

# 3D Point Cloud Classification Based on Discrete Conditional Random Field

Xinying Liu<sup>1,2</sup>, Hongjun Li<sup>3(✉)</sup>, Weiliang Meng<sup>1,2</sup>, Shiming Xiang<sup>1,2</sup>, and Xiaopeng Zhang<sup>1,2(✉)</sup>

<sup>1</sup> NLPR-LIAMA, Institute of Automation, CAS, Beijing, China  
xiaopeng.zhang@ia.ac.cn

<sup>2</sup> University of Chinese Academy of Sciences, Beijing, China

<sup>3</sup> Beijing Forestry University, Beijing, China  
lihongjun69@bjfu.edu.cn

**Abstract.** The advantage of the conditional random field (CRF) lies in the construction of the discriminant model and efficient parameter optimization. In the topic of the classification of three-dimensional point cloud, the parameters of the CRF are usually learnt through Gradient Descent and Belief Propagation, in order to optimize the objective energy function of the CRF. These optimization methods do not guarantee the highest global classification accuracy with a high classification accuracy on the smaller classes. In addition, differential features of the point cloud are not sufficiently utilized. In this paper, we use the local geometric shape features to construct the CRF, including the nearest neighbor tetrahedral volume, Gaussian curvature, the neighbourhood normal vector consistency and the neighbourhood minimum principal curvature direction consistency. We propose four discrete criteria for CRF parameter optimization to design the explicit functions, and present concrete solution procedures, in which Monte Carlo method and supervised learning method are employed to estimate the CRF parameters iteratively. Experimental results show that our method can be applied to the classification of the scene of 3D point cloud with plants, especially under the proposed second criterion that maximizing the accuracy with interclass weights. It can be used to improve significantly the classification accuracy of small scale point sets when different classes have great disparity in number.

**Keywords:** Point cloud classification · Feature extraction · Conditional Random Field · Parameter optimization criterion · Confusion matrix

## 1 Introduction

Point cloud refers to a set of multiple vertices with coordinate information, and the vertices position coordinates set sampled from the surface of 3D objects is a typical type of this data. Besides the position coordinate information, other auxiliary information can be attached to each vertex such as the color

information (RGB) or the intensity information. The point cloud data can be from scanning real scenes via terrestrial LiDAR [1] or the airborne 3D laser scanner [2], or reconstructing from the photos based on visual methods [3,4]. With the increasing performance of three-dimensional laser scanners, cameras and other hardware devices in recent years, the acquisition of point cloud data from the objects become easier than ever before, which in turn to promote point cloud analysis and process for widely applying in the industries of measurement, aviation, architecture, gardens, urban planning, digital city, monitoring and etc. Point cloud analysis and processing involves a variety of basic tasks, including point cloud denoising, feature calculation, segmentation, key points and lines recognition, object detection, surface reconstruction [5], registration and visualization, etc. The reconstruction based on the three-dimensional point cloud can generate more realistic three-dimensional models, and bring a stronger sense of immersion in the application of digital entertainment, education and so on.

The point cloud classification is a hot topic of the current research with the deepening needs of the applications. For example, through the analysis of the scene point cloud data, the roads, vegetation, buildings, bridges, vehicles, pedestrians and even windows can be identified. Compared with the image-based target recognition methods, the point cloud data realistically records the three-dimensional information of the objects in the scene, and we can get more accurate parsing results of the scene, which is conducive to the scene semantic analysis and reproduction.

As the basis for understanding the scene, point cloud classification is a difficult problem because of the huge data size, data noise, loss of partial information, and unknown geometric topology of data. Although a lot of related works have been done, the accuracy of point cloud classification is still to be further improved. On the one hand, the existing point cloud feature description methods don't fully introduce the local discriminant information; on the other hand, the existing point cloud classification methods don't combine the global geometric information with the discriminant classification methods effectively.

We adopt the discriminant information based on the both local neighborhood and global model, and improve the performance of point cloud classification by constructing new cloud point feature description methods and make a classification. Specifically, the idea of nearest neighbor classification is introduced into the local neighborhoods of each vertex in the cloud data, and the local geometric shape description method is constructed based on the nearest neighbor tetrahedron volume, the Gaussian curvature, and the pairwise potentials. Then the Conditional Random Field (CRF) model for point cloud data is built and the CRF parameter optimization criteria are designed to improve the classification performance. Experiments show that our discrete criteria for CRF parameter optimization have a better effect on improving the quality of the point cloud classification results, especially the proposed second criterion that based on the maximization accuracy with interclass weights.

## 2 Related Works

The related works mainly include three aspects: point cloud feature description, point cloud classification based on CRF, and point cloud classification based on pattern analysis.

### 2.1 Point Cloud Feature Description

The point cloud classification is mainly dependent on the local features of each vertex in the point cloud. The feature vector can be constructed by the 3D shape contexts and the harmonic shape contexts for point cloud shape recognition methods with high dimension eigenvectors [6]. In order to reduce the eigenvectors' dimension, the histogram of the local point cloud feature [7,8] and the surface feature histograms [9] can be used for analysis. The local features of the point cloud that used to build the histograms can be points scatterness, linearity and surfaceness [10]. In addition, normal vectors and the local flatness degree can also be taken as the features for point cloud classification [11]. Weinmann et al. [12] proposed a series of local shape features including the k-nearest neighbor's height difference, density, linearity, planarity, scattering, omni-variance, anisotropy, eigen-entropy, sum of eigenvalues, change of curvature and two-dimensional projection features. Yang et al. [13] proposed using local point cloud depth, density and deviation angles between normals to build local feature descriptor for point cloud recognition. Plaza-Leiva et al. [14] gave a general framework for supervised learning classifiers in which the support region features are defined by the voxels themselves.

### 2.2 Point Cloud Classification Based on CRF

The Conditional Random Field (CRF) is a probability graph model proposed by Lafferty et al. [15], which can be used to segment or classify sequence data and text. Then Kumar et al. [16] brought it in computer vision and constructed Discriminative Random Fields for region labeling and classification of natural images. Munoz et al. [17] proposed a 3D point cloud classification method based on Directional Associated Markov Network. Shapovalov et al. [2] extended this method to Non-Associated Markov Networks to distinguish vegetation points, and in turn a high-order Markov network is built by Najafi et al. [18]. Besides the position coordinates, echo amplitude, echo width and other property values associated with the vertex can also be used for point cloud classification. Rutzinger et al. [19] used these information to identify the higher plants (trees and shrubs) better. Niemeyer et al. [20] argued that the CRF is a generalization of the Markov random field, and using CRF is more suitable than the Markov random field. Husain et al. [21] classified the indoor point cloud by establishing a CRF. Wolf et al. [22] used the random forest classifier to initialize the unary potentials of the CRF and the parameters related to the pairwise potentials are deduced from the training set, in turn to accomplish indoor point cloud classification. Niemeyerar et al. [23] construct a 2-level CRF model for point

cloud classification, where the second layer can correct the initial classification error. Lang et al. [24] proposed an adaptive CRF model to achieve the outdoor point cloud classification. Ni et al. [25] use reflectance-based features, descriptor-based features, and the geometric features for Random Forests in order to classify airborne laser scanning point clouds.

As a typical classification method based on graph discrimination, the graph structure of CRF makes connections between vertices in the point cloud data through local neighbors, which in turn to provide a global discriminant learning model for classification. The core of the CRF lies in the construction of the discriminant model and the efficient parameters optimization. Existing methods mainly use the idea of discriminant regression to construct the discriminant model, which cannot make the best of the distribution of point cloud data. In addition, they mainly use the descending method to solve the parameters, making the result very sensitive to the initial value, gradient updating rate and other factors, and the parameter optimization process is inefficient.

