




Article

Higher-Order Conditional Random Fields-Based 3D Semantic Labeling of Airborne Laser-Scanning Point Clouds

Yong Li ^{1,2}, Dong Chen ^{1,*} , Xiance Du ², Shaobo Xia ³ , Yuliang Wang ^{4,5} , Sheng Xu ⁶ and Qiang Yang ¹

¹ College of Civil Engineering, Nanjing Forestry University, Nanjing 210037, China; liyong@stumail.neu.edu.cn (Y.L.); qiangyang@njfu.edu.cn (Q.Y.)

² College of Information Science and Engineering, Northeastern University, Shenyang 110819, China; 1670954@stu.neu.edu.cn

³ Department of Geomatics Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada; shaobo.xia@ucalgary.ca

⁴ College of Forestry, Nanjing Forestry University, Nanjing 210037, China; ylw@chzu.edu.cn

⁵ School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China

⁶ College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China; xusheng@njfu.edu.cn

* Correspondence: chendong@njfu.edu.cn; Tel.: +86-185-5165-6801

Received: 8 April 2019; Accepted: 21 May 2019 ; Published: 27 May 2019



Abstract: This paper presents a novel framework to achieve 3D semantic labeling of objects (e.g., trees, buildings, and vehicles) from airborne laser-scanning point clouds. To this end, we propose a framework which consists of hierarchical clustering and higher-order conditional random fields (CRF) labeling. In the hierarchical clustering, the raw point clouds are over-segmented into a set of fine-grained clusters by integrating the point density clustering and the classic K-means clustering algorithm, followed by the proposed probability density clustering algorithm. Through this process, we not only obtain a more uniform size and more homogeneous clusters with semantic consistency, but the topological relationships of the cluster's neighborhood are implicitly maintained by turning the problem of topology maintenance into a clustering problem based on the proposed probability density clustering algorithm. Subsequently, the fine-grained clusters and their topological context are fed into the CRF labeling step, from which the fine-grained cluster's semantic labels are learned and determined by solving a multi-label energy minimization formulation, which simultaneously considers the unary, pairwise, and higher-order potentials. Our experiments of classifying urban and residential scenes demonstrate that the proposed approach reaches 88.5% and 86.1% of “mF₁” estimated by averaging all classes of the F₁-scores. We prove that the proposed method outperforms five other state-of-the-art methods. In addition, we demonstrate the effectiveness of the proposed energy terms by using an “ablation study” strategy.

Keywords: airborne laser-scanning; segmentation; cluster classification; neighborhood topology; higher-order potential; CRF optimization

1. Introduction

Outdoor scene labeling of airborne laser-scanning (ALS) point clouds is a key step in many applications such as autonomous driving, urban scene understanding, surveying and mapping, smart city and remote sensing, among others [1–6]. Several point cloud classification approaches have been proposed in the last decade. Those classification methods can be generally divided into

two categories, namely the single point-based method and the cluster-based method. Generally, the single point-based method consists of neighborhood selection, feature extraction and selection, and classification of each individual point cloud [7–10]. A series of publications on this topic have demonstrated the effectiveness of the approach. For example, Li et al. [7] proposed an ALS point cloud classification method based on multilevel features fusion and pyramid neighborhood optimization. Weinmann et al. [9] established a point cloud classification framework based on individual point clouds to obtain optimal features and achieve accurate classification results. The authors evaluated 7 types of domain definitions, 21 geometric feature definitions, 7 feature selection strategies, and 10 classifiers. Li et al. [11] proposed a single point-based point cloud classification method based on multiscale voxels features fusion derived by the deep learning network. However, in single point-based methods, the representation of the geometric structure and contextual information in individual point clouds has not been fully used, resulting in lower accuracy of point labeling. To solve this problem, cluster-based classification approaches [12,13] have been proposed by adding a segmentation step prior to classification. In this case, the basic processing units are not individual point clouds but the segmented clusters, from which the high-level discriminative features are derived and used for the final classification. For example, Xu et al. [13] proposed a cluster-based ALS point cloud classification method. This method constructs content-sensitive multilevel point clusters based on the density distribution of ground objects. The multilevel features of point clusters are constructed by sparse coding and latent Dirichlet allocation based on point-based features. These multilevel cluster's features are finally used for classification. The advantage of pre-segmentation in works [12,13] is that it is not affected by discontinuities and the poor quality of the input point clouds because the contiguous point clouds are grouped into larger entities that share similar properties or relationships. This results in homogeneous regions that are easier to interpret. Subsequently, the user-defined features that are derived from the segmented clusters are fed into machine learning or deep learning algorithms for final labeling.

The segmentation is the basis of many higher-level applications such as point cloud classification or object detection. The segmentation performance and quality can significantly influence the accuracy of point cloud labeling. To generate an ideal cluster result, the following properties are generally desirable: ① The point clouds within each cluster should be homogeneous and the created clusters should adhere well to the different object boundaries. ② The clusters should be fast to calculate, simple to use, and should improve the quality of point clouds labeling. ③ The clusters should have the function of topological self-organization, which means that the neighborhood relationships between the clusters should be known. However, it is difficult to choose effective measures to generate the clusters to meet the above-mentioned criteria, especially when segmenting complex scenes using scattered point clouds [14]. Currently, segmentation algorithms based on point clouds can be generally categorized into region-growing method, model-based method, and clustering method.

Region-growing method: This method aggregates the adjacent points into the same region based on similar attributes. More specifically, a seed point is selected from the point set and the surrounding neighborhood points are evaluated according to the predefined data structure. If the difference in the measure between the seed point and its neighbors satisfies the growth criterion, the neighborhood point is merged into the current point set. The new accepted point then becomes a new seed point and the neighboring points are searched again. This process is repeated until all points have been processed. Generally, the breadth-first search (BFS) and depth-first search (DFS) are commonly used strategies and common aggregation criteria include the angle, normal vector, curvature, Euclidean distance, texture, and RGB Value. For instance, Tovari and Pfeifer [15] proposed a segmentation method in which the K-nearest neighbor (KNN) of each point was first determined and then the normal vector of each point was calculated and was used as a criterion for the segmentation. Papon et al. [16] presented a voxel-based method for region-growing segmentation. This method first transforms the point cloud into voxels and selects initial seed regions (voxels). In the neighborhood of the seed region, RGB color information, simple geometric features, and the Euclidean distance are used as the criteria

for region-growing. This method ensures that the differences between the points in each cluster are smallest and that the differences between the clusters are maximized. Although this method is simple and effective, it is greatly influenced by the seed point selection and is constrained by the boundary conditions. Furthermore, since the growth criterion is based on the correlation between the points, the selection of low-level features has a large impact on the segmentation results.

Model-based method: In this method, the point cloud is segmented according to predetermined geometric models. Initial simple assumptions are made, followed by model verification. Subsequently, the points belonging to the same model are assigned to the same class. Among numerous hypothesis-based models, the Random Sample Consensus (RANSAC) algorithm [17] is most frequently used because it is a simple and powerful tool well suited for outlier detection, shape recognition, and registration. For example, Wang and Shi [18] first segmented a point cloud into subunits using a spatial grid. Then a local RANSAC algorithm was used to fit different models in the subunits, e.g., planes, cylinders, and spheres. Finally, the best model was determined using statistical inference. Ni et al. [19] assumed that the inliers were located close to each other and proposed the so-called GroupSAC algorithm that divides the input data by selecting samples from clusters containing many inliers. A detailed discussion regarding the standard RANSAC algorithm and its variants can be found in [20,21]. Rather than using RANSAC, Awadallah et al. [22] projected a point cloud into a two-dimensional grid and obtained a two-dimensional gray-level image based on the mesh density. Then the point cloud was segmented according to the gray-level image segmentation results using a Snake model. In summary, model-based methods are well suited for extracting objects that can be fitted by linear and/or nonlinear parameter estimation but are infeasible for free-form surface because of lack of corresponding nonlinear models. Moreover, the applications of these kinds of methods are limited by the classes of the model.

Clustering method: This method is an unsupervised learning method, which heuristically clusters points with similar attributes into the same class to meet the requirements of cost functions. Representative methods include the classic K-means algorithm [23], Euclidean distance clustering algorithm [24], mean shift clustering [25–27], hierarchical clustering [28–30], sample density-based clustering [31,32], and mixed kernel density function clustering [33]. For example, Wu et al. [24] introduced a smooth threshold constraint to the traditional Euclidean clustering algorithm to prevent over- and/or under-segmentation problems. The enhanced algorithm was used to segment a Kinect point cloud contaminated with abundant noise and many outliers. Yang et al. [4] used a K-means algorithm to solve segmentation problems in model construction for a reverse engineering project. More specifically, the curvature of each points was first calculated. The Euclidean metric among points was redefined and recalculated in a local rather than a global coordinate system by considering the curvatures and geometric coordinates of points. Based on this newly defined Euclidean metric, the point clouds are segmented by K-means clustering algorithm. The overall performance of the clustering method is satisfactory, but it is significantly affected by different types of distance metrics. In some complex scenes, the proximity of different objects reduces the accuracy of the clustering results. In addition, it is usually impossible to segment objects with different scales, which is a desirable goal because information at different scales provides significant clues to object detection and recognition.

