess in conditions of large volumes of forest plots. For this purpose, in this work, a system of correctness and quality of segmentation was developed and used, which was carried out on a forest plot with a high planting density using the methods proposed below. The correctness model is trained and validated on the extracted data of individual trees, and is proposed to use quality assessment, as well as it will allow to improve the quality of LiDAR data segmentation into individual forest stand point clouds.

2. THE STRUCTURE OF THE INPUT DATA AND NETWORKS

In general, a 3D scene using lidar data can be represented as a matrix of vectors, each vector represents an Eq. (1).

$$p_i = \left\{x_i, y_i, z_i, \theta_i^1, \theta_i^2, \ldots, \theta_i^M\right\}, \quad i = 1 \ldots N, \tag{1}$$

where the first three parameters are the coordinates of the point in Euclidean space, the rest are additional parameters (color channels, intensity, etc.

Each object (a set of points of one tree) is represented as an Eq. (2).

$$P = \left\{p_1, p_2, \ldots, p_N\right\}^T. \tag{2}$$

As a tool for identifying trees, neural network models were chosen that allow working directly with dense point clouds: PointNet++ and CurveNet.

The PointNet architecture was designed with an important feature of the input data in mind: the set of points that form the cloud is unordered. Hence, the need arose to use an invariant approach when reading the input data. PointNet uses symmetric functions for this, which allow you to get the same result regardless of the order in which the data is read.

PointNet consists of two subnets: classifying and segmenting. The PointNet classification network takes points as input, processes each point independently and obtains its local representation, and then combines these representations to obtain a global representation of the entire set of points. The segmenting network is used as a superstructure over the classifying part and contains global and local features. PointNet learns to generalize the shape of objects using a sparse set of keypoints, which generally form the object's skeleton. The authors of [16] note the good ability of the network to process point clouds with a high degree of sparseness. So reducing the number of points by 2 times relative to the original number leads to a slight decrease in accuracy to 3%.

The architecture of the PointNet++ neural network is based on the use of the PointNet network, and is its modernization. The classical PointNet is limited in its ability to recognize detailed objects and has a low degree of generalization of complex multi-element 3d scenes.

PointNet++ uses a distance metric to divide points into local areas that overlap, then extract local features from small areas, capturing subtle geometric structures. For more efficient data processing in dense point clouds, special layers were added that can aggregate information from different scales, taking into account the non-uniform density of points in the data. Working with data at different levels of abstraction contributes to improving the accuracy of the model compared to PointNet.

One of the tasks of CurveNet [17] is to preserve not only the local regularities of cloud points, but those that are remote from each other. For this purpose continuous sequences of points are used. Such sequences of points are represented as curves. The learning phase involves grouping the curves by defining the starting points, the choice of the starting point is especially important. Using the method of choosing the best k from the U-Net, an estimate is made for each considered point, the points with the best indicators are set as the initial points. CurveNet studies the local structure of a point cloud using the LPFA (Local Point-Feature Aggregation) block. The CurveNet forecast is determined through TSH (Task-Specified Head), which allows it to be used for various research areas.

3. EXPERIMENT

3.1. Retrieving Individual Tree Data

We used ground scanning data from a portable LiDAR scanner of a forest area containing 624 trees, the number of which was established in the input of field work, of the following species composition: spruce, pine, birch, mountain ash, oak, aspen, which are indicated in descending order of prevalence on the site.

To determine the coordinates of isolated trees, a study [18] is used, which offers a reliable procedure for detecting and localizing stands. The method is to work with the lower layer of the point cloud with a height of no more than two meters. The selected layer is divided into parts along the lines of movement of the LiDAR operator performing ground scanning, then the parts are further segmented into separate visible parts of the trees in several stages using filtering and segmentation methods.