### 2.3 Point Cloud Classification Based on Pattern Analysis

Besides the CRF model, pattern analysis can also be used for point cloud classification. In recent years, most commonly used discriminant analysis methods in the field of pattern analysis and machine learning have been preliminarily transferred to apply in point cloud classification tasks. Specifically, Wang et al. [26] and Zhang et al. [27] used the Latent Dirichlet Allocation model for point cloud classification; Rodriguezcuenca et al. [28] used anomaly detection algorithm to identify column objects; Deep Convolutional Neural Networks (DCNN) [29] and spindle descriptors [30] were also proposed for point cloud classification method. Kang et al. [31] employed the geometric features from the point clouds with the spectral features from the optical images to train an optimal Bayesian network structure for classification. Maligo et al. [32] proposed a two-layer classification model, with the first layer that a GMM trained in an unsupervised manner and the second layer that corresponds to a grouping of the intermediary classes into final classes. Li et al. [33] showed that the local geometric features of the nearest neighbor tetrahedron volume, Gaussian curvature, point potential energy (neighbourhood normal vectors consistency and the neighbourhood minimum principal curvature directions consistency) had a good effect on point cloud classification. Based on the pattern analysis of these features, we construct our CRF to achieve point cloud classification and improves the classification accuracy.

## 3 Construction of CRF

In this paper, we use the position coordinate information of the point cloud to construct the CRF in order to determine the classification type of each vertex. Let  $(x_i, y_i, z_i)$  denote position of the  $i$ -th vertex  $P_i$  in the point cloud, and assume that the point cloud data can be divided into  $m$  class, denoted by  $\mathcal{L} = \{l_1, l_2, \dots, l_m\}$  respectively. First, we need to extract the local shape features of each vertex in the point cloud.

### 3.1 Local Shape Features Extraction

The local shape features of the vertex are the basis of the point cloud classification [33], and we mainly use the nearest neighbor tetrahedral volume, the Gaussian curvature, and pairwise potentials as the local shape features for our classification.

**Nearest neighbor tetrahedral volume.** The nearest neighbor tetrahedral volume  $\mathcal{V}_i$  of vertex  $P_i$  describes the density and position relation of the nearest neighbors, which can be calculated as the mixed product of the three vectors constructed by three the nearest neighbor  $P_{1_i}, P_{2_i}, P_{3_i}$  of the vertex  $P_i$  as follows:

$$\mathcal{V}_i = \frac{1}{6} \|(\overrightarrow{P_i P_{1_i}} \times \overrightarrow{P_i P_{2_i}}) \cdot \overrightarrow{P_i P_{3_i}}\| \quad (1)$$

**Gaussian curvature.** The Gaussian curvature  $\mathcal{K}_i$  reflects the bending degree of the local surface where vertex  $P_i$  is located, and the value equals to the product of two principal curvatures  $k_{1_i}$  and  $k_{2_i}$ , i.e.,  $\mathcal{K}_i = k_{1_i} k_{2_i}$ . The calculation of the principal curvature for the vertex  $P_i$  in the point cloud can be approximately calculated as in [34] based on the position information of the  $k$  nearest neighbor vertices of  $P_i$ .

**Pairwise potentials.** Pairwise potentials reflect the relationship between neighborhood pairs, including the neighbourhood normal vectors consistency  $\sigma_i$  and the neighbourhood minimum principal curvature directions consistency  $\delta_i$ . The consistency of the normal vectors between an arbitrary vertex  $P_i$  and its  $k$  nearest neighbor  $\{P_{j_i}, j = 1, 2, \dots, k\}$  can be used directly for point cloud classification. The neighbourhood normals vector consistency  $\sigma_i$  can be denoted by the variance ( $\theta_{j_i}$ ) of the cosine of the angles between the normal vector of  $P_i$  and normal vector of its  $k$  nearest neighbourhood  $P_{j_i}$  where  $j = 1, 2, \dots, k$ .

$$\sigma_i = \frac{1}{k} \sum_{j=1}^k (\theta_{j_i} - \bar{\theta}_i)^2 \quad (2)$$

where  $\theta_{j_i} = \vec{n}_i \cdot \vec{n}_{j_i}$  and  $\bar{\theta}_i = (\sum_{j=1}^k \theta_{j_i})/k$ . Here  $\vec{d}_i$  is the normal vector of  $P_i$ , and  $\vec{n}_{j_i}$  is the normal vector of  $P_{j_i}$  ( $j = 1, 2, \dots, k$ ).

Similarly, we define the neighbourhood minimum principal curvature directions consistency  $\delta_i$  as

$$\delta_i = \frac{1}{k} \sum_{j=1}^k (\tau_{j_i} - \bar{\tau}_i)^2 \quad (3)$$

where  $\tau_j = \vec{d}_i \cdot \vec{d}_{j_i}$  ( $j = 1, 2, \dots, k$ ) and  $\bar{\tau}_i = (\sum_{j=1}^k \tau_{j_i})/k$ . Here  $\vec{d}_i$  is the minimum principal curvature direction of  $P_i$ , and  $\vec{d}_{j_i}$  is the minimum principal curvature direction of  $P_{j_i}$  ( $j = 1, 2, \dots, k$ ).

According to the above equations, we can get the local shape features of an arbitrary vertex  $P_i$ , i.e., the nearest neighbor tetrahedron volume, Gaussian curvature, and pairwise potentials, and our CRF can be constructed based on these features.

### 3.2 CRF Objective Energy Function

Essentially, the CRF is an undirected graph, denoted as  $G = \langle V, E \rangle$ , where  $V$  is the vertices set in which every vertex  $P_i$  in the point cloud  $\Omega$  is taken as a vertex of the CRF, and  $E$  is the edge set. The user can specify an edge between two points within a certain distance [21]. In our case, considering that the distances between vertices in the point cloud may be non-uniform due to the occlusion, this edge definition may lead to an unconnected graph  $G$ , which makes the number of vertices in a subgraph is too small to determine its local shape, even with many isolated points. Therefore, we use the  $k$  nearest neighbors  $\{P_{j_i}, j = 1, 2, \dots, k\}$  of vertex  $P_i$  to define  $k$  edges correspondingly. Note that for vertex  $P_i$  and vertex  $P_j$ ,  $P_j$  may be one of  $P_i$ 's  $k$  nearest neighbors, while the reverse may be not true. For each vertex in the graph  $G$ , the corresponding degree is no less than  $k$ . The typical value of  $k$  is 20 in our tests.

As CRF  $G$  has the Markov property, i.e., the local shape at any vertex  $P_i$  is mainly determined by the  $P_i$ 's nearest neighbor vertices  $P_{j_i}$ , we can construct the objective energy function  $\mathcal{E}(\mathcal{L}, \Omega; w)$  based on our extracted features with the neighborhood volume  $\mathcal{V}_i$ , the Gaussian curvature  $\mathcal{K}_i$ , the point potential energy (including the neighbourhood normal vectors consistency  $\sigma_i$  and the neighbourhood minimum principal curvature directions consistency  $\delta_i$ ) as follows:

$$\mathcal{E}(\mathcal{L}, \Omega; w) = w_1^{(l)} \mathcal{V}_i + w_2^{(l)} \mathcal{K}_i + w_3^{(l)} \sigma_i + w_4^{(l)} \delta_i \quad (4)$$

Define  $w = \{w_1^{(l)}, w_2^{(l)}, w_3^{(l)}, w_4^{(l)}\}$ , and the optimal solution of  $w$  can be obtained by maximizing the following conditional probability  $P(\mathcal{L}|\Omega, w)$  [21].

$$P(\mathcal{L}|\Omega, w) = \frac{e^{\mathcal{E}(\mathcal{L}, \Omega; w)}}{\sum_{l \in \mathcal{L}} e^{\mathcal{E}(\mathcal{L}, \Omega; w)}} \quad (5)$$

The optimization process of conditional random field (CRF) parameter  $w$  can be found in Sect. 3.4, and the final optimized CRF parameter value is denoted as  $w^*$ .