After segmentation, the discriminative features are usually determined by fusing multiple features. Some commonly used features include the view feature histogram (VFH) [34], clustered view feature histogram (CVFH) [35], the length and width of the point set [36], the covariance features and shape features of the point set. These features are regarded as effective descriptors for cluster classification and recognition. These cluster-based features are fed into the classifier to finalize the labeling. From a technical point of view, the mainstream classifiers mainly include unsupervised classifiers, machine learning classifiers, probability and optimization-based classifiers, ensemble learning classifiers, and deep learning classifier [37]. Each kind of method has its own advantages and drawbacks. Especially for processing relatively small-sized point clouds, the probability and optimization-based classifiers, e.g., conditional random fields (CRF) and Markov random fields (MRF) tend to be more

promising than other methods because they result in more homogeneous clusters, and they consider the spatial correlation among multiple labels by incorporating contextual information and different types of prior knowledge into the algorithm, thereby improving greater flexibility and degrees of freedom for users. However, most existing MRF and/or CRF algorithms only consider the unary potential that is defined over a specific clique and pairwise potential that is defined over two cliques consisting of the two nearest clusters. In this case, the contextual information and prior knowledge have not been fully explored, thus leading to wrong labels in some complex scenes. To solve this problem, in this paper, we propose a higher-order CRF-based 3D semantic labeling framework. More specifically, the segmented framework simultaneously combines the existing classic algorithms, i.e., the K-means and the density-based spatial clustering of applications with noise (DBSCAN) [31] algorithms with the proposed probability density clustering algorithm to divide the raw ALS point clouds into a set of clusters. This preserves the clusters' homogeneity and neighborhood topology among clusters. It is to be noted that DBSCAN algorithm can be regarded as a kind of region-growing algorithms. After implementation of DBSCAN algorithm, the coarse-grained clusters are obtained. K-means and the proposed density clustering algorithms can be regarded as a kind of clustering methods. A set of fine-grained clusters is generated after implementation of K-means algorithm and the topological relationships between the fine-grained clusters are maintained by the proposed probability density clustering algorithm, i.e., the proposed segmented framework simultaneously combines the region-growing method with the clustering method, which commonly ensures the creation of homogeneous clusters with explicit topology. Based on these clusters and neighborhood topologies, we propose a higher-order CRF energy function that consists of unary, pairwise, and higher-order potentials defined over cliques consisting of more than two clusters to improve the accuracy of cluster labeling. The higher-order cliques are more useful for good perception of contextual information, thus improving the object classification and recognition. Given that our work is based on the conventional CRF framework, we explicitly state our original contributions as follows:

- **Hierarchical Point Cluster Generation:** We propose a multilevel clustering point set construction method. The first-level clustering uses a density clustering method to aggregate the points belonging to the same kind of object into a coarse cluster set. The second-level clustering uses the classic K-means algorithm to over-segment the coarse cluster set into a fine cluster set, from which the point set of the minimum processing unit is constructed. The third-level clustering uses the proposed probability density clustering method to construct a cluster set at a high-level scale, i.e., the third-level cluster set includes one-to-many relationships with the second-level cluster set. The multilevel clusters commonly provide greater use of contextual information, thus improving the accuracy of clustering labeling.
- **Cluster Topology Maintenance Strategy:** We present a strategy of constructing a neighborhood system among clusters by turning the problem of topology maintenance of clusters into a clustering problem based on the proposed probability density clustering method.
- **Higher-Order Energy Function:** We propose a higher-order CRF energy function that considers the constraints of the unary potential, the pairwise potential, and the higher-order potential defined over cliques consisting of more than two clusters.

This remainder of the paper is organized as follows. Section 2 describes the detailed methodology including data set descriptions, coarse-grained clustering, fine-grained clustering, topology maintenance, and CRF labeling optimization. In Section 3, the performance evaluation results of the labeling accuracy and effectiveness of the CRF cost function terms are presented, analyzed, and discussed. Finally, Section 4 concludes the paper along with a few suggestions for future research topics.

2. Methodology

To accurately assign semantic labels, e.g., trees, buildings, and vehicles to ALS point clouds, we propose a methodology that consists of four steps, which are outlined in Figure 1. The input

raw data (Figure 1a) are first segmented into a set of coarse-grained clusters using the classic DBSCAN algorithm, followed by over-segmentation using the K-means algorithm, thereby generating fine-grained clusters (Figure 1b,c). Subsequently, we finalize the topology maintenance of the fine-grained clusters by transforming the problem of topology maintenance into a clustering problem based on the proposed probability density clustering algorithm (Figure 1d,e). In the last step, three types of potentials, including the unary, pairwise, and higher-order potentials are embedded into the CRF model to finalize the classification. In the labeling framework, we use as input the ALS point clouds of residential and urban scenes, and generate point clouds with explicit semantic labels, i.e., trees, buildings, and vehicles (Figure 1f). It should be noted that the ground points are first filtered using a cloth simulation filtering algorithm [38] because we only focus on non-ground point clouds.

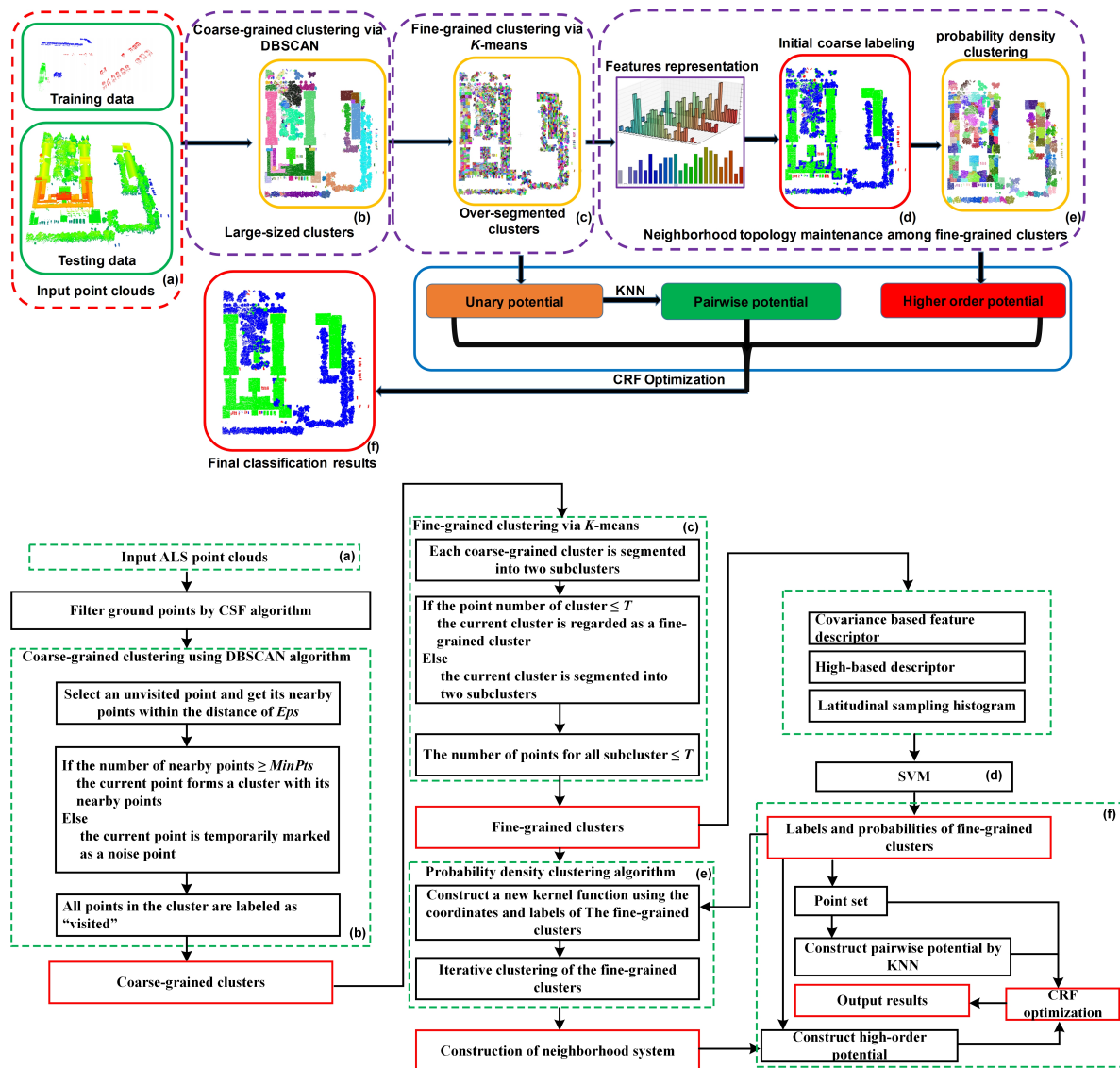


Figure 1. The flowchart of the proposed method.

2.1. Materials

To evaluate the performance of the proposed classification algorithm, we use scenes (Figure 2a,d) representing residential and urban areas in Tianjin, China to verify the accuracy of the proposed algorithm. The two scenes were acquired in August 2010 using a Leica ALS50 system with a mean flying height of 500 m above ground and a 45° field of view. The point density is approximately 20–30 points/m². Scene 1 contains large buildings that are adjacent to dense trees and some vehicles

scattered in this scene. Scene 2 is dominated by small-sized vehicles, which presents a challenge for the classification algorithm because of the unbalanced classes. Thus, these two scenes are appropriate for assessing the proposed algorithm. Because the proposed method is a supervised learning algorithm, we manually select the different classes in the point clouds as training data. To achieve this goal, registered aerial images and the commercial Terrasolid software package (<http://www.terrasolid.com/home.php>) are used to collect the training samples in two- and three-dimensional data space to improve the reliability of training data selection. In practice, we first use a stratified sampling scheme for the selection of the most representative point clouds. The stratified sampling guarantees that the selected point clouds are uniformly distributed and derived from different objects. Once the typical points are selected, visual inspection is conducted by combining the registered aerial images and interactive operations in two- and three-dimensional data space using Terrasolid software package. The remaining point clouds in the two scenes are regarded as testing points. The statistics of the training and testing data sets of the different classes in the two scenes are listed in Table 1.