The obtained coordinates of the location of trees on the site are used by us to automatically divide the site into sub-areas, each of which contains only one tree, using the Voronoi diagram. This method has proved its effectiveness for determining the main taxation parameters of plantations [19]. However, it is also necessary to clearly view the entire height of the tree, so it is proposed to view many points in each Voronoi cell over the entire height with some step. On each selected layer, segmentation of points is performed using the DBSCAN method [20] and a cluster is selected, the center of which is at the closest distance to the previous cluster center or the initial coordinate in the case of the first layer.

During ground scanning, the density of points on the tops of trees decreases. For these reasons, the cloud is pre-processed with octree partitioning. Each node in the octant tree divides the space into eight new octants. The splitting continues until the specified depth of the tree is reached (in this case, it is chosen equal to 10) or until all cells contain the same number of points or objects. Each octree cell stores the closest point to the center of the octree cell. Thus, the denser lower part of the cloud undergoes thinning while maintaining the density in the upper part. To equalize the density of the cloud in height, points are also added to its upper part inside each cell of the octree, which are the center of gravity of the octant, artificially adding new points that are not necessarily in the original data. This method makes it possible to obtain an almost uniform distribution of points in case of their lack in the upper part.

Since the tree can be tilted, the cell is gradually shifted by the detected offset, allowing the tilt of the tree to be tracked.

The cluster center is chosen using a two-dimensional kernel density estimate (KDE) [21], where the center of the densest projection of points onto a horizontal plane is chosen as the main one. Since KDE can return multiple centers at once, the one closest to the median center is chosen. Not all points are used as data, but points with the highest intensity values.

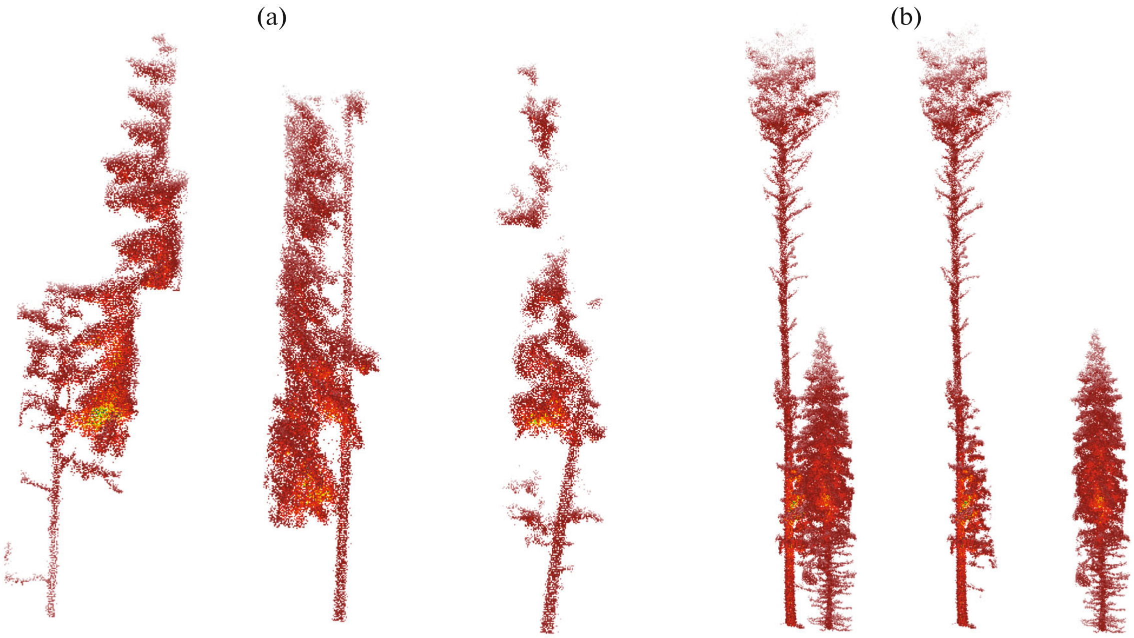

Fig. 1. ErrTrunk and 2Crowns segmentation defects.