### 3.3 Point Cloud Classification

For the given point cloud  $\Omega$  and the CRF parameter  $w^*$ , the vertex is classified according to the maximum value of the calculated posterior probability [21].

$$\mathcal{L}^* = \arg \max_{l \in \mathcal{L}} P(\mathcal{L}|\Omega, w^*) \quad (6)$$

If both the vertices number  $n$  and the classification number  $m$  are large, then the number of CRF parameters will be larger, and the computational complexity will be higher. In this case, belief propagation algorithm can be used for approximate calculation [35].

### 3.4 Parameter Learning

Parameter optimization is the core to the CRF. For a given training set  $\Omega_0$  and the identity labels of each vertex  $\mathcal{L}_0$ , the likelihood function can be constructed by regularized log likelihood probability [21, 36].

$$L(w) = \lambda \|w\|^2 - \sum_{l_i \in \mathcal{L}_0} \ln P(l_i | P_i; w) \quad (7)$$

And then we use the gradient method to get the optimal solution. The criterion for CRF parameter optimization in Eq. (7) is the maximum of the correct classification probability (the minimum of the likelihood function). However, the classification accuracy (i.e., the number of correctly classified vertices) is usually taken as the main criteria for evaluating the classification methods, while minimizing the likelihood function does not necessarily lead to the maximum of classification accuracy. Therefore, we propose a new parameter optimization criterion to maximize the classification accuracy  $\varepsilon(w)$ , called the parameter optimization criterion  $I$ , i.e.

$$w^* = \arg \max \varepsilon(w) \quad (8)$$

The discrete calculation of this parameter optimization criterion is in the following form:

$$\varepsilon(w) = \frac{1}{n} N(w) = \frac{1}{n} \sum_{j=1}^m \sum_{i=1}^n \Psi(l_j = l^*) \quad (9)$$

where

$$l^* = \arg \max \{P(l_1 | P_i; w), \dots, P(l_m | P_i; w)\} \quad (10)$$

and the explicit function  $\Psi$  is:

$$\Psi(l_i = l^*) = \begin{cases} 1, & l_i = l^* \\ 0, & l_i \neq l^* \end{cases} \quad (11)$$

Our criterion  $I$  in Eq. (8) is a discrete optimization problem that is not suitable to solve by the gradient method, while Monte Carlo (MC) Method [37] can be used for the parameter inference.

The problem of the parameter optimization criterion Eq. (7) that is commonly used in the literature, is that the classification accuracy is not guaranteed even when the sum of the probability of correct classification of the whole vertices is the maximum. Our parameter optimization criterion  $I$  is the discrete optimization criterion based on the optimal classification accuracy, and can obtain better classification result. However, this criteria  $I$  (Eq. (8)) may lead to relatively lower classification accuracy of the small scale vertices set for the case that the different classes with great disparity in number. In view of this defect, we propose another parameter optimization criterion that maximizes the accuracy with interclass weights, which is called the parameter optimization criterion  $II$ , i.e.,

$$w^* = \arg \max \varepsilon_2(w) \quad (12)$$

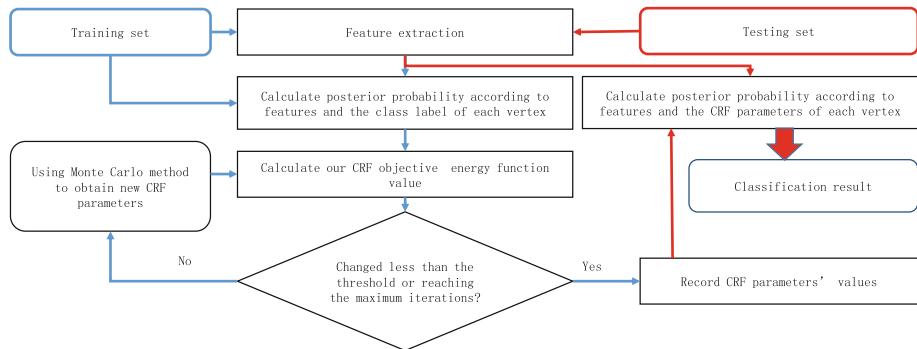
Here,  $\varepsilon_2(w)$  is called interclass weighting accuracy, and denoted as

$$\varepsilon_2(w) = \frac{1}{m} \sum_{j=1}^m \frac{1}{n_j} \left( \sum_{i=1}^n \Psi(l_j = l^*) \right) \quad (13)$$

where  $n_j$  is the number of vertices of the  $l_j$  class for the training set, and  $l^*$  is calculated according to the Eq. (10). Compared with our criterion  $I$  (Eq. (8)), our criterion  $II$  (Eq. (12)) pay more attention to the accuracies of different classes to balance the overall classification accuracy.

### 3.5 Flowchart of Our Algorithm

Figure 1 shows the flowchart of our algorithm, which mainly include two stages: the training stage and the testing stage. The target of the training stage is to calculate the CRF parameters of objective energy function, in order to using in the testing stage to acquire the classification result. On the training stage, we construct CRF from the input training set of point cloud, extract the features, and use one of our criteria to calculate the CRF parameters of the objective energy function iteratively. Stopping iterations has two conditions: (1) The changing of objective energy function value is less than the user-specified threshold; (2) The number of iterations reaches a maximum value that the user sets, as we have tested that when the iteration times is large enough, the improvement of optimized result tends to be stable and has little effect after each iteration. When the iteration is stopped, the corresponding optimized parameters' values are recorded, and if not, we use Monte Carlo method to get a new group of parameters for the next iteration. On the testing stage, we still firstly extract the features, and then use those CRF parameters obtained from the training stage to classify the testing set directly to generate the final classification result.



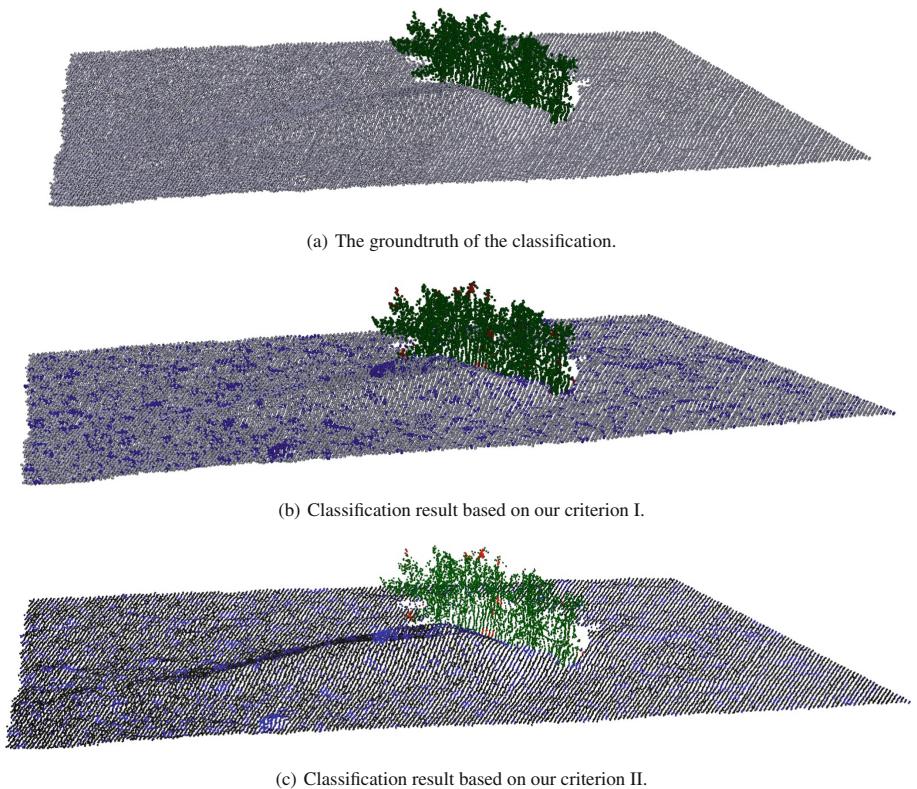
**Fig. 1.** Flowchart of our algorithm. The blue arrows guide the workflow of the training stage, while the red arrows guide the workflow of the testing stage. (Color figure online)

## 4 Experiments

All the experiments are done on a laptop computer, with an Intel Core i7-4710MQ CPU@2.50 GHz processor and 4.0 G memory. Our algorithm is implemented in C/C++ language with OpenGL to classify outdoor point cloud data. Usually outdoor point cloud data contain plants/vegetation information, so the main goal of our experiments is to distinguish the plants/vegetation area from the scene. We tested multiple point clouds sets with different sizes, different sources, and different complexities.