Table 1. The statistics of the training and testing datasets in Scenes 1 and 2.

	Training Data			Testing Data		
	Tree	Building	Vehicle	Tree	Building	Vehicle
Scene 1	68,802	37,128	5380	213,990	200,549	7816
Scene 2	39,743	64,952	4584	73,207	156,186	7409

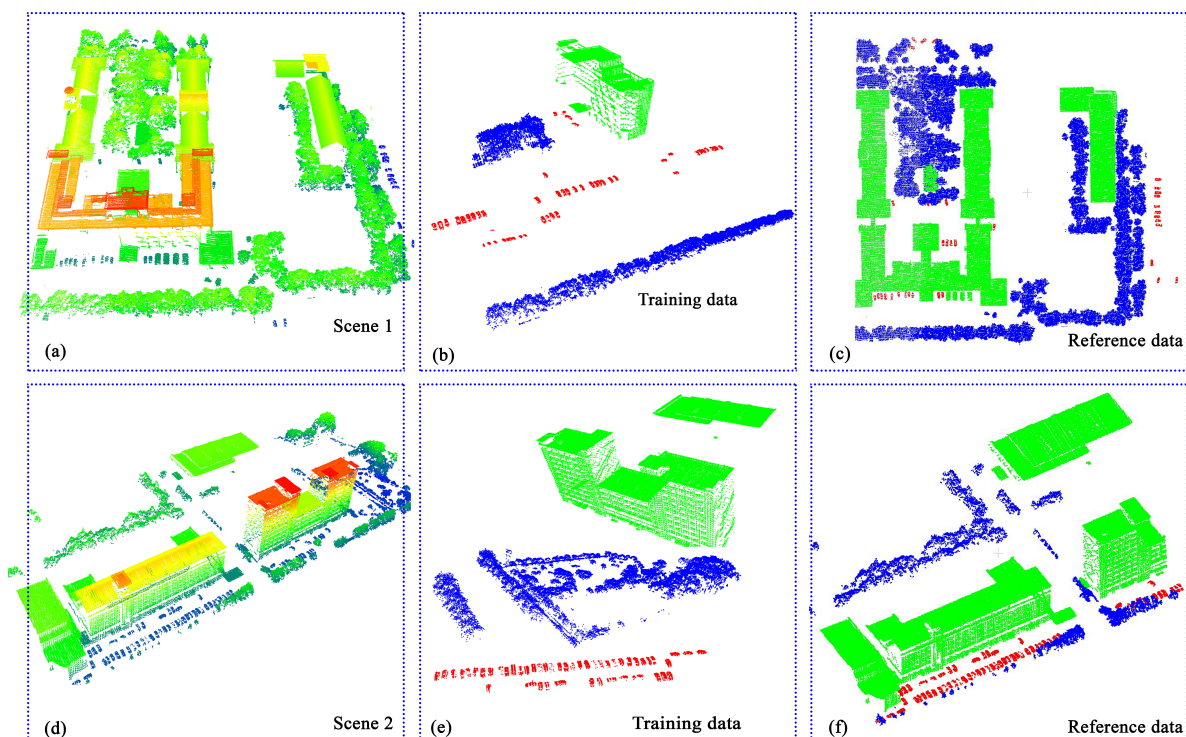


Figure 2. Two scenes for assessing the proposed algorithm. (a,d) are the residential and urban scenes in Tianjin. (b,e) are the selected points with semantic labels from (a,d) for training. The corresponding references are shown in subfigures (c,f). Please note that the point clouds in (a,d) are rendered according to point clouds' elevation, and other colors represent the semantic information, i.e., blue = trees, green = buildings and red = vehicles.

2.2. Coarse-Grained Clustering Using DBSCAN

Since the geometric shapes of scanned objects are complex, conventional clustering algorithms that are based on a rigidly fixed-point set/cluster are not suitable for complex scenes. To obtain

different numbers of clusters and different sizes of clusters corresponding to different objects, an adaptive clustering algorithm is needed. To this end, we implement the initial clustering based on the local point density and the connectivity of point clouds by using the classic DBSCAN algorithm. After being clustered, a complex scene can be simplified into a series of disposable clusters. This provides a solid foundation for the subsequent generation of a set of fine-grained clusters. The reason for choosing the DBSCAN algorithm is that it is resistant to noise and can find clusters with concave and convex shapes. In addition, the DBSCAN algorithm does not require the number of clusters as input. In fact, DBSCAN only requires two parameters: *Eps* and *MinPts*. The parameter *Eps* specifies how close the points should be to each other to be considered to be a part of the cluster, while the parameter *MinPts* specifies how many neighbors a point should have to be included into a cluster. After being clustered, the raw point clouds are clustered into three categories:

- **Core Point:** Any point p with several neighborhood points that is greater than or equal to *MinPts* is regarded as a core point.
- **Border Point:** For any point p , if the number of its neighbors is less than *MinPts*, it belongs to the ϵ -neighborhood of some core points.
- **Noise Point:** If a point is neither a core point nor a border point, then it is called a noise point or an outlier.

Based on these definitions, the core points with “density reachability” are clustered together to form one cluster. Here, “density reachability” is defined as follows: given a data set $P = \{p_1, p_2, \dots, p_n\}$, where $p = p_1$ and $q = p_n$. If a point from p_i ($i = 1, 2, \dots, n-1$) is within the ϵ -neighborhood of a point p_{i+1} , and the p_{i+1} is a core point, the point p_{i+1} ($i = 1, 2, \dots, n-1$) is defined as “density reachable” from a point p_i . If the set of connection points leading from p_i to p_{i+1} consists of only core points and is “density reachable”, the point q is defined as “density reachable” from point p . The border points are merged with the current cluster’s core points to form a class. The noise points are classified as another class.

As shown in the outdoor scene in Figure 3a, even though the point clouds are noisy and have outliers, we can roughly classify the point clouds of the same object into one cluster, as it is evident in Figure 3b. Please note that different colors represent different clusters after implementation of the DBSCAN algorithm. Compared with the labeling reference in Figure 3c, we observed that some trees are close to the house, resulting in under-segmented clusters. To solve this problem, the coarse-grained clusters generated by the DBSCAN algorithm are further refined by dividing them into smaller and fine-grained point sets to guarantee the homogeneity of points within each fine-grained cluster.

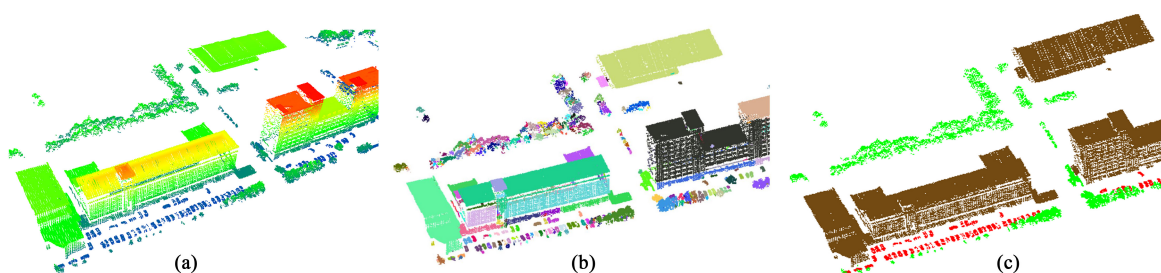


Figure 3. Coarse-grained cluster generation using the DBSCAN algorithm. (a) The raw point clouds. (b) The clustering results after using DBSCAN. (c) The reference data. Please note that in (a), the point clouds are colored by elevation. Different colors in (b) represent different clusters. A few colors have been reused, as a result, different disjoint clusters may share the same color. The semantic colors in (c) comply with the color code, i.e., green = trees, brown = buildings, and red = vehicles.

2.3. Fine-Grained Clustering Using K-Means

After coarse-grained clustering, the size of the segmented clusters is relatively large, and the results do not consider the trends of the point distribution. In addition, the coarse-grained clustering does not

provide homogeneous clusters, resulting in objects of multiple categories coexisting in each cluster. To ensure that the segmented clusters provide better discrimination, the clusters should be uniform size, and should contain only one single object or part of an object. To this end, the K-means algorithm is used to aggregate the points with uniform distribution into a series of subclusters. As previously mentioned in Section 1, three criteria are used for ideal segmentation. Fortunately, K-means algorithm meets the first criterion by tuning the number of clusters K . In addition, K-means algorithm is simple, efficient coverage, and easy implementation, which makes it compliance with the second criterion. However, the K-means algorithm do not comply with the third criterion, i.e., the generated clusters using K-means do not maintain the adjacent topological relationships, which are expected to be restored by the proposed probability density clustering algorithm (see Section 2.4). The direct use of the conventional K-means algorithm to segment the non-ground raw points is computationally inefficient and large-scale raw point clouds cannot be processed. In contrast, if we implement the K-means algorithm after each coarse-grained cluster has been created, the computational efficiency is improved significantly based on the concept of “divide and conquer”. Furthermore, by using the coarse-grained clusters as inputs of the K-means algorithm, multiple clusters can be processed in parallel, thereby further enhancing the computational efficiency. Therefore, the coarse-grained clusters derived from the DBSCAN algorithm are fed into the K-means algorithm to over-segment them into smaller cluster with relatively uniform size, which is below a predefined threshold \mathcal{T} . The detailed algorithm is given below:

