Towards Intricate Stand Structure: A Novel Individual Tree Segmentation Method for ALS Point Cloud Based on Extreme Offset Deep Learning

Yizhuo Zhang, Hantao Liu, Xingyu Liu and Huiling Yu *

School of Computer Science, Changzhou University, Changzhou 213000, China; yzzhang@cczu.edu.cn (Y.Z.); s21150812037@smail.cczu.edu.cn (H.L.); s21150812040@smail.cczu.edu.cn (X.L.)
* Correspondence: yhl2016@cczu.edu.cn; Tel.: +86-177-127-70200

Abstract: Due to the complex structure of high-canopy-density forests, the traditional individual tree segmentation (ITS) algorithms based on ALS point cloud, which set segmentation threshold manually, is difficult to adequately cover a variety of complex situations, so the ITS accuracy is unsatisfactory. In this paper, a top-down segmentation strategy is adopted to propose an adaptive segmentation method based on extreme offset deep learning, and the ITS set aggregation strategy based on gradient change criterion is designed for the over-segmentation generated by random offset, and the precise ITS is realized. Firstly, the segmentation sub-plot is set as 25 m × 25 m, the regional point cloud and its treetop are marked, and the offset network is trained. Secondly, the extreme offset network is designed to carry out spatial transformation of the original point cloud, and each point is offset to the position near the treetop to obtain the offset point cloud with a high density at the treetop, which enhances the discrimination among individual trees. Thirdly, the self-adaptive mean shift algorithm based on average neighboring distance is designed to divide and mark the offset point cloud. Fourthly, the offset point cloud, after clustering, is mapped back to the original space to complete the preliminary segmentation. Finally, according to the gradient change among different canopies, the ITS aggregation method is designed to aggregate adjacent canopies with a gentle gradient change. In order to investigate the universality of the proposed method on different stand structures, two coniferous forest plots (U1, U2) in the Blue Ridge area of Washington, USA, and two mixed forest plots (G1, G2) in Bretten, Germany, are selected in the experiment. The learning rate of the deep network is set as 0.001, the sampled point number of the sub-plot is 900, the transformer dimension is 512 × 512, the neighboring search number of points is 16, and the number of up-sampling blocks is 3. Experimental results show that in mixed forests (G1, G2) with complex structures, the F-score of the proposed method reaches 0.89, which is about 4% and 7% higher than the classical SHDR and improved DK, respectively. In high-canopy-density areas (U2, G2), the F-score of the proposed method reaches 0.89, which is about 3% and 4% higher than the SHDR and improved DK, respectively. The results show that the proposed method has high universality and accuracy, even in a complex stand environment with high canopy density.

Keywords: ALS point cloud; individual tree segmentation; stand structure; extreme offset deep learning; mean shift; dynamic bandwidth; set aggregation

1. Introduction

Timely and effective access to forest growth information is highly significant for forest resource protection and the development of reasonable forest management plans [1]. Airborne laser lidar overcomes the shortcomings of traditional manual field survey, which is time-consuming and laborious [2]. Therefore, airborne laser lidar has been widely used in forestry surveys in recent decades [3,4]. However, due to the complex stand structures, it is difficult to cover all kinds of situations with traditional ITS algorithms by artificially setting segmentation rules, and the universality and accuracy of these ITS algorithms still

have great room for improvement [5,6]. Therefore, it is particularly important to design an ITS method with high universality and high accuracy.

From the perspective of segmentation strategy, ITS methods can be divided into two categories: bottom-up and top-down [7], and the top-down segmentation strategy is adopted in this paper. As the trunk points of ALS forest point cloud with high canopy density are sparse [8,9], the bottom-up method is seriously affected by noise when selecting seed points, resulting in poor segmentation results.

In the bottom-up method, the tree trunks in vertical space are extracted, and then the tree trunks are used as seed points to delineate the individual trees. Existing methods include the tree trunk detection fusion tree crown segmentation method [7], the two-dimensional tree trunk detection method [10], and the adaptive trunk detection method [11]. In general, due to the poor penetration rate of airborne radar pulses, the tree trunk points scanned in a high-canopy-density environment are sparse [8], and there may be a large deviation in the seed points, thus affecting the depiction of the canopy [7], leading to the poor segmentation of the bottom-up method in the high canopy density ALS forest point cloud.

In the top-down method, canopy features (e.g., gradient characteristics among canopies, treetops, etc.) are identified to delineate the individual trees. Common methods include the spatial horizontal distance rule algorithm (SHDR) [12], which has high segmentation accuracy but high time complexity, and is very sensitive to the selection of the horizontal distance threshold parameter; therefore, it has poor universality. The mark-controlled watershed segmentation algorithm [13] uses the mark-controlled watershed method to segment individual trees; the segmentation accuracy is poor when dealing with mixed forests with unobvious canopy characteristics. The DSM segmentation algorithm [14] uses the slope characteristics of the canopies to segment the conifers and even the mixed forest with weak canopy characteristics. However, it is easy to treat branches as individual trees, resulting in over-segmentation errors. The minimization energy function segmentation algorithm [15] can be applied to a variety of different forest types, but there are a huge number of parameters that need to be manually adjusted; therefore, the universality is poor. The adaptive mean shift algorithm [16] estimates the average canopy width of the whole forest land through the slope feature and uses it as the parameter of the mean shift algorithm to delineate the individual trees. This method is suitable for plantations with similar crown widths but performs poorly for mixed forests with large differences in crown width. In general, the universality and accuracy of the traditional top-down segmentation algorithm need to be improved when segmenting different kinds of forest lands [12,15].

Benefiting from the strong robustness, accuracy, and universality of deep learning, more and more scholars have begun to adopt deep learning methods to ITS [17]. However, due to the interweaving and complexity in high-canopy-density forests, the deep learning ITS methods still have under-segmentation and over-segmentation [18–20]. In recent years, point cloud deep learning has been widely used in point cloud segmentation tasks due to its strong universality and accuracy. Charles R. Qi et al. [21] proposed PointNet, which can directly segment point clouds without additional transformation but does not consider the relationship among points, which makes it difficult to analyze the intricate forest structure. PointNet++ [22] is a deep network based on PointNet. Although it enhances the ability to extract complex features by fusing features extracted from different receptive fields, its ability to perceive complex stand structures still needs to be improved. Charles R. Qi et al. [23] proposed VoteNet, which can offset each point to the corresponding instance center to increase the discrimination of different objects. However, the center offset loss function does not take into account the different contributions of each point to the loss function, resulting in an unsatisfactory offset result. PointGroup [24] segments each object through the clustering algorithm after the center offset but ignores the over-segmentation problem caused by the deep network randomness. Therefore, the segmentation accuracy still needs to be improved. Shaoyu Chen et al. [25] proposed HAIS, which combines

PointGroup with a set aggregation algorithm to reduce the over-segmentation problem. However, the HAIS set aggregation algorithm aggregates the point sets which are close to each other, which causes the point set of a tree to be aggregated into another adjacent tree, causing over-segmentation. Ashish Vaswani et al. [26] proposed Transformer, which has a strong perception ability for complex structures. On this basis, Hengshuang Zhao [27] proposed the Point Transformer, which performs point cloud segmentation for indoor objects. However, how to segment forest point clouds with complex structures remains to be solved. In general, although deep learning methods based on point clouds have become more and more mature, there are still many problems to be solved when applying them to segment individual trees [19].

Inspired by VoteNet and HAIS, if the relatively scattered data points are offset to the center of each tree to enhance the discrimination among different trees, then the ITS can be realized by applying the clustering algorithm. However, airborne laser lidar has poor penetration and cannot scan complete individual trees (especially in locations with severe canopy occlusion) [8]. At this time, the calculated individual tree center point has a huge gap with the actual individual tree center point, which seriously interferes with the learning effect of the deep network, resulting in a poor ability to enhance the discrimination among trees.

Considering the treetop points are distinct and stable, an extreme offset loss function for ITS is designed based on HAIS, which makes the points offset to their respective treetops to enhance the discrimination among different trees. Then, the density of the treetop increases significantly after points are offset to their treetops. On this basis, the mean shift algorithm is adopted to segment the individual trees preliminarily. Due to the deep network randomness, the offset effect of some points is poor. Directly using the mean shift algorithm for clustering may suffer from serious under-segmentation and over-segmentation problems. Considering the treetop density increases significantly after extreme offset, the average proximity distance is adaptively generated as the bandwidth of the mean shift algorithm so that the dense points near the treetop are clustered into one class. At the same time, the isolated points with poor offset are clustered separately to eliminate the under-segmentation error as much as possible. Finally, in order to reduce the over-segmentation error, the ITS set aggregation based on the gradient change criterion is designed to merge the sets with close neighbors and gentle gradient change.

The remainder of the paper is organized as follows. Section 2 describes the key steps of the data preprocessing and methodology. Section 3 describes the experimental data and evaluation metrics. Section 4 carries out a validity experiment, ablation experiment, and contrast experiment and analyzes the results. The discussion of the proposed method is given in Section 5. Finally, concluding remarks are given in Section 6.

In general, the key contributions of our work are as follows.

- Extreme offset deep learning method. Aiming at the problem of incomplete artificial feature extraction ability in traditional ITS segmentation algorithm, an extreme offset deep learning method is proposed. The forest point cloud features are automatically extracted through the extreme offset deep network, and the points are offset to the vicinity of the corresponding extreme points to enhance the discrimination among neighboring individual trees.
- Dynamic bandwidth. Aiming at the problem that the mean shift algorithm cannot adaptively determine bandwidth in a spatial transformed offset point cloud, a dynamic bandwidth calculation strategy based on average nearest neighbor distance is designed. This strategy can automatically determine the bandwidth of the mean shift algorithm without any prior knowledge, which enhances the universality of the mean shift algorithm.
- ITS set aggregation. Aiming at the over-segmentation problem caused by the randomness of the deep network, considering the characteristics that the canopy gradient changes sharply among different trees and gently within the same tree, the ITS set ag-

gregation based on gradient change is designed to improve the segmentation accuracy in complex woodlands.