3.2. Segmentation Defects

The method used to determine the center and gradually offset sometimes loses the tree trunk when viewing the layers of the point cloud. The main part of the errors is the exclusion of point clouds, when the dense crown points were chosen as trunk points, since they had a high density and intensity values, which is not typical for most cases. This case is designated by us as the defect "incorrect determination of the tree trunk" (ErrTrunk) and is presented in Fig. 1a in three different instances.

The next common defect is the case when two trees of different breeds and heights grow close to each other, so that the crown of a low tree is in contact with the trunk of a higher one. The selection is erroneous for a taller tree, creating the case of "two crowns at different heights" (2Crowns), which is presented in Figure 1b, where the first is two trees, the second and third is the result of segmentation, where the second result needs to be improved.

Difficulties in segmentation also arise when crowns touch or tilt a tall tree (for example, pine) to a lower one (for example, spruce), as a result of which there are cases when the top of a pine falls into one cluster intended for spruce due to their strong closure, which the DBSCAN algorithm does not cope well with. This case is designated as an erroneous selection of height and is not considered further due to the non-proliferation and strong differences between instances containing this defect.

3.3. Deep Learning Results

The segmented data was manually labeled into two large classes: "well-segmented tree" and "badly segmented tree". The distribution of classes was 472 instances of 152, respectively. The class of badly segmented trees included trees with crown defects (29 units), erroneous allocation of height (45 units), incorrect determination of the tree trunk (74 units), and the presence of a large number of extra parts of other trees (4 units). Most of the considered defect of "two crowns at different heights" (123 specimens in total) was attributed to "good" trees, since the defect did not seriously violate the general appearance of the trees, otherwise the case was attributed to the case of incorrect determination of the tree trunk.

This distribution shows that the performed segmentation allows us to identify a significant part (75.6%) of suitable point clouds of plantations for further processing to extract the taxation parameters of trees. However, there is also a part of the trees with defects (24.4%), which cannot be used for the final result of extracting structural parameters. The decision to identify these two classes of trees will allow you to detect bad instances and process them using other, more accurate algorithms for extracting individual objects. A study was also conducted to detect the considered defects in order to correctly process these exceptions in the future.

Table 1. Comparison of model accuracy depending on number of points per point cloud

	Number of points per point cloud, pts					
	128 pts	256 pts	512 pts	1024 pts	2048 pts	8192 pts
PointNet++	0.846	0.846	0.846	**0.885**	0.865	0.859
CurveNet	0.846	0.865	0.878	**0.878**	0.872	0.878

Table 2. Accuracy, recall and precision of models trained to detect defects

	PointNet++			CurveNet		
defect name	accuracy	recall	precision	accuracy	recall	precision
Bin2Crowns	0.923	0.903	0.869	0.910	0.907	0.846
BinErrTrunk	0.949	0.874	0.874	0.929	0.767	0.856

On average, a segmented cloud was obtained with a dimension of about 50000 points. To train a neural network, clouds of the same dimension are required, and such a large amount of points can only unreasonably increase the training time for the network parameters. Therefore, the number of points per tree was compared with the final accuracy on the test set. The data is compressed using the Farthest Point Sampling (FPS) algorithm [22], which consists in choosing the most distant point from the point under consideration, which in turn is chosen as the closest to the previously selected point. Such sampling provides much better coverage of the point cloud than uniform or random sampling. The classes were balanced by duplicating the data and then augmenting it by rotating the points around the z-axis and adding noise. The results of comparing accuracy models are presented in Table 1.

It can be seen that the best result is observed in PointNet++ with the optimal value of points per tree equal to 1024. Although large values of points give good results, a further increase was impractical, due to the increase in the memory size required to train one batch. As a result, it was necessary to reduce the batch size, which increased the training time. At the same time, there was a difference in the training time of PointNet++ and CurveNet by an average of 3–4 times in favor of PointNet++.