The K-means algorithm is defined as follows: given a coarse-grained cluster \mathcal{C} with a set of point clouds $\mathcal{S}_c = \{p_i \in \mathcal{R}^3, (i = 1, 2, \dots, n)\}$, the number of predefined \mathcal{K} ($\mathcal{K} < n$) fine-grained clusters is obtained by minimizing the following energy function [23]:

$$\arg \min \sum_{c=1}^{\mathcal{K}} \sum_{p_i \in \mathcal{S}_c} \|p_i - p_c\|^2, \quad (1)$$

where \mathcal{S}_c represents the point set from cluster \mathcal{C} and p_c is the centroid set of \mathcal{S}_c . The detailed aggregation process is as follows:

- (1) Choose an arbitrary cluster \mathcal{S}_c from the coarse-grained cluster set as a processing unit and then randomly choose \mathcal{K} points as the beginning centroids, i.e., $p_i^{(0)} \in \mathcal{S}_c, i = 1, 2, \dots, \mathcal{K}$.
- (2) Assign the point clouds within \mathcal{S}_c to their associated centroids according to the criterion of the minimum Euclidean metric from a point to its associated centroid. After t iterations, we obtain:

$$\begin{cases} p_i \in \mathcal{S}_c^{(t)} \\ \|p_i - p_c^{(t)}\| = \min(\|p_i - p_m^{(t)}\|), \forall c, m = 1, 2, \dots, \mathcal{K}, \end{cases} \quad (2)$$

where $\min(\cdot)$ represents the minimum Euclidean metric between two points.

- (3) Update the centroid of each class:

$$p_c^{t+1} = \frac{1}{n_c} \sum_{j=1}^{n_c} p_j | p_j \in \mathcal{S}_c^{(t+1)}, \forall c = 1, 2, \dots, \mathcal{K}. \quad (3)$$

- (4) Repeat steps (2) to (3) until the centroid locations remain stable, i.e., $\mathcal{S}_c^{(t+1)} = \mathcal{S}_c^{(t)}, \forall c = 1, 2, \dots, \mathcal{K}$.

After the four steps, each coarse-grained cluster has been segmented into \mathcal{K} subclusters $\mathcal{S}_c, c = 1, 2, 3, \dots, \mathcal{K}$. The number of points $|\mathcal{S}_c|$ of each subcluster $\mathcal{S}_c, c = 1, 2, 3, \dots, \mathcal{K}$ is calculated. If it is less than or equal to the predefined threshold \mathcal{T} , the subcluster is remained, otherwise, we regard it as a new input to the K-means algorithm, i.e., we repeat steps (1) to (4) until the sizes of the segmented clusters are less than \mathcal{T} . This process is executed iteratively, until all coarse-grained clusters have been segmented. After the fine-grained clustering has been performed using the K-means algorithm, the

coarse-grained clusters shown in Figure 4a are segmented into the fine-grained clusters depicted in Figure 4b. It is observed that the clusters are more uniformly sized and have more homogeneous semantic consistency.

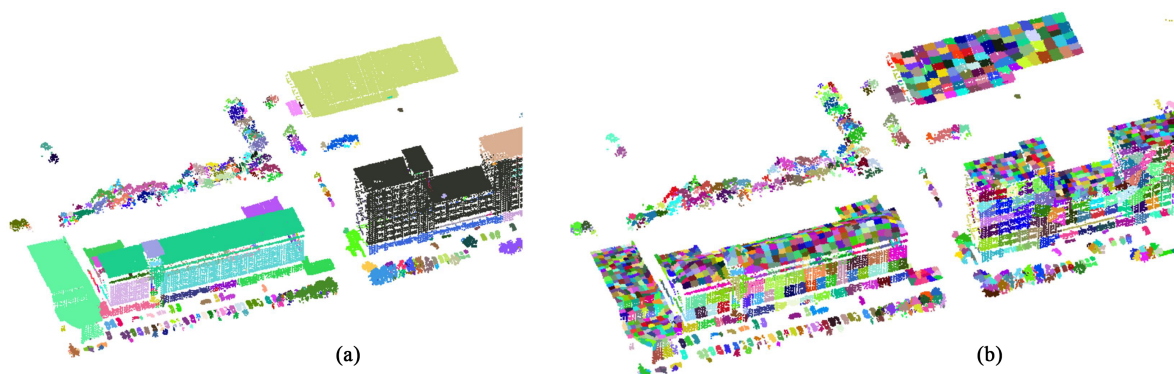


Figure 4. Fine-grained cluster generation using the K-means algorithm. (a) The coarse-grained clusters generated by DBSCAN. (b) The fine-grained clusters obtained by the K-means algorithm. Please note that different colors represent different clusters. A few colors have been reused, as a result, different disjoint clusters may share the same color.

2.4. Neighborhood Relationship Maintenance between Fine-Grained Clusters

Most methods that use neighborhood relationships (contextual information) for point labeling depend on information from the individual point clouds. This pointwise neighborhood relationship only represents the objects' local rather than the global geometric shapes. In addition, although the features extracted from the fine-grained clusters are more discriminative, the results as shown in Figure 4b do not include neighborhood relationships between the clusters. Intuitively, the adjacent topology between clusters is an important factor in object classification and recognition because it describes the objects' geometric shapes at a relatively high level. Although the KNN algorithm can be used to establish the neighborhood relationships between the fine-grained clusters, it only maintains a fixed number of neighboring clusters. This simple and rigid neighborhood topology may be effective for the description of relatively simple geometric shapes but not for complex scenes. To create a more effective cluster-based neighborhood system that is suitable for the subsequent CRF optimization described in Section 2.5, we propose a probability density clustering method that changes the problem of topology maintenance into a clustering problem between the fine-grained clusters. Because the proposed clustering algorithm requires initial coarse labels of the fine-grained clusters as constraints, we derive the initial coarse labels and execute the proposed clustering to obtain a new cluster set, from which the neighborhood relationship is obtained.

2.4.1. Initial Coarse Labeling of Fine-Grained Clusters

To obtain the initial coarse labels of the fine-grained clusters, we first create distinctive features for these clusters and then use a support vector machine (SVM) supervised classifier to train and label the clusters as trees, buildings, and vehicles. In [29,39], it was demonstrated that the point covariance matrix is an effective measure to describe the object's geometric shapes and point clouds' distribution; thus the covariance matrix and its derived high-level features are selected as feature descriptors. More specifically, we first extract the centroid of each cluster and estimate its covariance matrix. We can easily obtain the three eigenvalues, i.e., $\lambda_2 \geq \lambda_1 \geq \lambda_0$ and their corresponding eigenvectors, namely v_2 , v_1 , and v_0 . The features that are derived by linear or nonlinear combinations of these eigenvalues are regarded as covariance-based feature descriptors denoted as F_{cov} . For the ALS point clouds, the elevation is one of the discriminative features for object recognition, thus the high-based descriptor F_z is used. We use the feature F_{lsh} [7] for cluster labeling because it has good discriminative ability. Specifically, to calculate the feature F_{lsh} , we first select a point from a set of points as the center of a

sphere, and derive a histogram of the neighborhood points in the latitudinal direction. This feature is used to distinguish different classes of points according to the distribution of the neighborhood points in the latitudinal direction. This feature has the advantages of anti-occlusion, no influence of the local coordinate system, as well as high efficiency [7]. The features used in this study are listed in Table 2.

Table 2. The selected features for the initial coarse labeling of fine-grained clusters. Symbol “-” represents the specific descriptor that needs to be defined in each application.

	Features	Description
F_{cov}	$\lambda_0 + \lambda_1 + \lambda_2$	Sum of feature values
	$(\lambda_0 \cdot \lambda_1 \cdot \lambda_2)^{\frac{1}{3}}$	Full variance
	$-\sum_1^3 \lambda_i \ln(\lambda_i)$	The entropy of feature values
	$(\lambda_2 - \lambda_0) / \lambda_2$	Anisotropy property
	$(\lambda_1 - \lambda_0) / \lambda_2$	Planarity property
	$(\lambda_2 - \lambda_1) / \lambda_2$	Linear property
	λ_0 / λ_2	Scattered property
	$1 - < e_0 \cdot (0, 0, 1) >$	Horizontal property
F_z	-	Elevation characteristics
F_{lsh}	-	Latitudinal sampling histogram

For each cluster, we extract these features and concatenate them into high-dimensional feature vectors. The feature vectors are normalized and used in the SVM classifier with a Gaussian kernel to train the model. Once the model has been established, the probability of each class is obtained. The label of the class with the highest probability is the label of the current cluster. Please note that the initial labels are not required to be more precise, but they are needed as an approximate input to the subsequent clustering described in Figure 5. In addition, the initial coarse labeling provides the probability of fine-grained cluster of belonging to the different classes, which presents useful information for the subsequent CRF optimization described in Section 2.5.