2. Methods

Based on the HAIS framework, the proposed method focuses on optimizing the steps of offset, clustering, and set aggregation for ITS and realizes extreme offset, adaptive clustering, and ITS set aggregation, respectively. The specific process is shown in Figure 1, which includes preprocessing, extreme offset, mean shift, spatial mapping, set aggregation, and postprocessing. The process and function of specific links are as follows.

(1) Preprocessing. Data preprocessing included point cloud filtering, elevation normalization, dividing sub-plot (25 m × 25 m), point cloud denoising, down-sampling, and coordinate normalization.

(2) Extreme offset. The extreme offset deep learning method is used to perform a spatial transformation on the preprocessed point cloud, and each point is offset to the corresponding treetop to enhance the discrimination among different trees.

(3) Mean shift. In view of the large density of treetops in the offset point cloud, the self-adaptive mean shift algorithm based on average neighboring distance is adopted to cluster the offset point cloud, divide the offset point cloud set, and complete the labeling.

(4) Space mapping. The offset point cloud, after clustering and labeling, is mapped back to the original point cloud space to complete the preliminary segmentation.

(5) ITS set aggregation. Considering the characteristics that the gradient change among different tree canopies is sharp, while the same tree canopy is gentle, the adjacent canopies with gentle gradient change are aggregated to reduce the over-segmentation error.

(6) Postprocessing. The segmentation is completed after up-sampling and coordinate de-normalization. The flowchart of the proposed method is shown in Figure 1.

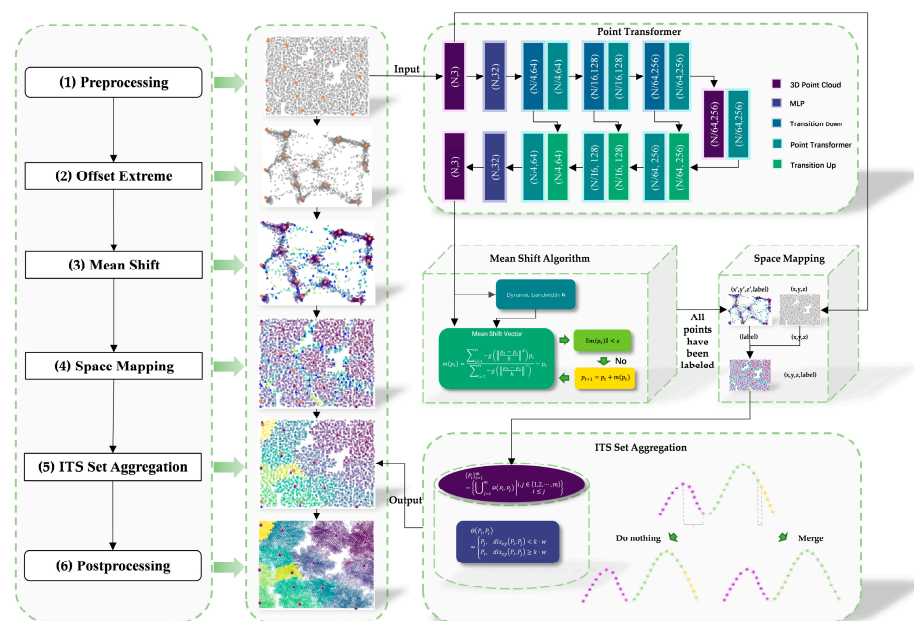



Figure 1. Flowchart of the extreme offset segmentation method. (1) Preprocessing. (2) Extreme offset. Each point is offset to the corresponding treetop to obtain an offset point cloud with large discrimination among different trees. (3) Mean shift. The offset point cloud is clustered. (4) Space mapping. The offset point cloud, after clustering and labeling, is mapped back to the original point cloud space. (5) ITS set aggregation. The adjacent canopies with gentle gradient change are aggregated to reduce the over-segmentation error. (6) Postprocessing. The segmentation is completed after up-sampling and coordinate de-normalization.

2.1. Data Preprocessing

After obtaining the ALS point cloud, the data are subjected to point cloud filtering, elevation normalization, subplot division, point cloud denoising, down-sampling, and coordinate normalization, and the whole process is shown in Figure 2. The specific steps of data preprocessing are as follows. (1) Point cloud filtering. The cloth simulation algorithm [28] is adopted to filter the point cloud, and the point cloud is divided into two types: ground points and non-ground points. (2) Elevation normalization. CloudCompare software is used to normalize the elevation of the point cloud to remove the influence of terrain relief. (3) Divide sub-plot. The sample plot is divided into 25 m × 25 m sub-plots. (4) Point cloud denoising. The residual outliers and branches at the sub-plot boundaries are removed by artificial visual denoising. (5) Down-sampling. Grids of size 0.5 m × 0.5 m are divided on the xoy plane, and the highest point in each grid is taken as the sampling point, and these sampling points are smoothed with a mean smoothing filter. (6) Coordinate normalization. After the coordinate normalization operation of the sub-plots, the data preprocessing is completed.

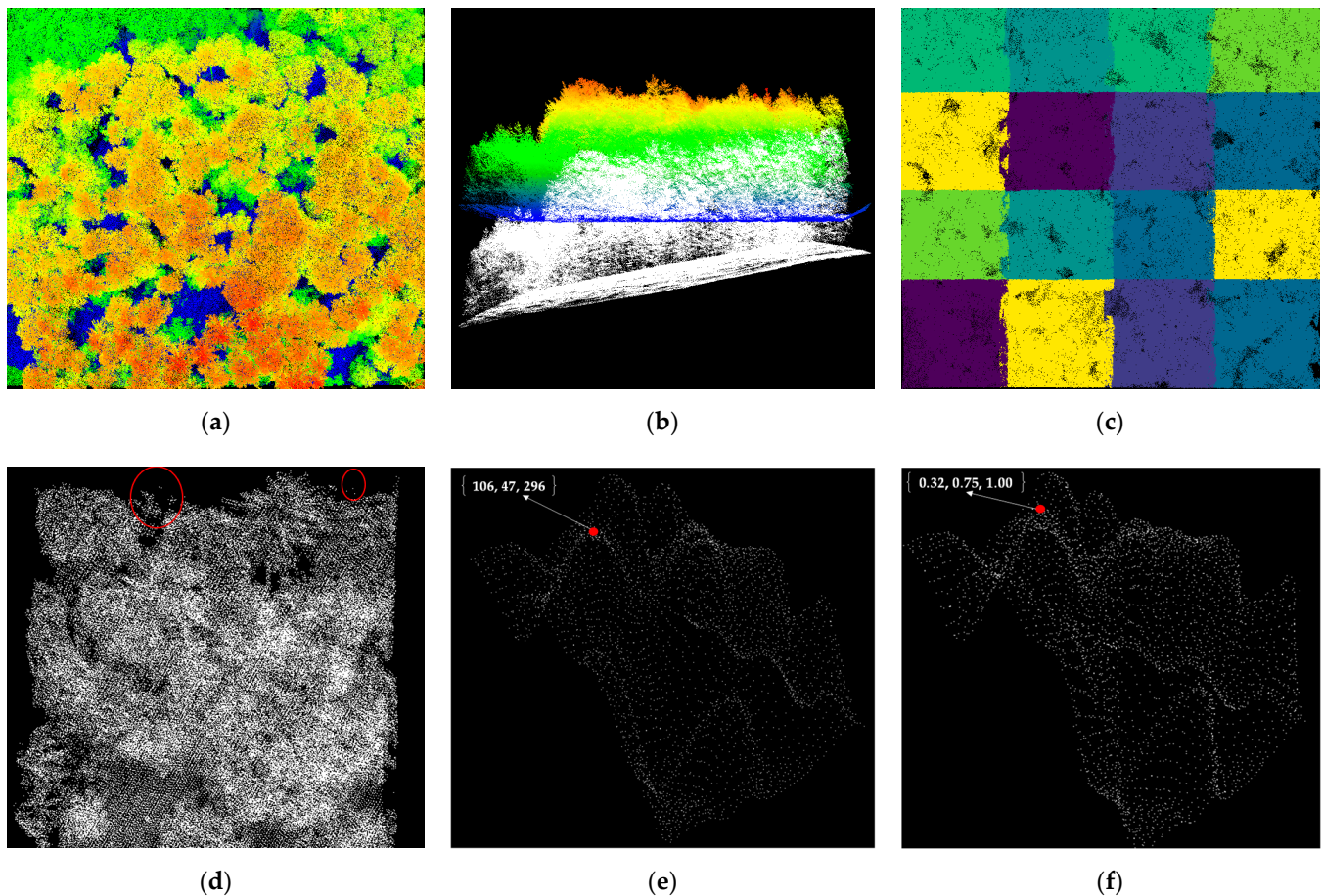



Figure 2. The main steps of data preprocessing. (**a**) Point cloud filtering; (**b**) Elevation normalization; (**c**) Divide subplot; (**d**) Point cloud denoising. The points surrounded by the red circle are the noise points at the boundary of the subplot; (**e**) Down-sampling; (**f**) Coordinate normalization.

2.2. Point Transformer with Extreme Loss Function

Point Transformer is a transformer-based deep network. The introduction of a self-attention mechanism also gives it a strong feature extraction ability for forest point clouds with complex structures. It consists of three core modules: a down-sampling module, an up-sampling module, and a transformer module. The Point Transformer network structure is shown in Figure 3.