### 4.1 Classification of Terrestrial Lidar Scanning Point Cloud

Distinguishing the plants and the ground from the terrestrial LiDAR scanning point cloud is vital to environmental analysis and geographical investigation. A natural scene data opened by the French Nicolas Brodu point cloud classification work [38] is employed in our experiments, referred as *Canupo* point cloud data.



**Fig. 2.** Ground and vegetation classification result of *Canupo* point cloud. The data is from <http://nicolas.brodu.net/en/recherche/canupo/>.

The *Canupo* data is made up of the ground data and plants data, as shown in Fig. 2(a), where the ground point cloud includes 40069 vertices and the plants point cloud is made up of 4608 vertices. The white part in the scene is the missing point cloud caused by occlusion. Figure 2(b) and (c) are the results of our methods, which can separate ground and plants in general. In the figures, the blue vertices denote the ground vertices that are wrongly classified as plants, while the red vertices denote the plants vertices that are wrongly classified as the ground.

The total classification accuracy based on our criterion *I* is 87.3%, and the corresponding confusion matrix is shown in Table 1; the total classification accuracy based on our criterion *II* is 84.58%, and the corresponding confusion matrix is shown in Table 2. Both the classification accuracies are close to each other.

**Table 1.** The confusion matrix of our classification of *Canupo* point cloud based on our criterion *I*.

	Ground	Plants	Total
Ground	34598(86.35%)	5471(13.65%)	40069
Plants	203(4.41%)	4405(95.59%)	4608

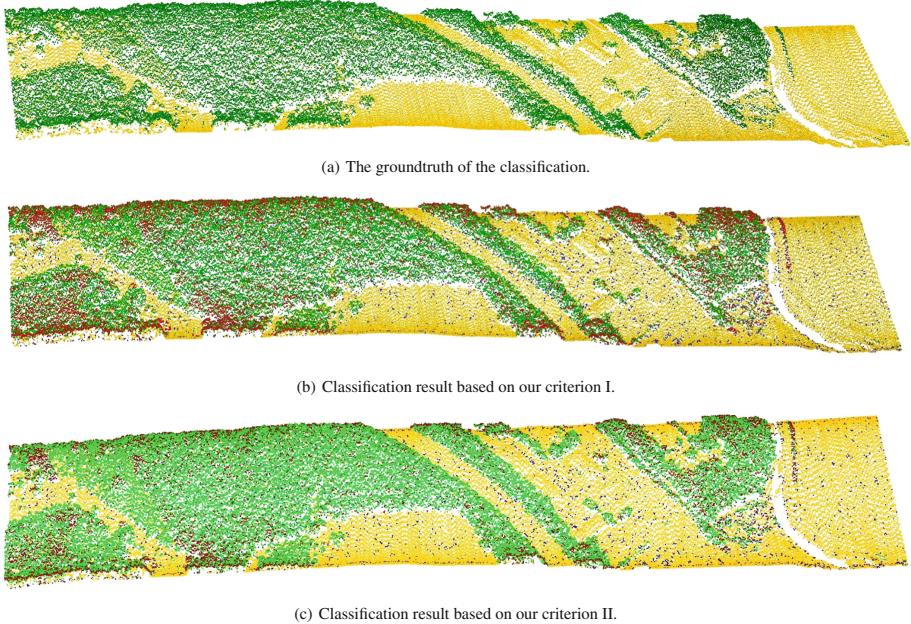
**Table 2.** The confusion matrix of our classification of Canupo point cloud based on our criterion *II*.

	Ground	Plants	Total
Ground	33293(83.09%)	6776(16.91%)	40069
Plants	114(02.47%)	4494(97.53%)	4608

## 4.2 Classification of Airborne LiDAR Scanning Point Cloud

Unlike terrestrial LiDAR scanner, the airborne LIDAR scanner has a larger scan range, and the captured point clouds are primarily on the top of the objects. In this section, an airborne scanning scene (abbreviated as *GML<sub>A</sub>* point cloud data) which comes from the Graphics and Media Lab (GML) at Lomonosov Moscow State University in Russia is used for our classification tests. The laboratory used non-joint Markov network method [2] for point cloud classification, accompanied by an accurate result of the point cloud classification. The *GML<sub>A</sub>* point cloud data contains 1,002,668 vertices, where the plants point cloud includes 539799 vertices and the remaining vertices number is 462869, as shown in Fig. 3(a).

For *GML<sub>A</sub>* point cloud data, the total classification accuracy based on our criterion *I* is 78.05% with the classification result in Fig. 3(b), and the corresponding confusion matrix is shown in Table 3; the total classification accuracy based on our criterion *II* is 78.58% with classification result in Fig. 3(c), and the corresponding confusion matrix is shown in Table 4. From our experiments, we can see that the classification accuracies based on our two criteria are almost same.



**Fig. 3.** Ground and vegetation classification result of  $GML_A$  point cloud. The data is from <http://graphics.cs.msu.ru/en/node/922>.

**Table 3.** The confusion matrix of our classification of  $GML_A$  point cloud based on our criterion *I*.

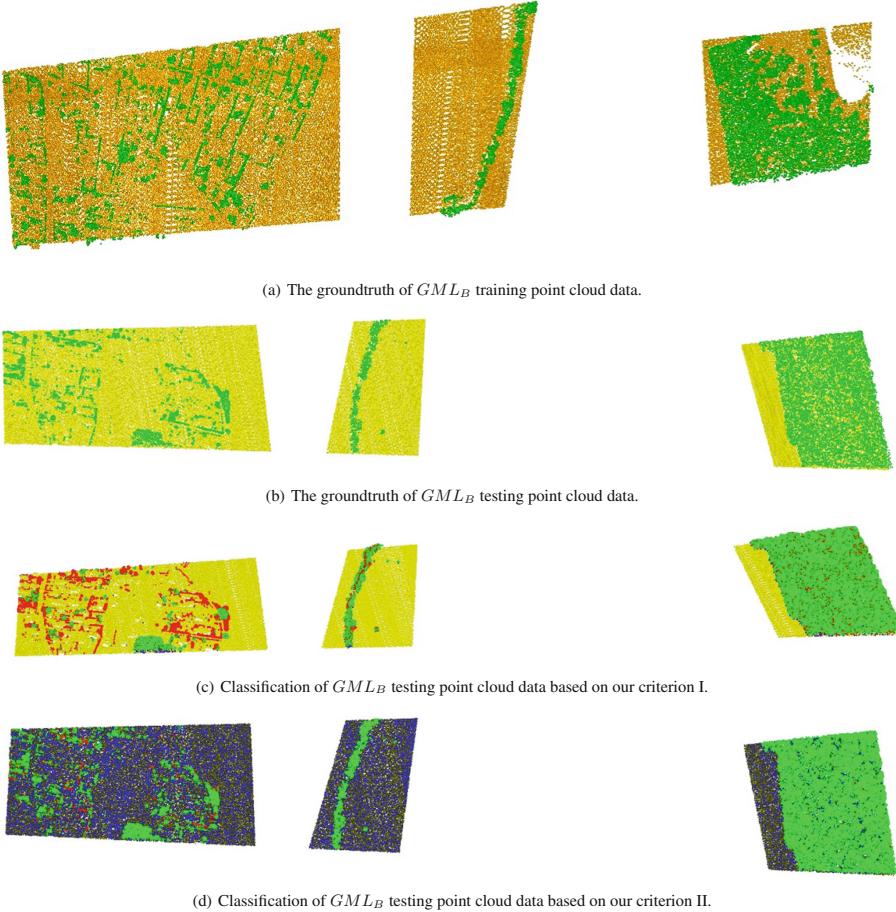
	Ground	Plants	Total
Ground	416068(89.89%)	46801(10.11%)	462869
Plants	173309(32.11%)	366490(67.89%)	539799