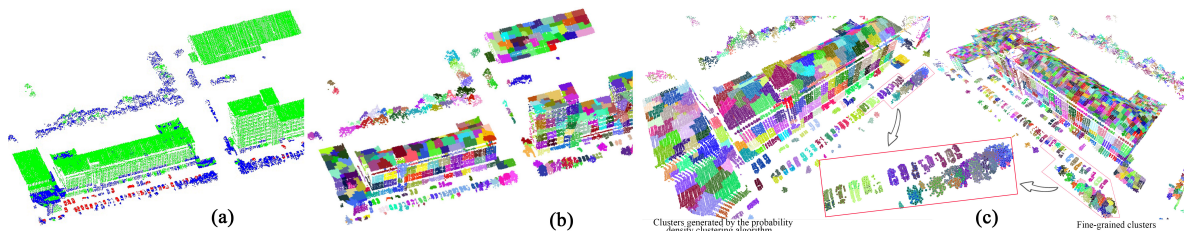


Figure 5. Neighborhood topology maintenance between the fine-grained clusters. (a) The initial coarse labels. (b) The results of the proposed probability density clustering. (c) The inclusive relationship between the clusters generated by the proposed probability density clustering and the fine-grained clusters. The inclusive relationships are clearly shown in the enlarged view in subfigure (c) by the overlap of the two kinds of clusters.

2.4.2. Creation of Neighborhood System

Once we assign the coarse labels to the fine-grained clusters, we use these labels as constraints to create the neighborhood system of fine-grained clusters. To this end, we transform the problem of developing a neighborhood system into a problem of clustering based on the fine-grained cluster set. More specifically, inspired by the concept of “probability density clustering” that is used in the classic mean shift algorithm, we use the initial coarse labels and centroids of the fine-grained clusters

as inputs. We assume that we have a centroid set $X = \{x_i, i = 1, 2, \dots, \mathcal{K}\}$ and we arbitrarily select x_i as the centroid of the first cluster. Then we update the probability density of the current cluster as follows:

$$M_h(x) = \frac{\sum_{x_i \in S_h} K(\frac{x_i - x}{h}) \delta(l_i - l) x_i}{\sum_{x_i \in S_h} K(\frac{x_i - x}{h}) \delta(l_i - l)} - x, \quad (4)$$

$$s.t. \quad \delta(l_i - l) \stackrel{def}{=} \begin{cases} 1, & \text{if } l = l_i, \\ 0, & \text{otherwise,} \end{cases}$$

where $K(\cdot)$ is a Gaussian kernel, the symbols l and l_i represent the corresponding initial coarse labels of the centroids x and x_i , and S_h represents the spherical areas with radius h and is defined by $S_h = \{y | (y - x)^T (y - x) \leq h^2\}$.

The entire clustering algorithm consists of the following seven steps:

- (1) Select an initial cluster center from the unlabeled set X , and set its class as the current cluster's label.
- (2) Label the centroids of the cluster in S_h . If they are consistent with the label of the current centroid, we update the accumulator.
- (3) Calculate the deviation vector using Equation (4) and update the center of the current cluster.
- (4) Repeat steps (2) to (3) until the mean deviation is less than a threshold, i.e., $\|M_h(x)\| < \epsilon$.
- (5) Determine whether the Euclidean metric from the current cluster's center to the existing center is less than σ . If it is true, these two clusters need to be merged, otherwise the current cluster is regarded as a new cluster.
- (6) Execute the steps from (1) to (5) until all the centroids of the clusters have been assigned a specific label.
- (7) For each centroid of a cluster, we assign the label with the highest frequency of visits as an associated label.

In summary, once the initial coarse labels are given (see Figure 5a), the proposed probability density clustering algorithm generates a new cluster set (see Figure 5b), each of which includes one or more fine-grained clusters, i.e., the neighborhood topology between the fine-grained clusters is implicitly embedded in the newly generated clusters. The inclusive relationships are shown in the enlarged view in Figure 5c by the overlap portion of the fine-grained and the newly generated clusters.

2.5. CRF Classification with Higher-Order Potentials

As described in Section 2.4.1, the point classification based on the features of the individual clusters is not stable and is subject to noise and outliers. In this section, we describe the establishment of the higher-order CRF model by integrating the label of the fine-grained clusters and the neighborhood topology between the fine-grained clusters to obtain the final class labels. The CRF model is a type of discriminative probabilistic model based on an undirected graph (UG). It is a modeling approach for determining the conditional probability of multiple variables given observation data [40]. This model not only considers the individual cluster's label but also the neighborhood relationship between the clusters. We use the initial labels of the clusters (Section 2.4.1) and the neighborhood topology (Section 2.4.2) to construct the high-level CRF model to finalize the classification of the fine-grained clusters.

The undirected graphical CRF model is a joint probability distribution model that satisfies the global Markov property. In this model, each vertex denoted by variable v represents one or a group of variables. The edge e connecting two vertices indicates the relational dependency between them. All vertices are stored in a set \mathcal{V} , and all edges are maintained in a set \mathcal{E} , and the UG is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. If two vertices in the subset of the UG have a connected edge, this subset is called a clique. If the clique does not exist exclusively within the vertex set of a larger clique, the subset of the vertex set constitutes a maximal clique, i.e., a maximum clique is a clique of the largest possible size in a

given UG. In our case, the vertices in the UG correspond to the fine-grained clusters and the edges correspond to the neighborhood topology between the fine-grained clusters.

We assume that a set $\mathcal{L} = \{l_1, l_2, \dots, l_n\}$ represents the initial labeling results of the fine-grained clusters and the set $\mathcal{Y} = \{y_1, y_2, \dots, y_n\}$ denotes the corresponding real labels of each cluster. The objective of CRF is the creation of conditional probability model $\mathcal{P}(\mathcal{Y}|\mathcal{L})$, which is defined as [41–43]:

$$\mathcal{P}(\mathcal{Y}|\mathcal{L}) = \frac{1}{\mathcal{Z}} e^{-\sum_{c \in \mathcal{C}} \varphi_c(y_c)}, \quad (5)$$

where \mathcal{Z} is a normalization factor that is used to normalize $\mathcal{P}(\mathcal{Y}|\mathcal{L})$ to a specific scale. $\varphi_c(y_c)$ represents a potential function of a group c belonging to a label y_c . Therefore, our objective is to obtain the optimal label configuration $\mathcal{Y} = \{y_1, y_2, \dots, y_n\}$ that maximizes the probability $\mathcal{P}(\mathcal{Y}|\mathcal{L})$. We use the logarithm of the negative value of Equation (5) and transform the maximization problem into a minimization problem [41–43]:

$$E(\mathcal{Y}) = -\log \mathcal{P}(\mathcal{Y}|\mathcal{L}) - \log \mathcal{Z} = \sum_{c \in \mathcal{C}} \varphi_c(y_c). \quad (6)$$

The labeling optimization problem is transformed into [41–43]:

$$E(\mathcal{Y}) = \arg \max_{\mathcal{Y}} \mathcal{P}(\mathcal{Y}|\mathcal{L}) = \arg \min_{\mathcal{Y}} E(\mathcal{Y}). \quad (7)$$

More specifically, our complete model is represented by three types of potentials, including the unary potential, pairwise potential, and higher-order potential:

$$E(\mathcal{Y}) = \sum_{i \in V} \varphi_i(y_i) + \sum_{(i,j) \in \mathcal{E}} \varphi_{ij}(y_i, y_j) + \sum_{|c| > 2} \varphi_c(y_c), \quad (8)$$

where the first two terms $\sum_{i \in V} \varphi_i(y_i)$ and $\sum_{(i,j) \in \mathcal{E}} \varphi_{ij}(y_i, y_j)$ are the unary and pairwise energies and the last term $\sum_{|c| > 2} \varphi_c(y_c)$ is the higher-order energy.

The unary term penalizes the discrepancy between the initial cluster and the solution labels and is defined as:

$$\varphi_i(y_i) = e^{1-\mathcal{P}(y_i)}, \quad (9)$$

where $\mathcal{P}(y_i)$ represents the point i 's probability belonging to class y_i . The unary potential is calculated using the initial labels and the probability of each fine-grained cluster belonging to different classes (see Section 2.4.1). Unlike with other feature descriptors, we directly use the initial probability of the cluster to create more accurate results and simplify the calculation.

The pairwise term is used to constraint the discrepancy between the nearest neighborhood clusters to ensure that the labels are identical. In other words, this interaction potential provides a definition of smoothness and penalizes changes between the current cluster and their nearest neighbors. The pairwise term is defined as follows:

$$\varphi_{ij}(y_i, y_j) = \begin{cases} \lambda_e e^{-dist}, & \text{if } y_i \neq y_j, \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

where λ_e is a balanced parameter and $dist$ represents the Euclidean distance between the centroids of two clusters.

Studies have shown that the potentials defined by using higher-order cliques are highly suitable [42,43]. Therefore, inspired by these works, in this study, we include the higher-order potential in the CRF model to improve the classification accuracy. Here the higher-order potential means that the potential defined over cliques consists of more than two vertices/nodes. The higher-order term

better describes the spatial context of 3D point clouds and it incorporates prior knowledge in the CRF model. In addition, it constrains the inconsistency among vertices and ensures that the labels are similar for higher-order cliques. To ensure smooth change of the higher-order term, we use the robust P_n Potts [41] model to design this energy term; it is defined as follows:

$$\varphi_c(y_c) = \begin{cases} \mathcal{N}_c^l(y_c) \frac{1}{Q} \gamma_{max}, & \text{if } \mathcal{N}_c^l(y_c) \leq Q, \\ \gamma_{max}, & \text{otherwise,} \end{cases} \quad (11)$$

where $\mathcal{N}_c^l(y_c)$ represents the number of vertices with label l in the cliques. Q is a truncation function threshold. It is used to determine the energy function of the higher-order term by comparing the number of clusters belonging to class c within a higher-order clique; if this number is small, the energy value of the higher-order term is high. γ_{max} is used to smooth the effect of each cluster in the higher-order clique.