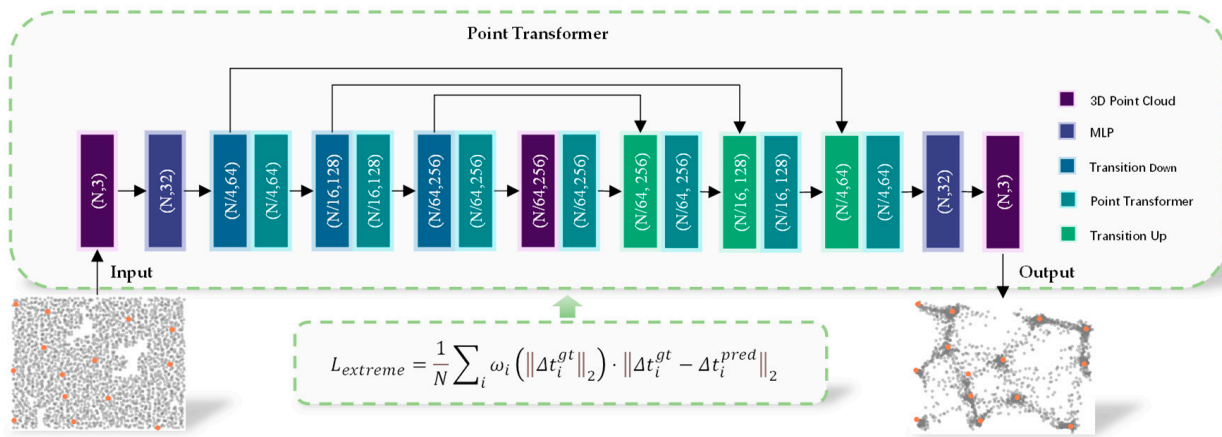



Figure 3. Point Transformer with extreme offset loss function.

The down-sampling module reduces the number of data points and extracts deeper feature information at the same time. Firstly, the FPS algorithm is used to sample the input point cloud to obtain the sampled point cloud. Secondly, in order to extract deeper feature information, each sampled point is taken as the center point, and the kNN graph is used to find the k nearest neighbor points around the center point. Finally, each center point and its neighbors form a set, which is input into the deep network for deep feature extraction.

The up-sampling module assigns deep features to non-sampled points. Firstly, the features of the point cloud composed of center points are extracted by MLP, BN, and ReLU. Secondly, the inverse distance weight interpolation algorithm is used to interpolate the features of each center point to its neighboring point. Finally, the features of the corresponding down-sampling block are aggregated to the corresponding up-sampling block by skip connection.

The Transformer module strengthens the local semantic awareness of each point to the points around it. It consists of two MLPs and a Transformer layer. The Transformer layer is shown in Equation (1):

$$y_i = \sum_{x_j \in X(i)} \varrho\big(\gamma\big(\beta\big(\varphi(x_i),\ \psi(x_j)\big)+\delta\big)\big) \odot \alpha\big(x_j+\delta\big)\,, \tag{1}$$

where β is a relation function; ϱ is a normalization function such as softmax; γ is an MLP with two linear layers and a ReLU nonlinearity; φ, ψ and α are pointwise feature transformations, such as linear projection or MLP; $X(i)$ is the set of X consisting of the k neighbors of x_i; x_i and x_j is the feature vector; y_i is the output feature; δ is the position encoding function, and its expression is $\delta = \theta(p_i - p_j)$, where, p_i and p_j are the 3D coordinates of points i and j, respectively; the encoding function θ is an MLP with two linear layers and a ReLU nonlinear layer.

The extreme offset loss function can measure the error between the predicted offset value and the true offset value so as to guide the update of the offset network. By continuously adaptively adjusting the parameters of the offset network to minimize the loss function, the offset network can gradually improve the accuracy of the predicted offset results and improve the model performance.

The proposed extreme offset loss function is shown in Equations (2) and (3):

$$L_{extreme} = \frac{1}{N} \sum_i \omega_i\big(\|\Delta t_i^{\text{gt}}\|_2\big) \cdot \|\Delta t_i^{gt} - \Delta t_i^{pred}\|_2\,, \tag{2}$$

$$\omega_i(X) = 2^{\|X\|_2} - 1, \tag{3}$$

where $L_{extreme}$ is the loss value of the loss function; N is the total number of points; $\|\Delta t_i^{gt}\|_2$ is smaller, so they should contribute less to the loss and therefore have less weight; Δt_i^{gt} is

the true offset of the point to the extreme point. Δt_i^{pred} is the offset predicted by the network for each point; X is a vector; $\omega_i(X)$ is a function to calculate the dynamic weight according to the length of vector X. Points near the extreme of the tree are less dependent on extreme offset and $\|X\|_2$ denotes the length of the vector X.

2.3. Mean Shift Algorithm with Dynamic Bandwidth

After the extreme offset, the scattered points gather toward the corresponding treetop, the density of the treetop increases significantly, and the density of the non-tree top decreases significantly. According to this density discrimination, the mean shift algorithm with dynamic bandwidth is adopted to cluster the offset point cloud, divide the offset point cloud set, and complete the labeling.

The principle of the mean shift algorithm [29] is to use the probability density to obtain the local optimal solution, and the schematic diagram of the mean shift process is shown in Figure 4. Firstly, the mean shift vector of the current center point is calculated. Secondly, the point is moved to the end of the mean shift vector. Then, it is taken as the new starting point and continued to move until the length of the mean shift vector is less than the allowable error. Finally, the iteration is terminated when all points have been marked. The mean shift vector is shown in Equation (4):

$$m(p_t) = \frac{\sum_{i=1}^{n} -g\left(\left\|\frac{p_t - p_i}{h}\right\|^2\right) p_i}{\sum_{i=1}^{n} -g\left(\left\|\frac{p_t - p_i}{h}\right\|^2\right)} - p_t,$$

(4)

where p_t is the center point; p_i is the point in the bandwidth; n is the number of points in the bandwidth; $g(\cdot)$ is the derivative of the RBF kernel function; h is the bandwidth, which is determined by the following dynamic bandwidth strategy.

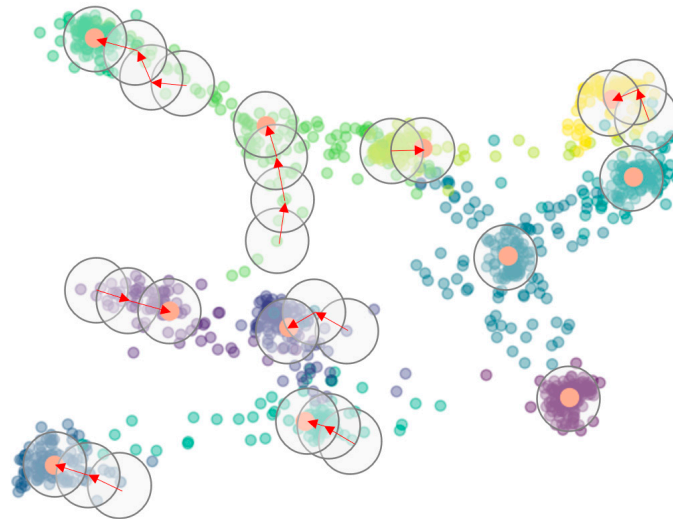



Figure 4. Schematic diagram of the mean shift process. The red arrow indicates the mean shift vector.

The dynamic bandwidth strategy is given in Equation (5). Firstly, the KDTree is constructed by the offset point cloud. KDTree is constructed to arrange the unordered point clouds in a certain order, which is convenient for fast and efficient retrieval. Second, the average distance of each point to its k nearest neighbors is calculated according to KDTree. Finally, these average distances are averaged, and the resulting average nearest neighbor distance is used as the dynamic bandwidth h.

$$h = \Lambda\left(\left\{\Upsilon_i^k\left(\{p_i'\}_{i=1}^n\right)\right\}_{i=1}^n\right),$$

(5)

where, $\left\{ p_i' \right\}_{i=1}^n$ is the offset point cloud; n is the total number of points. k is a number of nearest neighbors of a point, equal to 5% of the number of points in the point cloud. $\Upsilon_i^k(\cdot)$ is used to calculate the average distance of point p_i' to its k nearest neighbors. $\Lambda(\cdot)$ is used to find the average nearest neighbor distance, which is used as the dynamic bandwidth h.

After self-adaptive mean shift clustering on the offset point cloud, all the offset points are labeled. As the order of the points is not changed before and after the extreme offset, the label vector after the cluster in the offset point cloud is concatenated with the original point cloud space so that the offset point cloud after the cluster is mapped back to the original point cloud space, and the preliminary ITS is completed.

2.4. ITS Set Aggregation Based on Gradient Change

Due to the offset randomness caused by the deep network, the mean shift may individually cluster some branches, resulting in under-segmentation errors. In order to reduce the over-segmentation error, the ITS set aggregation algorithm is designed.

When two sets are aggregated, the smaller set should be aggregated into the larger set. In the process of extreme offset, most of the points are successfully offset to the treetop, but a small part of the points are invalid offset. Therefore, the larger scale of the set, the closer the set is to the correct segmentation result. Therefore, the set with a relatively large scale is regarded as part of the correctly segmented individual tree, and the set with a relatively small scale is regarded as a fragment generated by an invalid offset.

The ITS set aggregation algorithm based on gradient change criterion is as follows. If the horizontal distance between the fragment vertex point and the nearest neighbor point of the other set is greater than the preset bandwidth, then the gradient between the two sets is considered to change dramatically; therefore, they belong to different trees, respectively, and should not be aggregated, as shown in Figure 5a.