The selected optimal number of points per tree equal to 1024 was also used to train two more models to detect defects in segmented trees: binary classification of the defect "two crowns at different heights" (Bin2Crowns), binary classification of the defect "incorrect definition tree trunk" (BinErrTrunk). Table 2 shows the summary results of accuracy, recall and precision, the latter are given because the test sample contains unbalanced classes and accuracy may not fully reflect the results.

Thus, these models make it possible to separate the "good tree" from the "bad tree" with good accuracy, which will allow us to assess the correctness of the segmentation algorithm, fine-tune and select the hyperparameters of the above method for extracting individual trees. Also, the main defects of the plantations obtained as a result of segmentation were identified and the models were trained to identify the nature of the error. It is worth noting that the recall metric for the Bin2Crowns and BinErrTrunk models is 0.846 and 0.555, respectively, exclusively for classes with defects, that is, 28/31 and 10/18 defective trees were found in the test sample. These values indicate a good and acceptable detection result for error correction, respectively. So, for example, identified trees with a defect of two crowns at different heights (2Crowns) can pass the filtering of foliage points in the lower part of the tree, which will improve the calculation of the tree diameter at breast height and crown parameters. Detected trees with an incorrect tree trunk detection defect (ErrTrunk) can be processed using more accurate tree trunk detection, thereby allowing processing of the resulting exceptions in selected point clouds. This identification will improve the overall result of the distribution of trees into two large classes ("good tree" and "bad tree").

Comparing two models of working with unordered point clouds, preference is given to PointNet ++ with the best training results on the selected point size per cloud.

4. CONCLUSIONS

As a result of the work performed, the segmentation of the point cloud of a dense forest area of ground scanning was carried out using various methods for sampling points to reduce the dimension and subsequent segmentation by layers using the DBSCAN method based on the Voronoi partition. The resulting trees were evaluated manually and divided into two classes of suitable and unsuitable trees for their further processing to extract taxation parameters. The main defects obtained as a result of segmentation in the described manner are also noted. Models were trained using PointNet++ and CurveNet for binary classification of suitable and unsuitable trees, as well as to determine and identify the nature of the error for common segmentation defects. The obtained good detection results will allow us to use the models as an assessment of the correctness and quality of segmentation of point clouds of forest plots when changing the clustering hyperparameters. Automatic identification of trees with identified problems will allow them to be processed using other segmentation and filtering methods. Architectural solutions of neural networks will also allow training models to determine the rock composition of the site, which can be useful for segmentation in the second iteration, when all defects and rocks are found, with individual algorithms for each case.