To solve Equation (8), a primitive dual algorithm called SoSPD proposed in [44] is used to optimize the objective function. The algorithm uses a so-called “supplementary relaxation” rather than the conventional maximum flow to optimize the energy function. More precisely, this algorithm first performs linear programming relaxation for Equation (8) and executes a supplementary relaxation. During this process, these two steps must satisfy two conditions, i.e., low-order and higher-order relaxations, i.e., the original and dual solutions are similar initially to satisfy the low-order relaxation until it converges to simultaneously satisfy the low-order and higher-order relaxations.

3. Performance Evaluation and Discussion

In this section, we briefly introduce the hardware configurations and parameter settings of the workflow. Subsequently, we qualitatively evaluate the classification results by conducting a visual comparison with the reference data. In addition, we quantitatively evaluate the classification results using the selected measures, i.e., precision, recall, accuracy, F_1 -score, and mF_1 . We also compare our results with the results generated by the state-of-the-art methods and test the effectiveness of the proposed CRF model by using an “ablation study” strategy.

3.1. Implementation

The experiment is conducted using a personal computer equipped with a 4.2 GHz Intel Core i7-7700K CPU and 24-GB of main memory. The implementation of the proposed algorithm is performed using the open source Point Cloud Library, i.e., PCL1.8.0 (<http://pointclouds.org/>) embedded in the Microsoft Visual C++ integrated development environment. In addition, the relevant parameter setting is based on the characteristics of the objects in these two scenes, e.g., the geometric shapes and the ALS point density of the objects. To obtain the optimal input parameters, we use a trial and error strategy to determine the appropriate values. More specifically, in the DBSCAN clustering, we obtain good results when using the parameter Eps in the range $[0.7, 1.5]$ and the parameter $MinPts$ in the range $[6, 10]$. Similarly, in the over-segmented K-means algorithm and the procedure of topological maintenance using the proposed probability density clustering algorithm, we use the thresholds \mathcal{T} in the range $[200, 300]$ and the clustering radius h in the range $[0.8, 1.5]$, i.e., our algorithm is insensitive to changes in the parameter settings over a wide range of values. Based on the above-mentioned analysis, in our scenario, the optimal values $Eps = 1$ m, $MinPts = 8$, $\mathcal{T} = 300$, and $h = 1$ m are used as input to the workflow.

3.2. Comparisons

To demonstrate the superiority of the proposed algorithm, we conduct a qualitative comparison of our method with several state-of-the-art methods, namely Method 1 in [45], Method 2 in [46], Method 3 in [7], Method 4 in [47] and Method 5 in [48]. The characteristics of these methods are detailed in Table 3.

Table 3. Comparison with five methods in terms of “Basic Unit”, “Point Set Construction”, “Feature Expression”, “Optimization”, and “Classifier”.

Methods	Basic Unit	Point Set Construction	Feature Expression	Optimization	Classifier
Method 1	Individual point	None	Geometry, strength, and statistical characteristics	Region growth	JointBoost
Method 2	Individual point	None	Spin Image feature and F_{cov}	None	AdaBoost
Method 3	Individual point	None	Multiscale $F_z + F_{cov} + F_{plane}$	Multiscale neighbor	SVM
Method 4	Point set	Graph cut and linear transformation	LDA model based on Spin Image features and F_{cov}	None	AdaBoost
Method 5	Point set	Graph cut and exponential transformation	DD-SCLDA model based on Spin Image feature and F_{cov}	Discriminating inheritance	AdaBoost
Our Method	Point set	Multilevel clustering	$F_z + F_{cov} + F_{LSH}$	High-order CRF	SVM

The comparison results of Scene 1 and Scene 2 are shown in Figures 6 and 7. The results indicate that the other five methods result in numerous omission and commission error. More specifically, the building’s step edges and/or ridge edges are mistakenly classified as trees. Some parts of dense vegetation areas are incorrectly classified as buildings. Many small-sized vehicles have not been correctly and completely recognized. In contrast, thanks to the higher-order CRF optimization based on clusters, the point labels of the proposed method are more homogeneous, which makes it suitable for applications that require semantic class labels of point clouds. For example, the building points can be used for 3D building contouring and the tree points can be used for tree modeling and tree species identification. In Figure 7, we observe that this scene is dominated by small-sized vehicles and it is a challenge to segment these small objects from the imbalanced classes. Fortunately, it is evident that our proposed method is superior to the other methods, especially for recognizing the small-sized vehicles. Most of the vehicles in Scene 2 have been correctly labeled and recognized, except for some vehicles consisting of few points or being covered by dense vegetation. Although we obtain a good classification performance in the urban and residential scenes, the buildings in some areas denoted by the red rectangles are mistakenly classified as trees and some red ovals are falsely labeled as buildings. These misclassification errors occur because the features presented in Table 2 are straightforward and relatively simple, although we consider the labels of neighborhood clusters to assist in classification. However, once many neighborhood clusters have been mislabeled, the cluster itself suffers from misclassification errors.

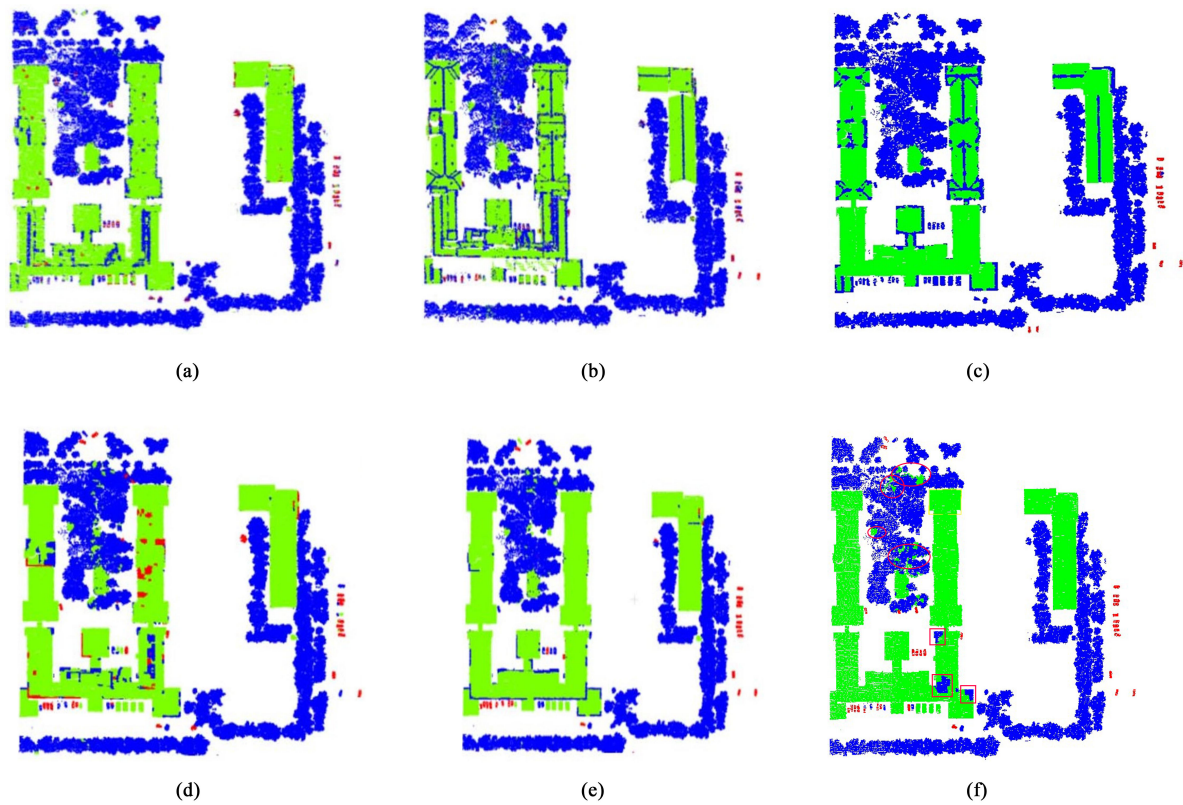


Figure 6. Comparison of the performance of six methods of classifying a residential scene. Subfigures (a–e) are the results generated by the state-of-the-art Methods 1 to 5. Our results are shown in (f). The color legend is defined in Figure 2.