**Table 4.** The confusion matrix of our classification of  $GML_A$  point cloud based on our criterion *II*.

	Ground	Plants	Total
Ground	412396(89.10%)	50473(10.90%)	462869
Plants	164299(30.44%)	375500(69.56%)	539799

For the comparison on the testing set, we take another piece of point cloud data (abbreviated as  $GML_B$  point cloud data) from the Graphics and Media Lab (GML) at Lomonosov Moscow State University, and divide  $GML_B$  into  $GML_B$  training point cloud data and  $GML_B$  testing point cloud data. The  $GML_B$  training point cloud data includes 1544660 vertices, in which 1389385 vertices represent the ground, 155275 vertices are plants as shown in Fig. 4(a), and the  $GML_B$  testing point cloud data consists of 1160717 points, in which

1032766 vertices are the ground part accounting for 88.98%, and 127951 vertices are plants accounting for 11.02% as shown in Fig. 4(a). Please note that both the training point cloud data and the testing point cloud data include 3 blocks with 2 obvious blanks between them.



**Fig. 4.** Classification result of ground and plants in  $GML_B$  point cloud data. The data is from <http://graphics.cs.msu.ru/en/node/922>.

According to our criterion *I* (Eq. (8)), the classification result of  $GML_B$  testing point cloud data after parameters optimization is shown in Fig. 4(c) with the classification accuracy of 94.16%. Although the classification accuracy is relatively high, the classification for plants point cloud is very bad seen from the corresponding confusion matrix (Table 5): many vertices belonging to the plants are wrongly classified as the ground, which are shown as the red dots in Fig. 4(c).

**Table 5.** The confusion matrix of our classification of  $GML_B$  testing point cloud data based on our criterion  $I$ .

	Ground	Plants	Total
Ground	1015321(98.31%)	17445(1.69%)	1032766
Plants	50398(39.39%)	77553(60.61%)	127951

By using our criterion  $II$  (Eq. (12)) to evaluate the parameters for the  $GML_B$  testing point cloud data, the result is shown in Fig. 4(d). The total classification accuracy is 78.1%, which is lower compared to the result based on our criterion  $I$  (Eq. (8)). However, the classification accuracy for plants is increased from 60.61% to 94.41%, with the classification accuracy for the ground preserving at an acceptable rate 76.09% as shown in Table 6.

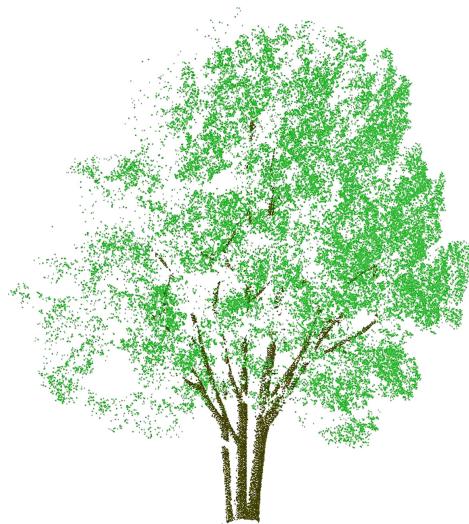
**Table 6.** The confusion matrix of our classification of  $GML_B$  testing point cloud data based on our criterion  $II$ .

	Ground	Plants	Total
Ground	785858(76.09%)	246908(23.91%)	1032766
Plants	7152(5.59%)	120799(94.41%)	127951

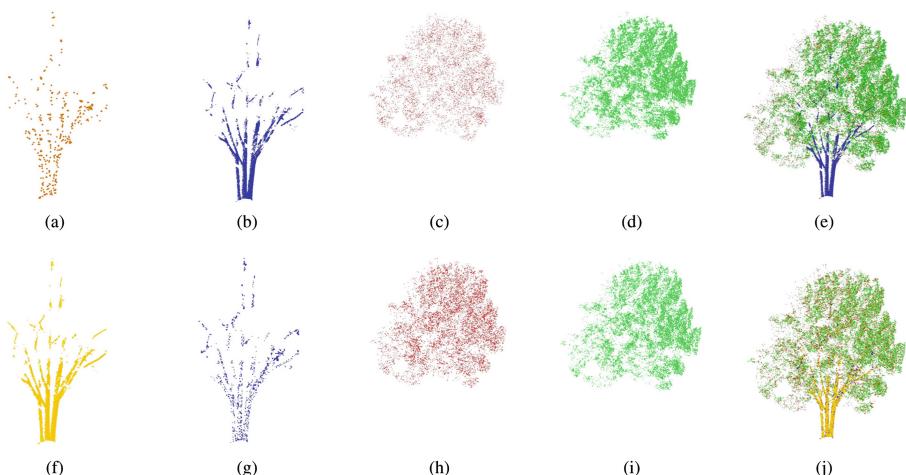
### 4.3 Classification for Branches and Leaves of Tree Point Cloud

Another kind of experiments is to distinguish the branches and leaves from the tree point cloud collected by ourselves. The tree point cloud is scanned from a pine tree, denoted as *Pine1* which consists of 487,555 points with 27,763(5.7%) branches vertices and 459,792(94.3%) leaf vertices as shown in Fig. 5. We choose part vertices from the *Pine1* for training to obtain CRF parameters.

The classification results are shown in Fig. 6. The top row is the classification result based on our criteria  $I$  (Eq. (8)) with the classification accuracy of 86.79%, and the bottom row is the classification result based on our criterion  $II$  (Eq. (12)) with the classification accuracy of 71.79%. By comparing the confusion matrix Tables 7 and 8 corresponding the classification results based on our two criteria respectively, we can see that the classification accuracy for the branches is much lower based on our criterion  $I$ . For the classification result based on our  $II$ , the classification accuracy for both branches and leaves are good with a slight loss on the overall classification accuracy compared to using our criterion  $I$ . This means the classification method based on the our criteria  $II$  can better balance classification the accuracy between the various classes and more stable to achieve a satisfied classification result.



**Fig. 5.** Point cloud *Pine1*.



**Fig. 6.** The classification results of point cloud *Pine1*. The top row (a–e) shows the classification results based on our criterion *I* (Eq. (8)), and the bottom row (f–j) shows the classification results based on our criterion *II* (Eq. (12)). From the left to right in each row, the subgraphs are correct branches point cloud, branches point cloud that wrongly classified as leaves, leaves point cloud that wrongly classified as branches, correct leaves point cloud, and the result combining the four previous subgraphs respectively.

**Table 7.** The confusion matrix of our classification for branches and leaves of *Pine1* based on our criterion *I* (Eq. (8)).

	Branches	Leaves	Total
Branches	424(1.53%)	27339(98.47%)	27763
Leaves	37066(8.06%)	422726(91.94%)	459792

**Table 8.** The confusion matrix of our classification for branches and leaves of *Pine1* based on our criterion *II* (Eq. (12)).