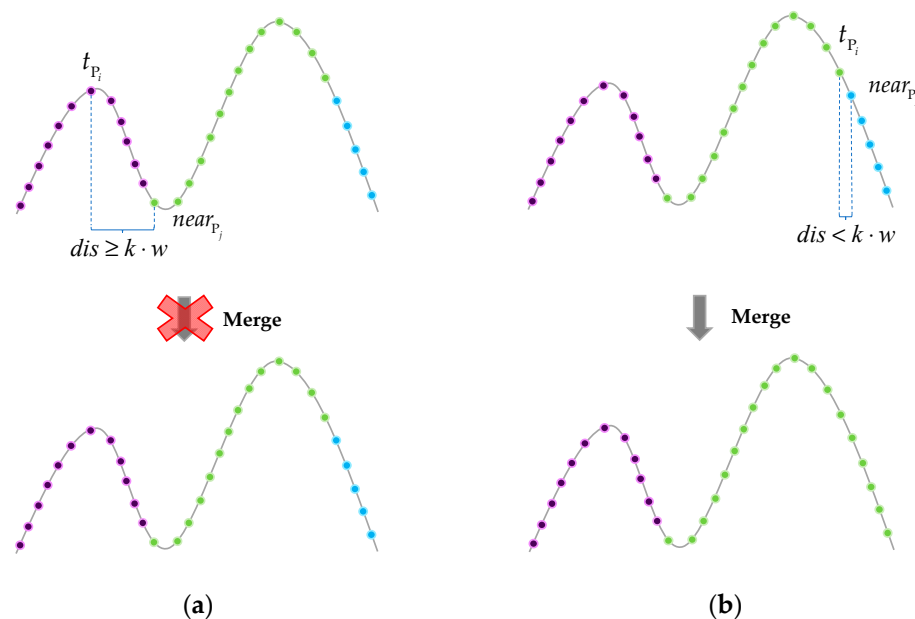



Figure 5. Schematic diagram of the ITS set aggregation. (**a**) The purple and green sets should not be aggregated; (**b**) The green and blue sets should be aggregated.

If the horizontal distance between the fragment vertex point and the nearest neighbor point of the other set is less than the preset bandwidth, then the gradient change between the two sets is considered to be gentle, and they should belong to the same tree; therefore,

the two sets are aggregated, as shown in Figure 5b. The ITS set aggregation is shown in Equations (6)–(8).

$$\{P_i\}_{i=1}^m = \left\{ \cup_{j=i}^m \Theta(P_i, P_j) \middle| \begin{matrix} i, j \in \{1, 2, \cdots, m\} \\ i \leq j \end{matrix} \right\}, \tag{6}$$

$$\Theta(P_i, P_j) = \begin{cases} P_j, & dis_{xoy}(P_i, P_j) < k \cdot w \\ P_i, & dis_{xoy}(P_i, P_j) \geq k \cdot w' \end{cases} \tag{7}$$

$$dis_{xoy}(P_i, P_j) = \begin{cases} \left\| t_{P_i} - near_{P_j} \right\|_2, & N_{P_i} < N_{P_j} \\ \left\| t_{P_j} - near_{P_i} \right\|_2, & N_{P_i} \geq N_{P_j}' \end{cases} \tag{8}$$

where $\{P_i\}_{i=1}^m$ represents the point set after set aggregation; m denotes the number of sets currently; $\Theta(P_i, P_j)$ decides whether to merge the two sets P_i, P_j; $dis_{xoy}(P_i, P_j)$ computes the Euclidean distance between two sets in the xoy plane; t_{P_i} is the highest point in the set P_i; $near_{P_j}$ is the nearest point from the set P_j to other set P_i; N_{P_i} denotes the total number of points in the set P_i; N_{P_j} denotes the total number of points in the set P_j; w is the raster width (0.5 by default); k is the preset bandwidth (1.5 w by default).

After the above processing, there may be noisy point sets with a small number of points. These noisy point sets will be directly aggregated into the nearest sets.

3. Experimental Data and Evaluation Methods

3.1. ALS Point Cloud in Germany and USA

In order to test and investigate the universality of the proposed method to different stand structures, we use the ALS point cloud provided by Weiser et al. [30] in Germany and http://forsys.cfr.washington.edu (accessed on 11 June 2018) in the USA. In Germany, the ALS point cloud was acquired by MILAN Geoservice GmbH using a RIEGL VQ-780i installed on a Cessna C207 aircraft. The nominal specifications of the ALS instrument are as follows. The precision is 20 mm in the 250 m scanning range, the laser beam divergence is less than 0.25 mrad, and the point density is 115.2 points/m². In the USA, a small footprint, discrete return lidar system was used to map the study area. The contractor used a Saab TopEye lidar system mounted on a helicopter to collect data over the study site. Flying height is 200 m, flying speed is 25 m/s, scanning swath width is 70 m, forward tilt is 8 degrees, laser pulse rate is 7000 points/s, maximum returns per pulse is 4, footprint diameter is 40 cm, and point density is 4.86 points/m².

3.2. Dataset

After data preprocessing, in order to better train and evaluate the network, 211 sub-plots 25 m $\times$ 25 m are randomly divided into a training set and validation set according to the ratio of 8:2, resulting in 169 plots as the training set (containing 1558 trees) and 42 sub-plots as the validation set (containing 382 trees). During training, the epoch of the extreme offset deep network is 50, the training time is about 2 h, the initial learning rate is set to 0.001, the RMSprop optimizer is used, and the smoothing constant alpha is set to 0.9. The StepLR method, which dynamically adjusts the learning rate, is used, where the learning rate adjustment factor gamma is set to 0.1.

Experimental area 1, the conifer forest in Washington, USA, is selected. In experimental area 2, a mixed forest in Bretten, Germany, is selected. The test set consists of four different plots, labeled U1, U2, G1, and G2, where U1 and U2 belong to the Blue Ridge region of the Congressional National Forest in western Washington. G1 and G2 belong to the mixed forest near Bretten, Germany. The four sample plots of the test set are shown in Figure 6.

(**a**) U1   (**b**) U2   (**c**) G1   (**d**) G2

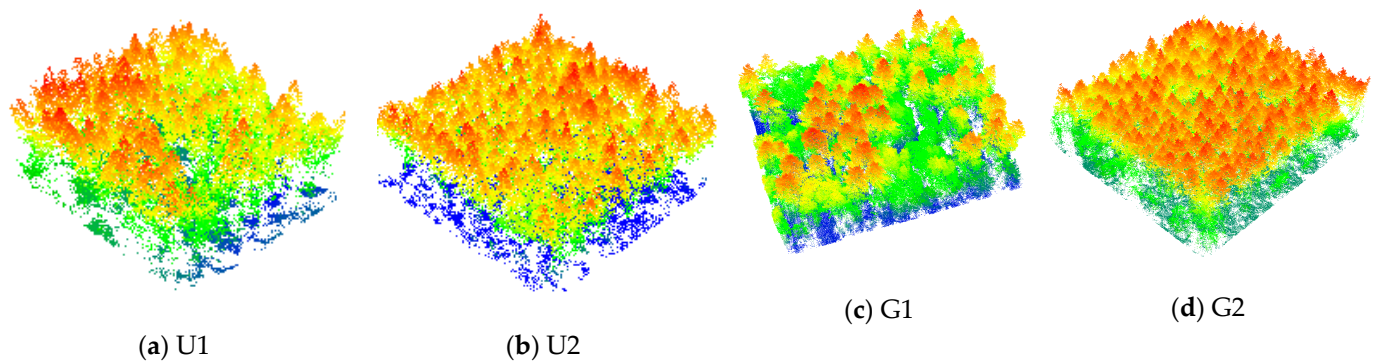



Figure 6. Schematic representation of the test plots.

Refer Canopy Density (RCD) is used to indicate the canopy density among different plots. RCD is obtained by dividing the number of grids containing points by the total number of grids, as shown in Equation (9):

$$\text{RCD} = \sum_{n}^{i} \frac{o_i}{n}, \tag{9}$$

where RCD is used to indicate the canopy density among different plots; n is the total number of grids; o_i is the marking bit, if any points exist in grid i, $o_i = 1$, otherwise $o_i = 0$.

In U1, U2, G1, and G2, the number of individual trees is 127, 186, 108, and 261; RCD is 61.58%, 69.09%, 93.25%, and 96.40%, respectively. Due to the difference in vegetation distribution and different acquisition equipment in the two areas, the point cloud density is quite different. Therefore, it is representative that the sample plots in these two areas are selected to test the method. A detailed description of the data is shown in Table 1.

Table 1. A detailed description of the test set.

Plot IDs	Number of Individual Trees	RCD (%)	Number of Sub-Plots
U1	132	61.58	16
U2	186	69.09	16
G1	108	93.25	12
G2	261	96.40	16

3.3. Assessment Criteria

In order to evaluate the performance of the proposed method, the matching strategy proposed by Kaiguang Zhao et al. [31] is used to pair the reference tree (true value) with the estimated tree (predicted value), as shown in Equation (10):

$$D = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + k \cdot \left(z_i - z_j\right)^2}, \tag{10}$$

where D is the relative distance between the reference tree and the estimated tree; (x_i, y_i, z_i) and (x_j, y_j, z_j) represent the treetop coordinates of the reference tree and the estimated tree, respectively. The estimated tree i and the reference tree j match successfully when the two three-dimensional spatial points representing the location information of the tree are closest. k is the weight for the height difference (0.5 by default).