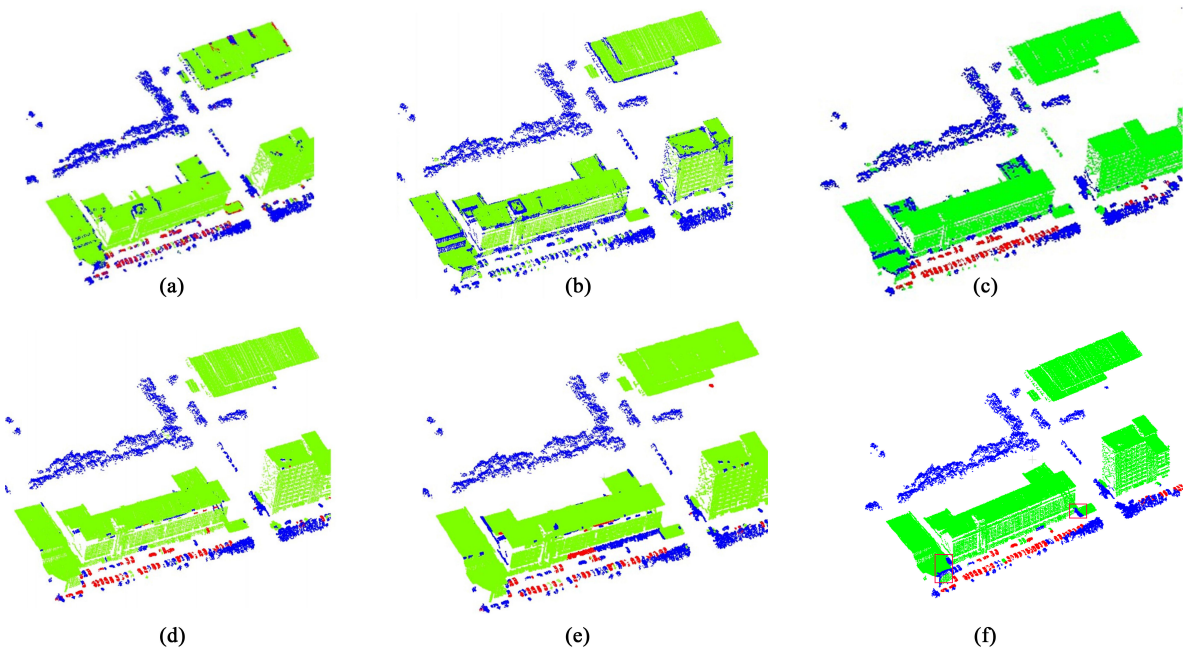


Figure 7. Comparison of the performance of six methods of classifying an urban scene. Subfigures (a–e) are the results generated by the state-of-the-art Methods 1 to 5. Our results are shown in (f). The color legend is defined in Figure 2.

We further use five measures, namely the precision, recall, accuracy, F_1 -score, and mF_1 to quantitatively evaluate the proposed method. Precision is the ratio of correctly predicted positive

observations to the total predicted positive observations, i.e., $\text{precision} = TP / (TP + FP)$ [49]. Recall is the ratio of correctly predicted positive observations to all observations in the positive class i.e., $\text{recall} = TP / (TP + FN)$ [49]. Accuracy is the most intuitive performance measure and is the ratio of correctly predicted observations to the total observations, i.e., $\text{accuracy} = (TP + TN) / (TP + FP + TN + FN)$ [50]. F_1 -score is the weighted average of the precision and recall and is defined as: $F_1\text{-score} = 2 \times (\text{recall} \times \text{precision}) / (\text{recall} + \text{precision})$ [13]. Therefore, this score takes both false positives and false negatives into account. It is more useful than the other three measures, especially if the scene has an uneven class distribution.

In Table 4, we provide the statistics of the four measures. Although our method does not have the highest precision or recall for all classes, we achieve a good tradeoff between precision and recall and the overall accuracies are 95.4% and 95.2% for Scene 1 and Scene 2, respectively. Since the percentage of the training vehicle points is relatively low (see Table 1), the accurate labeling of vehicles has become more challenging, i.e., we confront class imbalanced problem in Scene 1 and Scene 2. However, in this case, we still obtain the highest F_1 -score values of 73.7% and 66.9% for vehicle labeling in Scene 1 and Scene 2. This proves that our method has an advantage in terms of classification of small vehicles over other state-of-the-art methods. The F_1 -score values for trees and buildings in Scene 1 are 95.9% and 95.8% and are the highest values among Methods 1–5, thereby obtaining the highest mF_1 value. Please note that the mF_1 measure is calculated by averaging all classes of the F_1 -scores in the scene. It is a comprehensive indicator to represent the overall performance of scene labeling. Similarly, we obtain the highest mF_1 value of 86.1% for Scene 2 and the F_1 -scores for trees, buildings, and vehicles are 93.4%, 98.0%, and 66.9%. Although 93.4% for the F_1 -score for the tree class is not the highest value, we achieve a good balance among all classes and the highest mF_1 value of 86.1% for Scene 2. It should be noted that although numerous small-sized vehicles are scattered in Scene 2, we still obtain 95.2% accuracy, which is slightly smaller than 95.5% of Method 5 because some building point clouds are falsely classified as vegetation. Although our method for classifying buildings and vehicles outperforms Method 5, the overall number of point clouds with corrected semantic labels is lower than that of Method 5. In fact, the number of tree points in Scene 2 is ten times that of the number of vehicle points; therefore, many misclassified tree points greatly affect the accuracy of the proposed method. This also explains why the F_1 -score of 93.4% for the tree class is lower than that of Method 3 (95.6%) and Method 4 (94.6%). However, it should be noted that a high accuracy does not mean that Method 5 has a good classification performance, i.e., the classification algorithm should consider the performance of all classes. Therefore, some measures such as F_1 -score or mF_1 are recommended when evaluating the overall performance of classification algorithms.

Table 4. Quantitative comparison of six classification methods using measures of “precision/recall”, “accuracy”, “ F_1 -score”, and “ mF_1 ”. The values from columns of tree, building, and vehicle correspond to “precision/recall”. The measure of “accuracy” means the overall performance of three classes. The values of “ F_1 -score” correspond to tree, building, and vehicle. The last column represents the value of “ mF_1 ” measure, which is calculated by averaging the three classes.

	Methods	Tree (%)	Building (%)	Vehicle (%)	Accuracy (%)	F_1 -Score (%)	mF_1 (%)
Scene 1	Method 1	85.7/92.9	92.0/83.8	56.9/54.7	87.9	89.2/87.7/55.8	77.5
	Method 2	89.7/98.1	97.9/89.1	65.2/46.6	92.9	93.7/93.3/54.4	80.5
	Method 3	99.2/84.9	86.8/ 99.3	99.9/42.7	91.9	91.5/92.7/59.8	81.3
	Method 4	94.8/93.8	93.5/92.3	41.2/66.7	92.6	94.3/92.9/50.9	79.4
	Method 5	93.1/96.0	95.2/92.6	73.3/62.2	93.7	94.5/93.9/67.3	85.2
	Our Method	95.5/ 96.4	96.1/95.4	76.7/70.9	95.4	95.9/95.8/73.7	88.5
	Methods	Tree (%)	Building (%)	Vehicle (%)	Accuracy (%)	F_1 -Score (%)	mF_1 (%)
Scene 2	Method 1	73.9/91.2	93.6/88.2	29.5/25.4	87.2	81.6/90.8/27.3	66.6
	Method 2	86.8/91.2	96.8/95.5	44.1/34.8	92.2	88.9/96.1/38.9	74.7
	Method 3	83.2/92.9	98.5/92.8	62.6/ 65.7	92.0	95.6/87.8/64.1	82.5
	Method 4	90.3/93.9	97.6/96.5	49.4/42.0	94.1	92.1/97.0/45.4	78.2
	Method 5	94.7/94.5	98.1/97.7	53.9/60.5	95.5	94.6/97.9/57.0	83.2
	Our Method	92.3/ 94.5	98.2/ 97.8	71.5/62.8	95.2	93.4/ 98.0/66.9	86.1

3.3. Effectiveness of CRF Model

To demonstrate the superiority of the higher-order CRF classification model, we use an “ablation study” to compare three different configurations including classification without CRF optimization, a low-level CRF model using only first- and second-order terms, and a high-level CRF model using first- and second-order terms, as well as higher-order terms. In these three cases, we use the F_1 -score and mF_1 measures to evaluate the classification accuracy of the two scenes. The results are shown in Table 5. It is evident that without CRF optimization, we obtain the smallest F_1 -score and mF_1 score for all classes in the two scenes, indicating the effectiveness of the CRF optimization. If we use the low-level CRF model in the workflow, it is expected that the results are superior to the methods without using CRF optimization, but the improvement is not large. For the high-level CRF model, we obtain mF_1 score of 88.5% and 86.1% for Scene 1 and Scene 2, indicating that this method outperforms the other two methods.

Table 5. Classification performance of methods (F_1 -score and mF_1) with three configurations.

	Configurations	Tree (%)	Building (%)	Vehicle (%)	mF_1 (%)
Scene 1	Without CRF	92.5	90.5	67.9	83.7
	Low-level CRF	92.7	93.6	68.9	85.1
	High-level CRF	95.9	95.8	73.7	88.5
	Configurations	Tree (%)	Building (%)	Vehicle (%)	mF_1 (%)
Scene 2	Without CRF	85.9	94.7	61.1	80.6
	Low-level CRF	90.2	96.1	64.2	83.5
	High-level CRF	93.4	98.0	66.9	86.1

3.4. Discussions

From the above-mentioned qualitative and quantitative analyses, we can conclude that the cluster-based classification algorithms, i.e., Methods 4 and 5 and the proposed method are more accurate than the point-based methods because the former methods include more contextual information and the estimated features are more stable and robust. Our results in Figures 6 and 7 are more homogeneous than those of the other methods and there are few scattered points of other classes within the boundary of a specific class. The class boundaries are better defined. Although Method 3 resulted in relatively high precision and/or recall, the completeness of a specific object

cannot be guaranteed. Although the cluster-based Method 5 provides good point labeling, it includes some obvious error and the results are not as homogeneous as the proposed method. In contrast, our method results in more accurate point labels and complete geometry, which is important for detailed applications of semantic point clouds, such as vectorization of buildings and reconstruction of buildings.