	Branches	Leaves	Total
Branches	25098(90.40%)	2665(9.60%)	27763
Leaves	134898(29.34%)	324894(70.66%)	459792

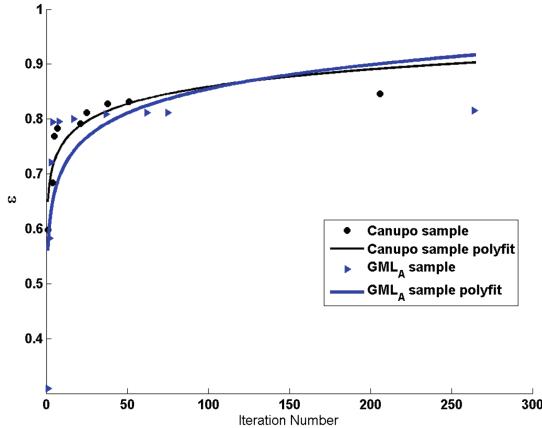
#### 4.4 Performance Analysis

**Computational complexity analysis.** In order to quickly and accurately extract the features based on  $k$  nearest neighbors of each vertex, we use the Kd-tree [39] for data organization, and calculate the normal vectors and the main curvature directions by the method [34]. The complexity of constructing Kd-tree is  $O(n)$ , and the complexity of the  $k$  nearest neighbors query is  $O(\log(n))$ , leading to the complexity of the normal vectors calculation and main curvature directions calculation are both  $O(n\log(n))$ . In the training stage, the iterations for parameter learning is related to the convergence rate of the Monte Carlo algorithm in the data classification, and each iteration is  $O(1)$ . In the testing stage, the complexity of the feature extraction is the same with the training stage, and the classification process is  $O(1)$  level, making the whole algorithm is linear logarithm  $O(n\log(n))$ .

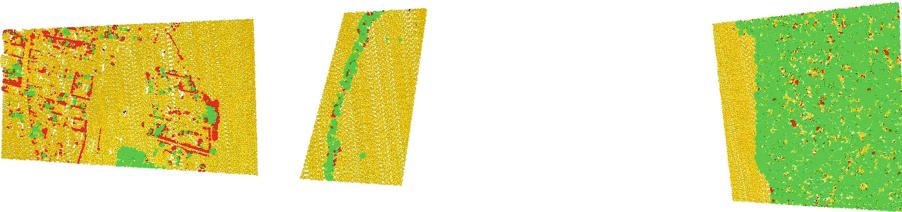
**The relationship between the iteration and the accuracy.** We uses the *Canupo* and *GML<sub>A</sub>* point clouds data to illustrate the relationship between the iteration and the accuracy. The size of these two poind clouds are quite different, but in the process of obtaining the optimal CRF parameters during the training stage, we find that both the numbers of iterations to achieve the best accuracy are close to each other. As shown in Fig. 7, when the number of iterations reaches 100, both the accuracies increasing rates become slow.

#### 4.5 Comparison with Other Works

One of the recent works in point cloud classification is based on the probability mixture method [33]. We test this probability mixing method to classify the *GML<sub>B</sub>* point cloud data, and the result is shown in Fig. 8 with he confusion matrix shown in Table 9. Comparing Fig. 8 to Fig. 4(c) and (d), we can see that all classification effects are very well. Calculated from Table 9, the total classification accuracy of [33] is 97.29%, which is slightly higher than both the



**Fig. 7.** The relationship between the iteration and the accuracy in the training stage.



**Fig. 8.**  $GML_B$  point cloud classification results based on [33].

classification accuracies based on our criterion *I* and criterion *II* (94.16% and 78.11% respectively). Compare Table 9 with Table 5, we can find that the classification accuracies for both the ground and plants based on probability mixture method [33] are slightly higher than the method based on our criterion *I*.

**Table 9.** The confusion matrix of  $GML_B$  point cloud classification results based on [33].

	Ground	Plants	Total
Ground	1032248(99.95%)	518(0.05%)	1032766
Plants	30957(24.19%)	96994(75.81%)	127951

However, comparing the confusion matrix Table 9 with Table 6, we can see that classification accuracy for plants on  $GML_B$  point cloud based on [33] is significantly inferior to the method based on our criterion *II*. The main reason is that the vertices number of plants is much smaller than the vertices number of the ground, and most of the methods such as [33] are based on the idea of maximizing

the total accuracy, which makes them a priority to the classification accuracy of larger subclass and ignore the classification accuracy of smaller subclass. Based on our criterion *II*), we can classify small scale class better in the point cloud with great disparity in number between different classes.

## 5 Discussion of Criteria Design

Our criterion *I* and criterion *II* are designed according to the idea of maximizing the classification accuracy and maximizing accuracy with interclass weights. In fact, we can also design other criteria for classification. For example, in order to classify the data point cloud into two classes, we can design the criterion of maximizing the first class classification accuracy (denoted as parameter optimization criterion *III*) or the criterion of maximizing the second class classification accuracy (denoted as parameter optimization criterion *IV*) as follows.

Parameter optimization criterion *III*:

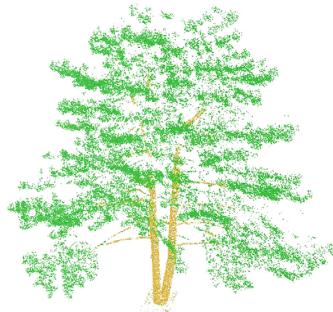
$$w^* = \arg \max \frac{1}{n_1} \left( \sum_{i=1}^n \Psi(l^* = 1) \right) \quad (14)$$

Parameter optimization criterion *IV*:

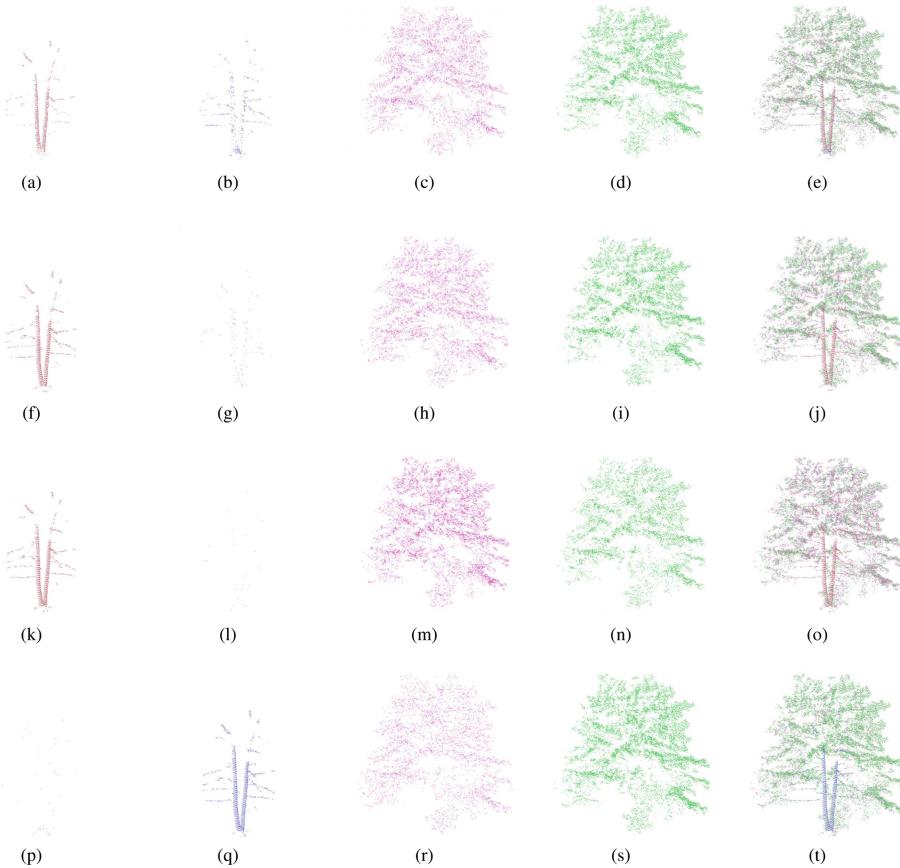
$$w^* = \arg \max \frac{1}{n_2} \left( \sum_{i=1}^n \Psi(l^* = 2) \right) \quad (15)$$

where the  $l^*$  is calculated as Eq. (10) both in criterion *III* and criterion *IV*.