Three metrics, recall (r), precision (p), and F-score (F), are used to evaluate the performance of the proposed method, as shown in Equations (11)–(13):

$$r = \frac{\text{MT}}{\text{MT} + \text{OE}}, \tag{11}$$

$$p = \frac{MT}{MT + CE}, \tag{12}$$

$$F = 2 \times \frac{r \times p}{r + p}, \tag{13}$$

where MT represents the one-to-one matching relationship between the reference tree and the estimated tree, that is, the correct segmentation; OE represents that some reference trees are not matched to the estimated tree, that is, an individual tree is considered as a branch belonging to other trees, so it is not detected, resulting in under-segmentation; CE indicates that the reference tree and the estimated tree have a one-to-many relationship; that is, an individual tree is divided into multiple trees, resulting in over-segmentation; r, p, and F are recall, precision, and F-score, respectively.

4. Results

4.1. Validity Experiment

The segmentation process and results of partial subgraphs in the test set are shown in Figure 7. Different colors represent different sets, where the red point represents the treetop of the actual individual tree, and the blue triangle represents the treetop of the predicted individual tree. Where (a) represents the forest point cloud after data preprocessing; (b) represents the offset point cloud obtained after extreme offset; (c) represents the offset point cloud with label obtained by mean shift; (d) shows the result after mapping back to the origin point cloud space; (e) represents the result after ITS set aggregation; (f) represents the final segmentation result after postprocessing.

(a)　(b)　(c)　(d)　(e)　(f)

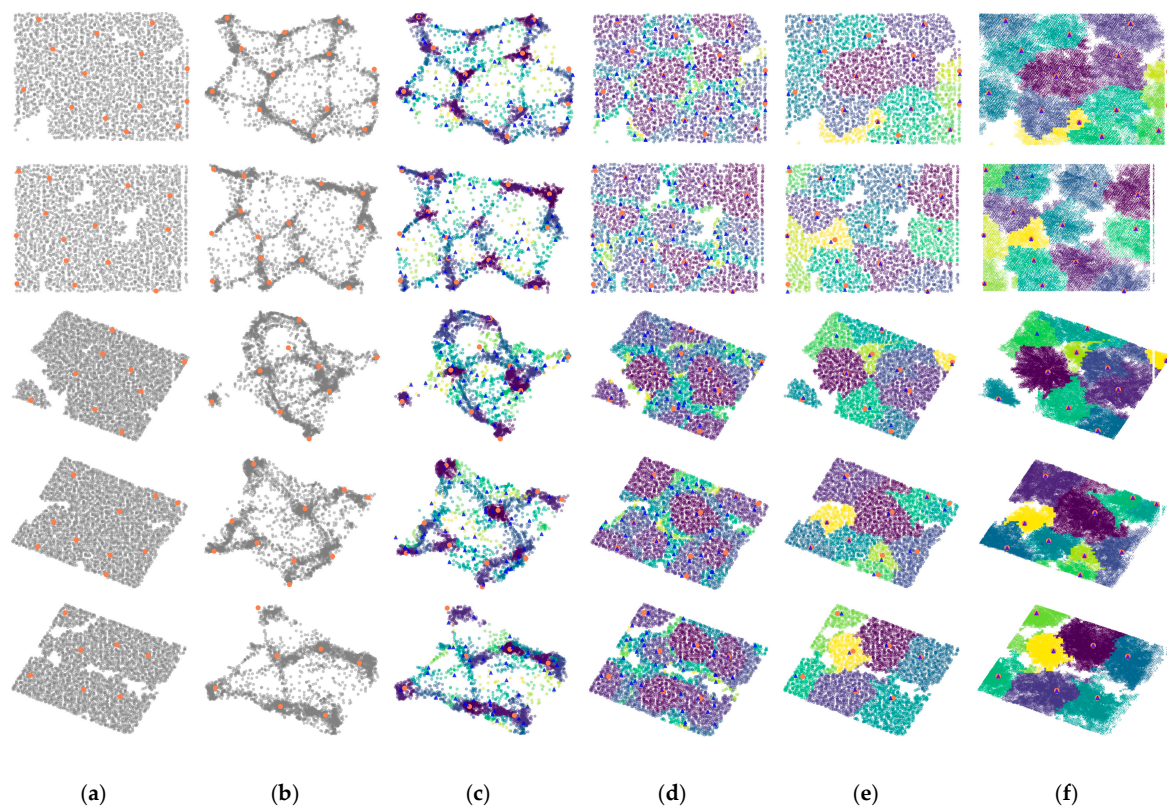



Figure 7. Partial segmentation results of the proposed method are shown, where the red circle represents the actual individual treetop, and the blue triangle represents the predicted individual treetop. (**a**) The point cloud after data preprocessing; (**b**) The offset point cloud after extreme offset; (**c**) The offset point cloud after mean shift cluster; (**d**) The point cloud after space mapping; (**e**) The point cloud after ITS set aggregation; (**f**) The origin point cloud after postprocessing.

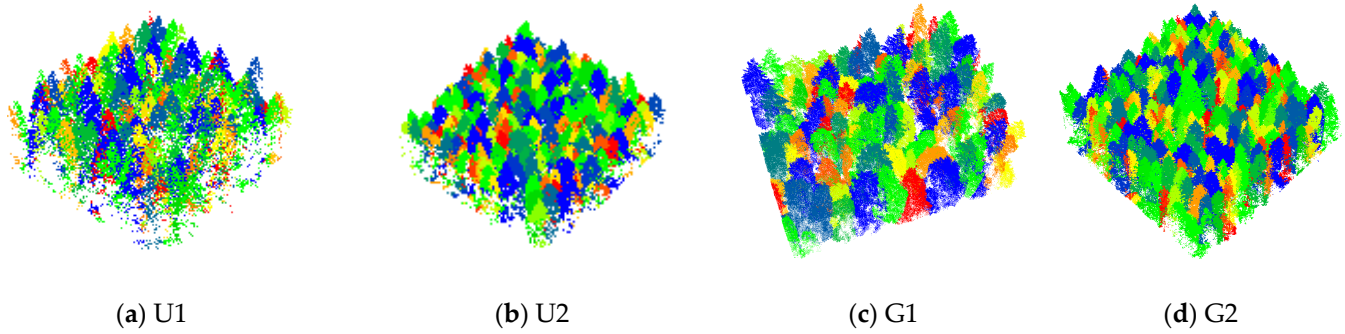
The 3D segmentation results of the proposed method are shown in Figure 8, where (a), (b) represent the sample plot of the Blue Ridge region of the United States, and (c), (d) represent the sample plot of the Bretten in Germany. The ITS results of the proposed method in four sample plots are shown in Table 2, where TN is the actual number of tree individuals; MT is the number of correctly segmented individual trees; OE is the number of under-segmented individual trees; CE is the number of over-segmented individual trees; Calculate *p*, *r*, and *F* according to Equations (11)–(13).

(**a**) U1   (**b**) U2   (**c**) G1   (**d**) G2

Figure 8. The overall 3D segmentation results of the proposed method are shown. (**a**) U1; (**b**) U2; (**c**) G1; (**d**) G2.

Table 2. Assessment of the accuracy of the proposed method in four plots.

Plot	TN [1]	MT [2]	OE [3]	CE [4]	*p*	*r*	*F*
U1	132	110	9	15	0.88000	0.92437	0.90164
U2	186	158	13	15	0.91329	0.92398	0.91860
G1	108	87	16	5	0.94565	0.87931	0.90667
G2	261	203	46	13	0.93981	0.81526	0.87312
Overall	687	558	84	48	0.91968	0.88573	0.90000

[1] The actual number of individual trees in the plot. [2] The number of correctly segmented individual trees. [3] The number of under-segmented individual trees. [4] The number of over-segmented individual trees.

4.2. Ablation Experiment

In order to verify the necessity of the three core steps of the proposed method, ablation experiments were designed, as shown in Table 3. Firstly, the advantage of extreme offset over center offset is explored. Secondly, the advantage of dynamic bandwidth mean shift clustering compared with fixed bandwidth HAIS clustering is explored. Finally, the advantage of the ITS set aggregation compared with the HAIS set aggregation are explored.

Table 3. For the ablation study of extreme offset, dynamic bandwidth, and ITS set aggregation.

Extreme Offset	Dynamic Bandwidth	ITS Set Aggregation	*p*	*r*	*F*
			0.54379	0.38114	0.43816
√			0.79675	0.72826	0.75743
√	√		0.87075	0.85266	0.85876
√		√	0.91801	0.76391	0.83285
√	√	√	0.91968	0.88573	0.90000

(1) Extreme offset

In order to verify the effectiveness of extreme offset, the deep network is trained by extreme offset and center offset, respectively. The loss curve obtained from the training and validation set is shown in Figure 9a,b, respectively. It can be seen that when using extreme offset, the loss value with the smallest final convergence value of the training set and the validation set is the smallest, which is 0.064 and 0.059, respectively. When using the center offset, the final convergence loss values of the training set and the validation set are 0.11

and 0.088, respectively. It shows that the introduced extreme offset strategy can effectively enhance the robustness of the deep network.