FUNDING

The work was supported by the program of strategic academic leadership "Priority-2030" "PRIOR/SN/NU/22/SP1/4", no. 122070700013-5.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. Malankina E.L. and Tbsitsylin A.N., *Lekarstvennyye i efiromaslichnyye rasteniya* (Medicinal and Essential oil Plants), Ekaterinburg, 2016. ISBN 978-5-16-010957-2 (in Russian).
2. Mukhin, A., Danil, G., and Paringer, R., Semantic segmentation of hyperspectral imaging using Convolutional Neural Networks, *Opt. Mem. Neural Networks,* 2022, vol. 31 (Suppl. 1), pp. 38–47. https://doi.org/10.3103/S1060992X22050071
3. Ganchenko, V. and Doudkin, A., Agricultural vegetation monitoring based on aerial data using convolutional neural networks, *Opt. Mem. Neural Networks,* 2019, vol. 28, pp. 129–134. https://doi.org/10.3103/S1060992X1902005X
4. Zhen, Z., Quackenbush, L.J., and Zhang L., Trends in automatic individual tree crown detection and delineation—Evolution of LiDAR data, *Remote Sens.,* 2016, vol. 8, no. 4, p. 333. https://doi.org/10.3390/rs8040333
5. Liu, B. et al., Individual tree species classification using the pointwise MLP-based point cloud deep learning method, *Environ. Sci. Proc.,* 2022, vol. 22, no. 1, p. 19. https://doi.org/10.3390/IECF2022-13049
6. Maschler, J., Atzberger, C., and Immitzer, M., Individual tree crown segmentation and classification of 13 tree species using airborne hyperspectral data, *Remote Sens.,* 2018, vol. 10, no. 8, p. 1218. https://doi.org/10.3390/rs10081218
7. Michele, D., Frizzera, L., and Gianelle, D., Individual tree crown delineation and tree species classification with hyperspectral and LiDAR data, *PeerJ,* 2019, vol. 6: e6227. https://doi.org/10.7717/peerj.6227
8. Ramalho de Oliveira, L.F. et al., Moving to automated tree inventory: Comparison of UAS-derived lidar and photogrammetric data with manual ground estimates, *Remote Sens.,* 2020, vol. 13, no 1, p. 72. https://doi.org/10.3390/rs13010072
9. Peng Sun, Xuguang Yuan, and Dan Li, Classification of individual tree species using UAV LiDAR based on transformer, *Forests*, 2023, vol. 14, no. 3, p. 484. https://doi.org/10.3390/f14030484
10. Lin, Y.C. et al., Leaf-off and leaf-on UAV lidar surveys for single-tree inventory in forest plantations, *Drones,* 2021, vol. 5, no. 4, p. 115. https://doi.org/10.3390/drones5040115
11. Wang Yang et al. Individual tree segmentation and tree-counting using supervised clustering, *Comput. Electron. Agr.,* 2023, vol. 205, 107629. https://doi.org/10.1016/j.compag.2023.107629
12. Burt, A., Disney, M., and Calders, K., Extracting individual trees from lidar point clouds using treeseg, *Methods Ecol. Evol.,* 2019, vol. 10, no. 3, pp. 438–445. https://doi.org/10.1111/2041-210X.13121
13. Xu Shanshan, and Sheng Xu, Identification of street trees' main nonphotosynthetic components from mobile laser scanning data, *Opt. Mem. Neural Networks,* 2020, vol. 29, pp. 305–316. https://doi.org/10.3103/S1060992X20040062

14. Chen, S.W. et al., Sloam: Semantic lidar odometry and mapping for forest inventory, *IEEE Rob. Autom. Lett.,* 2020, vol. 5, no. 2, pp. 612–619.
https://doi.org/10.48550/arXiv.1912.12726
15. Grishin, I.A. et al., Tree Inventory with LiDAR Data, *International Conference on Neuroinformatics,* Cham: Springer, 2023, pp. 3–11.
https://doi.org/10.1007/978-3-031-19032-2_1
16. Qi, C.R. et al., Pointnet: Deep learning on point sets for 3d classification and segmentation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2017, pp. 652–660.
https://doi.org/10.48550/arXiv.1612.00593
17. Xiang, T. et al., Walk in the cloud: Learning curves for point clouds shape analysis, *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 2021, pp. 915–924.
https://doi.org/10.48550/arXiv.2105.01288
18. Grishin, I.A. and Terekhov, V.I., Procedure for locating trees and estimating diameters using LiDAR data, *2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE),* IEEE, 2023, vol. 5.
https://doi.org/10.1109/REEPE57272.2023.10086843
19. Chumachenko, S.I., Terekhov, V.I., Mitrofanov, E.T., and Grishin, I.A., An approach for estimating trees parameters using lidar data, *Dinamika slozhnykh system—XXI vek* (Dynamics of Complex Systems—XXI Century), 2022, vol. 16, no. 4, pp. 63–73.
https://doi.org/10.18127/j19997493-202204-06
20. Ester, M. et al., A density-based algorithm for discovering clusters in large spatial databases with noise, *Kdd,* 1996, vol. 96, no. 34.
21. Silverman, B.W., *Density Estimation for Statistics and Data Analysis,* CRC Press, 1986, vol. 26.
22. Eldar, Y., Lindenbaum, M., Porat, M., and Zeevi, Y.Y., The farthest point strategy for progressive image sampling, *IEEE Trans. Image Process.,* 1997, vol. 6, no. 9, pp. 1305–1315.
https://doi.org/10.1109/83.623193