We use two experiments to compare the classification effects of our four parameter optimization criteria. The first experiment uses another pine tree, denoted as *Pine2* as shown in Fig. 9. The scanned point cloud *Pine2* includes 269366 vertices, and the manual segmentation results are 20654 branches vertices and 248712 leaves vertices shown in Fig. 10. The corresponding classification accuracies for four criteria are shown in Table 10, where our first three criteria



**Fig. 9.** The manual classification result (groundtruth) of point cloud *Pine2*.



**Fig. 10.** The classification results of point cloud *Pine2* based on our four criteria, each row corresponding one criterion and from top to bottom are criterion *I*, *II*, *III*, and *IV* respectively. From the left to right in each row, the subgraphs are correct branches point cloud, branches point cloud that wrongly classified as leaves, leaves point cloud that wrongly classified as branches, correct leaves point cloud, and the result combining the four previous subgraphs in sequence.

can classify the branches better, and the criterion *IV* can classify leaves with best accuracy (79.61%).

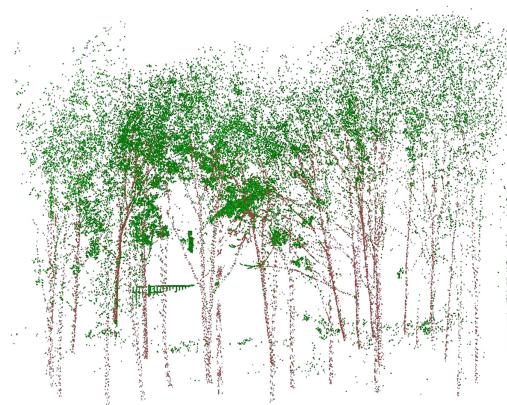
Another experiment is on the multiple pines point cloud, denoted as *Forest* with the scan data shown in Fig. 11. *Forest* consists of 556,308 vertices with manual segmentation of 151473 tree branches vertices and 414835 leaves vertices. The classification results based on our four criteria are shown in Fig. 12, and the corresponding classification accuracies are shown in Table 11, where the results are similar to the experiment on *Pine2*, that is, the first three criteria can extract the branches is better, and the criterion *IV* can classify the leaves with best accuracy (86.63%).

**Table 10.** The comparison of four confusion matrices corresponding the classification results of *Pine2* based on our four criteria.

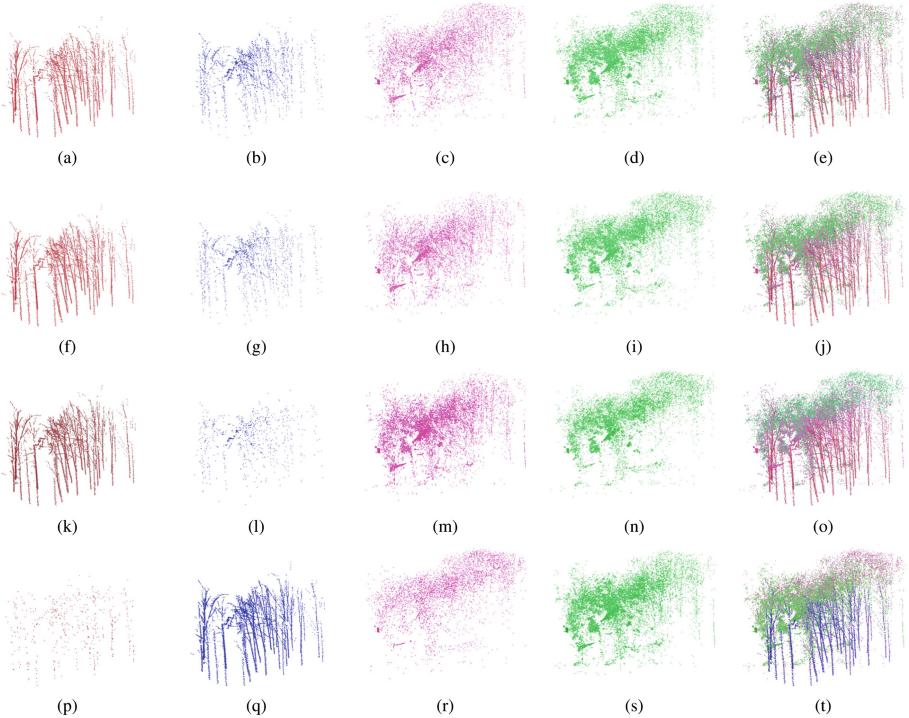
<i>Pine2</i>	Classify branches as branches	Classify branches as leaves	Classify leaves as branches	Classify leaves as leaves
Criterion <i>I</i>	66.77%	33.23%	33.23%	66.77%
Criterion <i>II</i>	93.20%	6.80%	34.37%	65.63%
Criterion <i>III</i>	98.55%	1.45%	51.16%	48.84%
Criterion <i>IV</i>	2.19%	97.81%	20.39%	79.61%

**Table 11.** The comparison of four confusion matrices corresponding the classification results of *Forest* based on our four criteria.

<i>Forest</i>	Classify branches as branches	Classify branches as leaves	Classify leaves as branches	Classify leaves as leaves
Criterion <i>I</i>	76.35%	23.65%	33.72%	66.28%
Criterion <i>II</i>	84.12%	15.88%	38.44%	61.56%
Criterion <i>III</i>	93.45%	6.55%	56.95%	43.05%
Criterion <i>IV</i>	4.07%	95.93%	13.37%	86.63%

**Fig. 11.** The manual classification result(groundtruth) of point cloud *Forest*.

From these two experiments, we can see that our criterion *I* and criterion *II* are comparable in the overall classification effect, while criterion *II* can balance the classification accuracies between different classes. Criterion *uppercaseiv* has the best effect on the separation of the leaves, but the classification of the branches is not very well; on the contrary, our criterion *III* can distinguish the branches better, but the classification of the leaves is less than



**Fig. 12.** The classification results of point cloud *Forest* based on our four criteria, each row corresponding one criterion and from top to bottom are criterion *I*, *II*, *III*, and *IV* respectively. From the left to right in each row, the subgraphs are correct branches point cloud, branches point cloud that wrongly classified as leaves, leaves point cloud that wrongly classified as branches, correct leaves point cloud, and the result combining the four previous subgraphs in sequence.

our criterion *I* and criterion *II*. When focusing on the extraction of a special class, criterion *III* and criterion *IV* can be taken as supplement control methods, and in common case, our criterion *I* and criterion *II* can satisfy our needs better.

## 6 Conclusion

In this paper, we have proposed new parameter optimization criteria in order to overcome the problem of current methods that maximizing the correct classification probability cannot guarantee the high classification accuracy. For the point cloud data, when the vertices numbers belong two classes of the point cloud data are not very different, our criterion *I* and criterion *II* can achieve better classification effects than previous methods; when different classes sets have great disparity in number, our criterion *II* can balance the accuracies between classes,

and make the recognition of the class with small vertices number achieve better classification accuracy. Our criterion *III* and criterion *IV* can be used as supplement control methods for better classification on some special classes. Since all the parameter optimization criteria are discrete, the Monte Carlo method can be used for the parameter optimization process. Our algorithm is simple with the convergence of the algorithm a little slow. The efficient solution of our discrete parameter optimization problem could be an important direction in the follow-up studies.

**Acknowledgements.** This work is partly supported by National Natural Science Foundation of China with Nos. 61331018, 61372190, 61501464, 61571439, and 61561003, partly supported by the National High Technology Research and Development Program (863 Program) of China with No. 2015AA016402 and Project 6140001010207.