(**a**)   (**b**)

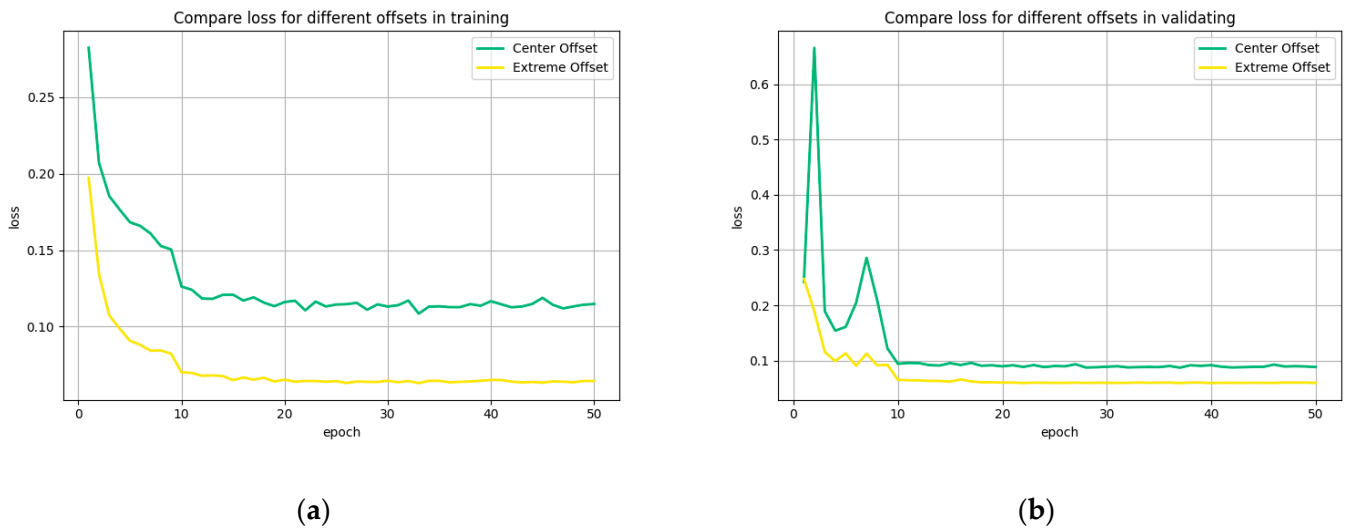



Figure 9. The curve of loss value and epoch when training and validation are performed in different offset ways. (**a**) training loss and epoch; (**b**) validation loss and epoch.

In addition, Table 4 compares the effect of using extreme offset or center offset on the segmentation accuracy of the proposed method. When extreme offset is adopted, the average F-score of the proposed method is 0.75, which is much higher than the average F-score of center offset (0.43). It shows the effectiveness and superiority of extreme offset.

Table 4. Extreme offset vs. Center offset.

Method	Plots	RCD	TN [1]	MT [2]	OE [3]	CE [4]	*p*	*r*	*F*
	U1	61.58%	132	51	52	28	0.64557	0.49515	0.56044
	U2	69.09%	186	57	75	91	0.38514	0.43182	0.40714
Center offset	G1	93.25%	108	26	56	18	0.59091	0.31707	0.41270
	G2	96.40%	261	62	159	50	0.55357	0.28054	0.37237
	Overall		687	196	342	187	0.54379	0.38114	0.43816
	U1	61.58%	132	86	19	27	0.76106	0.81905	0.78899
	U2	69.09%	186	112	58	16	0.87500	0.65882	0.75168
Extreme offset	G1	93.25%	108	63	18	17	0.78750	0.77778	0.78261
	G2	96.40%	261	142	74	44	0.76344	0.65741	0.70647
	Overall		687	403	169	104	0.79675	0.72826	0.75743

[1] The actual number of individual trees in the plot. [2] The number of correctly segmented individual trees. [3] The number of under-segmented individual trees. [4] The number of over-segmented individual trees.

(2)   Dynamic bandwidth

Mean shift with dynamic bandwidth and HAIS clustering with fixed bandwidth are used, respectively, and the segmentation results of the proposed method are shown in Figure 10. After many experiments, when the HAIS clustering with fixed bandwidth is set to 0.05, the segmentation result is the best, and the average F-score of the four plots is 0.78, 0.75, 0.78, and 0.70, respectively. When the mean shift with dynamic bandwidth is used, the average F-score of the algorithm in the four plots reaches 0.85, 0.90, 0.84, and 0.84, respectively, which is significantly higher than the HAIS clustering with fixed bandwidth, as shown in Table 5. The results show that the mean shift with dynamic bandwidth is effective and superior.

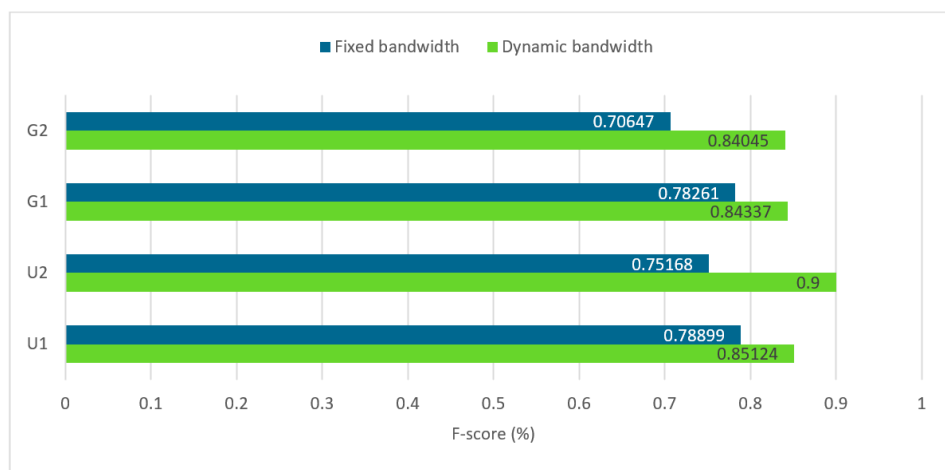



Figure 10. Fixed bandwidth vs. Dynamic bandwidth.

Table 5. Fixed bandwidth vs. Dynamic bandwidth.

Bandwidth	Plots	RCD	TN [1]	MT [2]	OE [3]	CE [4]	*p*	*r*	*F*
Fixed bandwidth	U1	61.58%	132	86	19	27	0.76106	0.81905	0.78899
	U2	69.09%	186	112	58	16	0.87500	0.65882	0.75168
	G1	93.25%	108	63	18	17	0.78750	0.77778	0.78261
	G2	96.40%	261	142	74	44	0.76344	0.65741	0.70647
	Overall		687	403	169	104	0.79675	0.72826	0.75743
Dynamic bandwidth	U1	61.58%	132	103	9	27	0.79231	0.91964	0.85124
	U2	69.09%	186	153	18	16	0.90533	0.89474	0.90000
	G1	93.25%	108	70	15	11	0.86420	0.82353	0.84337
	G2	96.40%	261	187	55	16	0.92118	0.77273	0.84045
	Overall		687	513	97	70	0.87075	0.85266	0.85876

[1] The actual number of individual trees in the plot. [2] The number of correctly segmented individual trees. [3] The number of under-segmented individual trees. [4] The number of over-segmented individual trees.

(3)   ITS set aggregation

Table 6 and Figure 11 show that after the introduction of ITS set aggregation, the segmentation accuracy of the proposed method for the four plots is effectively improved, the F-score in the four plots reach 0.86, 0.83, 0.80, and 0.82, respectively, the average F-score in all plots reach 0.83. In the coniferous forest plots (U1, U2) and mixed forest plots (G1, G2), the average F-score is improved by 16% and 14%, respectively, compared with the HAIS set aggregation.

Table 6. HAIS set aggregation vs. ITS set aggregation.

Method	Plots	RCD	TN [1]	MT [2]	OE [3]	CE [4]	*p*	*r*	*F*
HAIS set aggregation	U1	61.58%	132	86	19	27	0.76106	0.81905	0.78899
	U2	69.09%	186	112	58	16	0.87500	0.65882	0.75168
	G1	93.25%	108	63	18	17	0.78750	0.77778	0.78261
	G2	96.40%	261	142	74	44	0.76344	0.65741	0.70647
	Overall		687	403	169	104	0.79675	0.72826	0.75743
ITS set aggregation	U1	61.58%	132	100	20	12	0.89286	0.83333	0.86207
	U2	69.09%	186	133	47	6	0.95683	0.73889	0.83386
	G1	93.25%	108	65	24	7	0.90278	0.73034	0.80745
	G2	96.40%	261	183	60	16	0.91960	0.75309	0.82805
	Overall		687	481	151	41	0.91801	0.76391	0.83285

[1] The actual number of individual trees in the plot. [2] The number of correctly segmented individual trees. [3] The number of under-segmented individual trees. [4] The number of over-segmented individual trees.

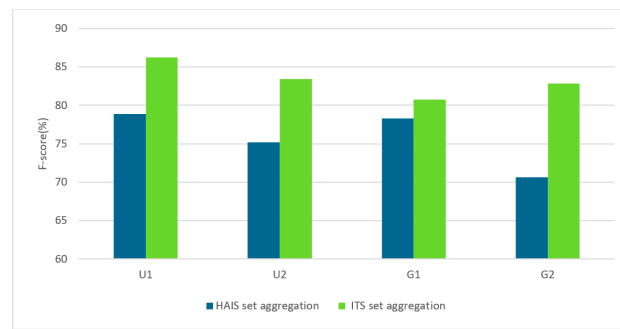



Figure 11. HAIS set cluster algorithm vs. ITS set cluster algorithm.

5. Discussion

5.1. Comparison with Existing Methods

Multiple ITS methods are used in the four plots to test separately, and the test results are shown in Table 7. The average *p*, *r*, *F*, and *time(min)* of the SHDR method [12] in the four plots reached 0.84, 0.85, 0.84, and 23.3, respectively. The average *p*, *r*, *F*, and *time(min)* of the DK method [10] in the four plots reached 0.66, 0.59, 0.62, and 12.5, respectively. The average *p*, *r*, *F*, and *time(min)* of the Improved DK method in the four plots reached 0.83, 0.86, 0.85, and 14.9, respectively. The average *p*, *r*, *F*, and *time(min)* of the HAIS method [25] in the four plots reached 0.79, 0.72, 0.75, and 13.7, respectively. The average *p*, *r*, *F*, and *time(min)* of the proposed method in the four plots reached 0.91, 0.88, 0.90, and 13.5, respectively.