## References

1. Zhang, X., Li, H., Dai, M., Ma, W.: Data-driven synthetic modeling of trees. *IEEE Trans. Vis. Comput. Graph.* **20**(9), 1214–1226 (2014)
2. Shapovalov, R., Velizhev, A., Barinova, O.: Non-associative markov networks for 3D point cloud classification. In: *Photogrammetric Computer Vision and Image Analysis* (2010)
3. Faugeras, O.: *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge (1993)
4. Jin, H., Wang, X., Zhong, Z., Hua, J.: Robust 3D face modeling and reconstruction from frontal and side images. *Comput. Aided Geom. Des.* **50**, 1–13 (2017)
5. Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guennebaud, G., Levine, J.A., Sharf, A., Silva, C.T.: A survey of surface reconstruction from point clouds. *Comput. Graph. Forum*, n/a–n/a (2016)
6. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: Pajdla, T., Matas, J. (eds.) *ECCV 2004*. LNCS, vol. 3023, pp. 224–237. Springer, Heidelberg (2004). doi:[10.1007/978-3-540-24672-5\\_18](https://doi.org/10.1007/978-3-540-24672-5_18)
7. Himmelsbach, M., Luettel, T., Wuensche, H.-J.: Real-time object classification in 3D point clouds using point feature histograms. In: *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 994–1000. IEEE Press, Piscataway (2009)
8. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3D registration. In: *IEEE International Conference on Robotics and Automation*, pp. 3212–3217 (2009)
9. Paulus, S., Dupuis, J., Mahlein, A.K., Kuhlmann, H.: Surface feature based classification of plant organs from 3D laserscanned point clouds for plant phenotyping. *BMC Bioinform.* **14**(1), 1–12 (2013)
10. Lalonde, J.F., Vandapel, N., Huber, D.F., Hebert, M.: Natural terrain classification using three-dimensional ladar data for ground robot mobility. *J. Field Robot.* **23**(10), 839–861 (2006)
11. Vosselman, G.: Point cloud segmentation for urban scene classification. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **XL-7/W2**(7), 257–262 (2013)

12. Weinmann, M., Jutzi, B., Hinz, S., Mallet, C.: Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.* **105**, 286–304 (2015)
13. Yang, J., Cao, Z., Zhang, Q.: A fast and robust local descriptor for 3D point cloud registration. *Inf. Sci.* **346–347**, 163–179 (2016)
14. Plaza-Leiva, V., Gomez-Ruiz, J.A., Mandon, A., Garc-Cerezo, A.: Voxel-based neighborhood for spatial shape pattern classification of Lidar point clouds with supervised learning. *Sensors* **17**(3), 594 (2017)
15. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the Eighteenth International Conference on Machine Learning, ICML 2001*, pp. 282–289. Morgan Kaufmann Publishers Inc., San Francisco (2001)
16. Kumar, S., Hebert, M.: Discriminative random fields. *Int. J. Comput. Vision* **68**(2), 179–201 (2006)
17. Munoz, D., Vandapel, N., Hebert, M.: Directional associative Markov network for 3-d point cloud classification. In: *International Symposium on 3-D Data Processing, Visualization, and Transmission* (2008)
18. Najafi, M., Taghavi Namin, S., Salzmann, M., Petersson, L.: Non-associative higher-order Markov networks for point cloud classification. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014. LNCS*, vol. 8693, pp. 500–515. Springer, Cham (2014). doi:[10.1007/978-3-319-10602-1\\_33](https://doi.org/10.1007/978-3-319-10602-1_33)
19. Rutzinger, M., Höfle, B., Hollaus, M., Pfeifer, N.: Object-based point cloud analysis of full-waveform airborne laser scanning data for urban vegetation classification. *Sensors* **8**(8), 4505–4528 (2008)
20. Niemeyer, J., Rottensteiner, F., Soergel, U.: Conditional random fields for lidar point cloud classification in complex urban areas. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **I–3(I–3)**, 263–268 (2012)
21. Husain, F., Dellen, L., Torras, C.: Recognizing point clouds using conditional random fields. In: *2014 22nd International Conference on Pattern Recognition*, pp. 4257–4262, August 2014
22. Wolf, D., Prankl, J., Vincze, M.: Fast semantic segmentation of 3D point clouds using a dense CRF with learned parameters. In: *IEEE International Conference on Robotics and Automation*, pp. 4867–4873 (2015)
23. Niemeyer, J., Rottensteiner, F., Soergel, U., Heipke, C.: Hierarchical higher order CRF for the classification of airborne lidar point clouds in urban areas. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **XLI-B3**, 655–662 (2016)
24. Lang, D., Friedmann, S., Paulus, D.: Adaptivity of conditional random field based outdoor point cloud classification. *Pattern Recognit. Image Anal.* **26**(2), 309–315 (2016)
25. Ni, H., Lin, X., Zhang, J.: Classification of ALS point cloud with improved point cloud segmentation and random forests. *Remote Sens.* **9**(3), 288 (2017)
26. Wang, Z., Zhang, L., Fang, T., Mathiopoulos, P.T.: A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification. *IEEE Trans. Geosci. Remote Sens.* **53**(5), 2409–2425 (2015)
27. Zhang, Z., Zhang, L., Tong, X., Takis Mathiopoulos, P.: A multilevel point-cluster-based discriminative feature for ALS point cloud classification. *IEEE Trans. Geosci. Remote Sens.* **54**(6), 3309–3321 (2016)
28. Rodriguezcuenca, B., Garcacorts, S., Ordóñez, C., Alonso, M.: Automatic detection and classification of pole-like objects in urban point cloud data using an anomaly detection algorithm. *Remote Sens.* **7**(10), 12680–12703 (2015)

29. Hu, X., Yuan, Y.: Deep-learning-based classification for DTM extraction from ALS point cloud. *Remote Sens.* **8**(9), 730 (2016)
30. Chen, J., Fang, Y., Yong, K.C., Kim, C.: Principal axes descriptor for automated construction-equipment classification from point clouds. *J. Comput. Civil Eng.* **31**(2), 1–36 (2017)
31. Kang, Z., Yang, J., Zhong, R.: A bayesian-network-based classification method integrating airborne lidar data with optical images. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **10**(4), 1601–1609 (2017)
32. Maligo, A., Lacroix, S.: Classification of outdoor 3D lidar data based on unsupervised gaussian mixture models. *IEEE Trans. Autom. Sci. Eng.* **14**(1), 5–16 (2017)
33. Li, H., Liu, X., Zhang, X., Yan, D.: A semi-automatic 3D point cloud classification method based on probability mixture of local shape features. *J. Zhejiang Univ. (Sci. Edn.)* **44**(1), 1–8 (2017)
34. Zhang, X., Li, H., Cheng, Z., Zhang, Y.: Robust curvature estimation and geometry analysis of 3D point cloud surfaces. *J. Inf. Computat. Sci.* **6**(5), 1983–1990 (2009)
35. Xiang, S., Nie, F., Zhang, C., Zhang, C.: Interactive visual object extraction based on belief propagation. In: Cham, T.-J., Cai, J., Dorai, C., Rajan, D., Chua, T.-S., Chia, L.-T. (eds.) *MMM 2007. LNCS*, vol. 4351, pp. 24–33. Springer, Heidelberg (2006). doi:[10.1007/978-3-540-69423-6\\_3](https://doi.org/10.1007/978-3-540-69423-6_3)
36. Sutton, C., McCallum, A.: An introduction to conditional random fields. *Found. Trends Mach. Learn.* **4**(4), 267–373 (2012)
37. Chen, Y., Raudenbush, S.W.: Maximum likelihood estimation in generalized linear: large mixed models using monte carlo methods: application to small-area estimation of breast cancer mortality. *Chin. J. Appl. Probab. Stat.* **22**(1), 69–80 (2006)
38. Brodu, N., Lague, D.: 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: applications in geomorphology. *ISPRS J. Photogramm. Remote Sens.* **68**(1), 121–134 (2012)
39. Mount, D.M.: Ann programming manual (2006). <http://www.cs.umd.edu/mount/ANN/Files/1.1>. Accessed 6 Apr 2012