Table 7. Comparison of individual tree segmentation accuracy of SHDR method, DK method, HAIS method, Improved DK method, and the proposed method in four plots.

Method	Plots	RCD	TN [1]	MT [2]	OE [3]	CE [4]	*p*	*r*	*F*	*Time(Min)* [5]
SHDR method	U1	61.58%	132	91	25	12	0.88350	0.78448	0.83105	4.5
	U2	69.09%	186	141	22	26	0.84431	0.86503	0.85455	4.7
	G1	93.25%	108	94	13	18	0.83929	0.87850	0.85845	6.6
	G2	96.40%	261	194	27	42	0.82203	0.87783	0.84902	7.5
	Overall		687	520	87	98	0.84728	0.85146	0.84826	23.3
DK method	U1	61.58%	132	56	40	38	0.59574	0.58333	0.58947	1.9
	U2	69.09%	186	74	48	82	0.47436	0.60656	0.53237	2.2
	G1	93.25%	108	61	35	18	0.77215	0.63542	0.69714	3.9
	G2	96.40%	261	67	52	13	0.83750	0.56303	0.67337	4.5
	Overall		687	258	175	151	0.66993	0.59708	0.62308	12.5
Improved DK method	U1	61.58%	132	102	7	16	0.86441	0.93578	0.89868	2.7
	U2	69.09%	186	140	19	22	0.86420	0.88050	0.87227	2.6
	G1	93.25%	108	76	14	23	0.76768	0.84444	0.80423	4.1
	G2	96.40%	261	184	42	34	0.84404	0.81416	0.82883	5.5
	Overall		687	502	82	95	0.83508	0.86872	0.85084	14.9
HAIS method	U1	61.58%	132	86	19	27	0.76106	0.81905	0.78899	2.2
	U2	69.09%	186	112	58	16	0.87500	0.65882	0.75168	2.6
	G1	93.25%	108	63	18	17	0.78750	0.77778	0.78261	4.1
	G2	96.40%	261	142	74	44	0.76344	0.65741	0.70647	4.8
	Overall		687	403	169	104	0.79675	0.72826	0.75743	13.7
The proposed method	U1	61.58%	132	110	9	15	0.88000	0.92437	0.90164	2.1
	U2	69.09%	186	158	13	15	0.91329	0.92398	0.91860	2.5
	G1	93.25%	108	87	16	5	0.94565	0.87931	0.90667	4.0
	G2	96.40%	261	203	46	13	0.93981	0.81526	0.87312	4.9
	Overall		687	558	84	48	0.91968	0.88573	0.90000	13.5

[1] The actual number of individual trees in the plot. [2] The number of correctly segmented individual trees. [3] The number of under-segmented individual trees. [4] The number of over-segmented individual trees. [5] The time consumption of the method.

The spatial horizontal distance rule (SHDR) is a top-down method. This method has high segmentation accuracy, but it has high time complexity and is sensitive to the selection of the horizontal distance threshold parameter, which leads to poor universality.

DBSCAN with K-means (DK) is a bottom-up method. The advantage of this method is that it can segment the trees with obvious trunk characteristics well. However, the trunk points scanned by airborne laser lidar in forests with high canopy density are sparse, which leads to the inaccurate center points found by the DBSCAN algorithm, and then affects the final segmentation results.

HAIS is a deep learning method for instance segmentation. This method can achieve good results when segmenting indoor datasets. However, indoor objects are far apart from each other, and individual trees are close to each other. If HAIS is directly applied to ITS, it may cause serious over-segmentation and under-segmentation problems.

The Improved DBSCAN and K-means method (Improved DK) is an improved strategy that integrates extreme offset deep learning and the DK method. Experiments show that the introduction of extreme offset makes the segmentation accuracy of the DK method significantly improved in high-canopy-density forests.

As shown in Figure 12a, the overall *p*, *r*, and *F* of the proposed method in the four plots are higher than other comparison methods. As shown in Figure 12b, the F-score of the proposed method is only slightly higher than the improved DK method in U1 with low canopy density. However, in the actual forest resource survey, the difficulty is usually the forest with high canopy density. Although the F-score of the proposed method is only slightly higher than the Improved DK method in U1, the F-score of the proposed method is much higher than the other comparison methods in other plots with high canopy density, which indicates the superiority of the proposed method. In addition, the average F-score of the Improved DK method (85%) is much higher than the average F-score of the DK method (68%). Although the proposed method consumes slightly more time (13.5 min total) than the DK method (12.5 min total) in the four plots, the segmentation accuracy of the proposed method is significantly higher than the DK method.

(**a**) (**b**)

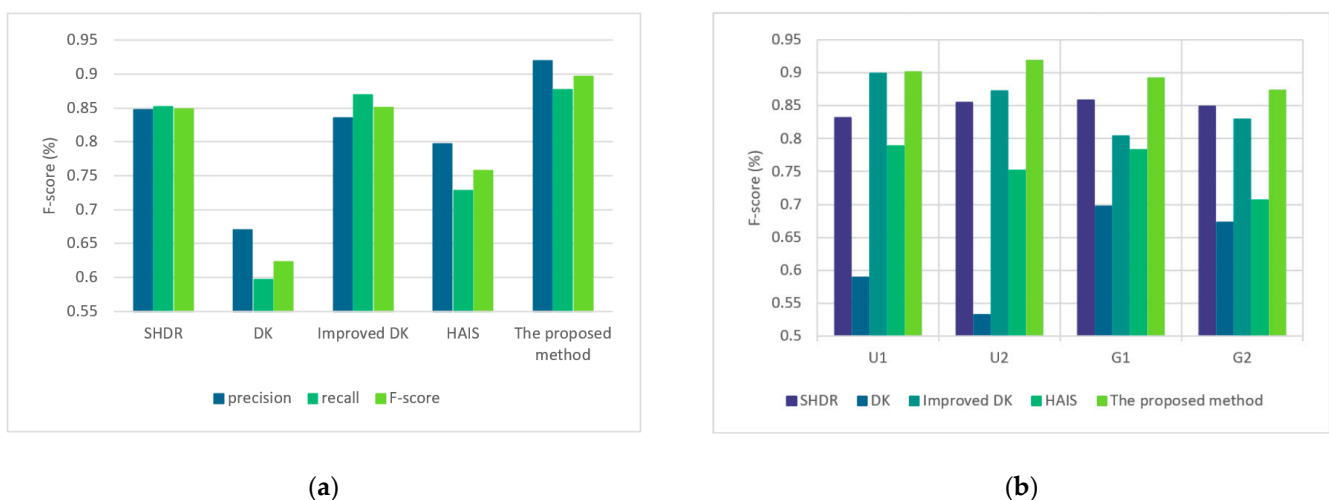



Figure 12. Comparative test results. (**a**) Overall precision, recall, and F-score on different methods; (**b**) F-score on different plots and methods.

This indicates that the proposed method is effective, and the extreme offset deep learning can significantly improve the segmentation accuracy of the bottom-up method and has good extendibility in the high-canopy-density woodland of the ALS point cloud.

5.2. Analysis of Extreme Offset, Dynamic Bandwidth Strategy, and ITS Set Aggregation

Extreme offset analysis. Due to airborne laser lidar having poor penetration and cannot scan complete individual trees (especially in locations with severe canopy occlusion) [8], at this time, the calculated individual tree center point has a huge gap with the actual individual tree center point, which seriously interferes with the learning effect of the deep network, resulting in the poor ability to enhance the discrimination among trees.

Considering the treetop points are distinct and stable, an extreme offset loss function for ITS is designed based on HAIS, which makes the points offset to their respective treetops to enhance the discrimination among different trees. Then, the density of the treetop increases significantly after points are offset to their treetops.

Dynamic bandwidth strategy analysis. Due to the deep network randomness, the offset effect of some points is poor. Directly using the mean shift algorithm for clustering may suffer from serious under-segmentation and over-segmentation problems. Considering the treetop density increases significantly after extreme offset, the average proximity distance is adaptively generated as the bandwidth of the mean shift algorithm so that the dense points near the treetop are clustered into one class. At the same time, the isolated points with poor offset are clustered separately to eliminate the under-segmentation error as much as possible.

ITS set aggregation analysis. Due to the erroneous offset point, this inevitably causes an under-segmentation issue of the mean shift algorithm. The HAIS set aggregation approach merely aggregates sets that are close to one another, which may result in the improper aggregation of neighboring sets of different trees. In order to effectively prevent neighboring sets of various trees from being aggregated, the ITS set aggregation approach takes into account the gradient features of the individual trees. This method reduces over-segmentation mistakes and boosts segmentation accuracy.

5.3. Potential Improvements

In the future, we intend to improve the proposed method from the following aspects. (1) Improving deep networks. In order to better adapt to the ITS task, researchers can continue to improve the deep learning networks and make improvements to the existing ITS methods to improve the accuracy and robustness of the model so that the model may better handle the tree morphology with complex structures. (2) Application of leafless data. Using leafless data to extract trunk points and send them as features into deep networks for learning may further improve the accuracy and robustness of ITS tasks because leafless data can provide more information on the trunk structure and better reflect the morphological characteristics of trees. (3) Data Augmentation. A suitable data augmentation strategy may enhance the learning ability of deep networks and further improve the generalization ability and robustness of the model. Data augmentation can increase the diversity of the dataset, reduce overfitting, and improve the accuracy and robustness of the model by processing the data. In conclusion, by enhancing the deep network, applying leafless data, and data augmentation, which may better serve applications in tree-related fields, the accuracy and resilience of ITS tasks can be further enhanced.

6. Conclusions

In this paper, an ITS method based on extreme offset deep learning is designed for complex stand structures. The key steps are as follows: (1) preprocessing; (2) extreme offset; (3) self-adaptive clustering; (4) space mapping; (5) set aggregation; (6) postprocessing. In order to verify the universality and accuracy of the proposed method, coniferous forest plots in the Blue Ridge area of Washington, USA, and mixed forest plots near Bretten, Germany, are selected as the test plots. The point density is low, and the stand structure is relatively simple in the coniferous forest of the Blue Ridge in the USA. The point density is high, and the stand structure is complex in the mixed forest of Bretten in Germany. The test of the algorithm in these two areas can effectively verify the universality and accuracy of the algorithm. The experimental results show that after the introduction of step (2) extreme offset, it can effectively enhance the discrimination among different trees and then improve the ITS accuracy (the average p, r, and F reach 0.79, 0.72, and 0.75, respectively). After the introduction of step (2) and step (3) adaptive bandwidth strategy, it has better segmentation accuracy and can better adapt to complex scenes and changing environments (the average p, r, F reach 0.87, 0.85, 0.85, respectively). After the introduction of step (2) and step (5) ITS set aggregation, it can effectively reduce the over-segmentation error and improve the

segmentation accuracy of the proposed method (the average *p*, *r*, *F* reach 0.91, 0.76, 0.83, respectively). After the introduction of steps (2), (3) and (5), the average *p*, *r*, and *F* of the proposed method in all plots reach 0.91, 0.88, and 0.90, respectively. In the future, we intend to improve the proposed method from the following aspects. (1) Improve the deep network to better adapt to the ITS task, (2) the tree trunk point coordinates may be extracted using the leafless data and fed into the deep network as features for learning, and (3) appropriate data augmentation methods may be used to enhance the learning ability of the network. In summary, the experimental results effectively verify the universality and accuracy of the proposed method and show the superiority and application potential in segmenting complex forest environments by comparing them with other algorithms.

Author Contributions: H.L. and Y.Z. designed the method and performed the experiments; X.L. analyzed and preprocessed the data; H.Y. and Y.Z. reviewed and revised the manuscript; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available on https://doi.pangaea.de/10.1594/PANGAEA.942856 (accessed on 1 January 2022) and http://forsys.cfr.washington.edu/FUSION/fusion_overview.html (accessed on 11 June 2018).

Acknowledgments: We acknowledge the Weiser, H et al., and the pacific northwest research station USDA forest service for providing data for this research.

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

References

1. Luo, H.; Shu, Q.; Xi, L.; Huang, J.; Liu, Y.; Yang, Q. Research Progress of Lidar in Forest Biomass Estimation. *Green Sci. Technol.* **2022**, *24*, 23–28.
2. Pu, Y.; Xu, D.; Wang, H.; Li, X.; Xu, X. A New Strategy for Individual Tree Detection and Segmentation from Leaf-on and Leaf-off UAV-LiDAR Point Clouds Based on Automatic Detection of Seed Points. *Remote Sens.* **2023**, *15*, 1619. [CrossRef]
3. Hu, Y.; Gao, H.; Xia, W.; Huang, Q.; Chen, Z.; Wang, D. Exploration of Individual Tree Segmentation Method in Subtropical Coniferous Forest Using Airborne Radar Data. *Appl. Laser* **2021**, *41*, 1301–1309.
4. Xu, D.; Wang, H.; Xu, W.; Luan, Z.; Xu, X. LiDAR applications to estimate forest biomass at individual tree scale: Opportunities, challenges and future perspectives. *Forests* **2021**, *12*, 550. [CrossRef]
5. Kong, M. *Single Tree Extraction of Airborne Forest Point Cloud Based on Deep Learning of Candidate Regions*; China University of Geosciences: Beijing, China, 2021.
6. Hui, Z.; Jin, S.; Li, D.; Ziggah, Y.Y.; Liu, B. Individual Tree Extraction from Terrestrial LiDAR Point Clouds Based on Transfer Learning and Gaussian Mixture Model Separation. *Remote Sens.* **2021**, *13*, 223. [CrossRef]
7. Wang, X.; Zhang, Y.; Luo, Z. Combining trunk detection with canopy segmentation to delineate single deciduous trees using airborne LiDAR data. *IEEE Access* **2020**, *8*, 99783–99796. [CrossRef]
8. Huo, L.; Lindberg, E.; Holmgren, J. Towards low vegetation identification: A new method for tree crown segmentation from LiDAR data based on a symmetrical structure detection algorithm (SSD). *Remote Sens. Environ.* **2022**, *270*, 112857. [CrossRef]
9. Huiling, L.; Xiaoli, Z.; Ying, Z.; Yunfeng, Z.; Hui, L.; Longyang, W. Review on Individual Tree Detection Based on Airborne LiDAR. *Laser Optoelectron. Prog.* **2018**, *8*, 82805. [CrossRef]
10. Chen, Q.; Wang, X.; Hang, M.; Li, J. Research on the improvement of single tree segmentation algorithm based on airborne LiDAR point cloud. *Open Geosci.* **2021**, *13*, 705–716. [CrossRef]
11. Fu, H.; Li, H.; Dong, Y.; Xu, F.; Chen, F. Segmenting Individual Tree from TLS Point Clouds Using Improved DBSCAN. *Forests* **2022**, *13*, 566. [CrossRef]
12. Li, W.; Guo, Q.; Jakubowski, M.K.; Kelly, M. A new method for segmenting individual trees from the lidar point cloud. *Photogramm. Eng. Remote Sens.* **2012**, *78*, 75–84. [CrossRef]
13. Hu, B.; Li, J.; Jing, L.; Judah, A. Improving the efficiency and accuracy of individual tree crown delineation from high-density LiDAR data. *Int. J. Appl. Earth Obs. Geoinf.* **2014**, *26*, 145–155. [CrossRef]
14. Hamraz, H.; Contreras, M.A.; Zhang, J. A robust approach for tree segmentation in deciduous forests using small-footprint airborne LiDAR data. *Int. J. Appl. Earth Obs. Geoinf.* **2016**, *52*, 532–541. [CrossRef]

15. Yun, T.; Jiang, K.; Li, G.; Eichhorn, M.P.; Fan, J.; Liu, F.; Chen, B.; An, F.; Cao, L. Individual tree crown segmentation from airborne LiDAR data using a novel Gaussian filter and energy function minimization-based approach. *Remote Sens. Environ.* **2021**, *256*, 112307. [CrossRef]

16. Yan, W.; Guan, H.; Cao, L.; Yu, Y.; Li, C.; Lu, J. A self-adaptive mean shift tree-segmentation method using UAV LiDAR data. *Remote Sens.* **2020**, *12*, 515. [CrossRef]

17. Hu, Y.; Luo, D.; Hua, K.; Lu, H.; Zhang, X. Overview on deep learning. *CAAI Trans. Intell. Syst.* **2019**, *14*, 1–19.

18. Weinstein, B.G.; Marconi, S.; Bohlman, S.; Zare, A.; White, E. Individual tree-crown detection in RGB imagery using semi-supervised deep learning neural networks. *Remote Sens.* **2019**, *11*, 1309. [CrossRef]

19. Chen, X.; Jiang, K.; Zhu, Y.; Wang, X.; Yun, T. Individual tree crown segmentation directly from UAV-borne LiDAR data using the PointNet of deep learning. *Forests* **2021**, *12*, 131. [CrossRef]

20. Wang, F.; Bryson, M. Tree Segmentation and Parameter Measurement from Point Clouds Using Deep and Handcrafted Features. *Remote Sens.* **2023**, *15*, 1086. [CrossRef]

21. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.

22. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems. In Proceedings of the Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.

23. Qi, C.R.; Litany, O.; He, K.; Guibas, L.J. Deep hough voting for 3d object detection in point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2019, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9277–9286.

24. Jiang, L.; Zhao, H.; Shi, S.; Liu, S.; Fu, C.W.; Jia, J. Pointgroup: Dual-set point grouping for 3d instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and PATTERN Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4867–4876.

25. Chen, S.; Fang, J.; Zhang, Q.; Liu, W.; Wang, X. Hierarchical aggregation for 3d instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 15467–15476.

26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**. [CrossRef]

27. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 16259–16268.

28. Zhang, W.; Qi, J.; Wan, P.; Wang, H.; Xie, D.; Wang, X.; Yan, G. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sens.* **2016**, *8*, 501. [CrossRef]

29. Chen, W.; Hu, X.; Chen, W.; Hong, Y.; Yang, M. Airborne LiDAR Remote Sensing for Individual Tree Forest Inventory Using Trunk Detection-Aided Mean Shift Clustering Techniques. *Remote Sens.* **2018**, *10*, 1078. [CrossRef]

30. Weiser, H.; Schäfer, J.; Winiwarter, L.; Krašovec, N.; Fassnacht, F.E.; Höfle, B. Individual tree point clouds and tree measurements from multi-platform laser scanning in German forests. *Earth Syst. Sci. Data* **2022**, *14*, 2989–3012. [CrossRef]

31. Zhao, K.; Suarez, J.C.; Garcia, M.; Hu, T.; Wang, C.; Londo, A. Utility of multitemporal lidar for forest and carbon monitoring: Tree growth, biomass dynamics, and carbon flux. *Remote Sens. Environ.* **2018**, *204*, 883–897. [CrossRef]