The good performance of the proposed method is attributed to the following three reasons:

- (a) **Self-adaptive Segmentation:** In outdoor scenes, the objects have various shapes and sizes. The simple segmentation according to rigid number and the size of the clusters cannot adapt to various objects. In this case, the self-adaptive segmentation algorithm is required. The proposed workflow is adaptation-aware and is shown in two aspects: ① The coarse-grained clustering using the DBSCAN can identify any shapes of clusters according to the distribution of point clouds. The implementation cannot require the number of clusters and can easily find the noise and outliers. ② We propose probability density clustering algorithm to ensure topological maintenance between adjacent fine-grained clusters. The adaptive characteristics is represented by aggregating different number of fine-grained clusters into a high-level cluster, from which one-to-many relationships are included. That is the number of fine-grained clusters in a generated cluster is not fixed and is determined by the shapes and local properties of the objects. Self-adaptive segmentation ensures reasonableness of generated clusters at different levels.
- (b) **Multilevel Clustering:** The feature of cluster is more robust and stable compared to individual point clouds. Therefore, we propose three-level cluster generation strategy: the first level creates coarse-level clusters using DBSCAN algorithm. The second level clustering uses conventional k-means algorithm to over-segment the coarse-level clusters into a set of fine-grained clusters. The proposed probability density clustering algorithm works at the third level; it aggregates the fine-grained cluster set into a high-level scale, i.e., the third-level cluster set includes one-to-many relationships with the second-level clusters. Multilevel clustering can better explore the contextual information between the objects in outdoor scenes. Although there exists a multilevel cluster construction algorithm in [13], the size of generated clusters is linearly increased, causing no apparent feature discrepancy of the objects.
- (c) **Higher-Order CRF Optimization:** To optimize the point labeling using the proposed higher-order CRF model, we not only consider the adjacent clusters but a wider local area based on the neighborhood relationship between the clusters. More precisely, we design the higher-order potential and embed it into the CRF optimization. The higher-order potential can fully perceive the prior knowledge and contextual clues, thereby improving the accuracy of classification and recognition.

Through the above analysis, we believe that the three strategies, namely self-adaptive segmentation, multilevel clustering, and higher-order CRF optimization guarantee that the proposed method can theoretically result in a better outcome compared to other point-based and/or cluster-based methods listed in Table 3.

4. Conclusions

In this paper, we have presented a novel CRF-based 3D semantic labeling algorithm for assigning semantic information in ALS point clouds. Instead of using point-based semantic labeling, this algorithm is based on higher-level clusters that are created by integrating the DBSCAN and the classic K-means algorithms, followed by the proposed probability density clustering algorithm. Using the over-segmented clusters as the basic processing units has two advantages: ① The computational efficiency is significantly improved during the CRF optimization. ② All individual points and clusters have explicit contextual information, which facilitates the detection of the object by the classifiers. To determine the neighborhood topology between the generated clusters, we transform the problem of

topological maintenance into a clustering problem using the proposed probability density clustering algorithm. This algorithm maintains a flexible neighborhood system in a natural intuitive manner and does not require the rigid and fixed numbers of a neighborhood system used in conventional methods. Finally, these cluster entities and their contextual information are used for labeling the points by incorporating unary, pairwise, and higher-order potentials established on different levels of cliques, thereby achieving better point cloud labeling. Compared with five other state-of-the-art methods, the classification of urban and residential scenes with the proposed method has better classification accuracy and there is a good tradeoff between precision and recall; the mF_1 measures are 88.5% and 86.1% for Scene 1 and Scene 2, respectively. In addition, it is found that the CRF optimization with higher-level potentials defined over cliques consisting of more than two clusters is effective and the highest labeling performance is obtained in terms of the F_1 -score and mF_1 .

Although our algorithm achieves promising classification accuracy for the Tianjin data set, there are some drawbacks and interesting ideas which can be further explored to extend the research reported in this paper. We use the initial labels to design the CRF' unary term. These initial labels are obtained by using the cluster features defined in Table 2. In some cases, the low-level features of the clusters are not stable. In future works, we plan to use a deep learning method to extract the high-level cluster features for better discrimination of the cluster's properties. In addition, if we use higher-order potential terms in the CRF optimization, a large number of iterations is required, and the computational cost will increase along with an increase in the cluster number; this makes the processing of large-scale ALS point clouds impossible. For this problem, we plan to explore the in-depth properties and topologies of different levels of clusters to control the number of participating cluster entities.

Author Contributions: Y.L. and X.D. analyzed the data and wrote the C++ source code. D.C. and S.X. (Shaobo Xia) helped with the project and study design, paper writing, and analysis of the results. Y.W., S.X. (Sheng Xu), and Q.Y. helped with the data analysis, experimental analysis, and comparisons.

Funding: This work was supported in part by State Key Laboratory of Geo-Information Engineering under Grant SKLGIIE2018-K-3-1, and in part by the National Natural Science Foundation of China under Grant 031050004.

Acknowledgments: The authors would like to thank Zhenxin Zhang for providing the Tianjin data set for comparisons.

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

References

1. Yang, B.; Dong, Z. A shape-based segmentation method for mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* **2013**, *81*, 19–30. [[CrossRef](#)]
2. Yang, B.; Dong, Z.; Zhao, G.; Dai, W. Hierarchical extraction of urban objects from mobile laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2015**, *99*, 45–57. [[CrossRef](#)]
3. Landrieu, L.; Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4558–4567.
4. Yang, B.; Dong, Z.; Liu, Y.; Liang, F.; Wang, Y. Computing multiple aggregation levels and contextual features for road facilities recognition using mobile laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2017**, *126*, 180–194. [[CrossRef](#)]
5. Bircher, A.; Alexis, K.; Burri, M.; Oettershagen, P.; Omari, S.; Mantel, T.; Siegwart, R. Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 25–30 May 2015; pp. 6423–6430.
6. Gonzalez-Aguilera, D.; Crespo-Matellan, E.; Hernandez-Lopez, D.; Rodriguez-Gonzalvez, P. Automated urban analysis based on LiDAR-derived building models. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 1844–1851. [[CrossRef](#)]

7. Li, Y.; Tong, G.; Du, X.; Yang, X.; Zhang, J.; Yang, L. A Single Point-Based Multilevel Features Fusion and Pyramid Neighborhood Optimization Method for ALS Point Cloud Classification. *Appl. Sci.* **2019**, *9*, 951. [\[CrossRef\]](#)
8. Ni, H.; Lin, X.; Zhang, J. Classification of ALS point cloud with improved point cloud segmentation and random forests. *Remote Sens.* **2017**, *9*, 288. [\[CrossRef\]](#)
9. 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.* **2015**, *105*, 286–304. [\[CrossRef\]](#)
10. Mei, J.; Zhang, L.; Wang, Y.; Zhu, Z.; Ding, H. Joint Margin, Cograph, and Label Constraints for Semisupervised Scene Parsing From Point Clouds. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 3800–3813. [\[CrossRef\]](#)
11. Li, Y.; Tong, G.; Li, X.; Zhang, L.; Peng, H. MVF-CNN: Fusion of Multilevel Features for Large-Scale Point Cloud Classification. *IEEE Access* **2019**, *7*, 46522–46537. [\[CrossRef\]](#)
12. Zhang, Z.; Zhang, L.; Tan, Y.; Zhang, L.; Liu, F.; Zhong, R. Joint discriminative dictionary and classifier learning for ALS point cloud classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 524–538. [\[CrossRef\]](#)
13. Xu, Z.; Zhang, Z.; Zhong, R.; Chen, D.; Sun, T.; Deng, X.; Li, Z.; Qin, C.Z. Content-Sensitive Multilevel Point Cluster Construction for ALS Point Cloud Classification. *Remote Sens.* **2019**, *11*, 342. [\[CrossRef\]](#)
14. Jeong, J.; Lee, I. Classification of LIDAR Data for Generating a High-Precision Roadway Map. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 251–254. [\[CrossRef\]](#)
15. Tóvári, D.; Pfeifer, N. Segmentation based robust interpolation—a new approach to laser data filtering. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2005**, *36*, 79–84.
16. Papon, J.; Abramov, A.; Schoeler, M.; Worgotter, F. Voxel cloud connectivity segmentation-supervoxels for point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2027–2034.
17. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [\[CrossRef\]](#)
18. Wang, Y.; Shi, H. A Segmentation Method for Point Cloud Based on Local Sample and Statistic Inference. In *Geo-Informatics in Resource Management and Sustainable Ecosystem*; Springer: Berlin, Germany, 2015; pp. 274–282.
19. Ni, K.; Jin, H.; Dellaert, F. Groupsac: Efficient consensus in the presence of groupings. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009; pp. 2193–2200.
20. Shahbazi, M.; Sohn, G.; Théau, J. Evolutionary optimization for robust epipolar-geometry estimation and outlier detection. *Algorithms* **2017**, *10*, 87. [\[CrossRef\]](#)
21. Raguram, R.; Chum, O.; Pollefeys, M.; Matas, J.; Frahm, J.M. USAC: A universal framework for random sample consensus. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2022–2038. [\[CrossRef\]](#)
22. Awadallah, M.; Abbott, L.; Ghannam, S. Segmentation of sparse noisy point clouds using active contour models. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 6061–6065.
23. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, 21 June–18 July 1967; Volume 1, pp. 281–297.
24. Wu, Y.; Li, F.; Liu, F.; Cheng, L.; Guo, L. Point Cloud Segmentation Using Euclidean Cluster Extraction Algorithm With the Smoothness. *Meas. Control Technol.* **2016**, *35*, 36–38.
25. Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 790–799. [\[CrossRef\]](#)
26. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *5*, 603–619. [\[CrossRef\]](#)
27. Mei, J.; Wang, Y.; Zhang, L.; Zhang, B.; Liu, S.; Zhu, P.; Ren, Y. PSASL: Pixel-Level and Superpixel-Level Aware Subspace Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**. [\[CrossRef\]](#)

28. Feng, C.; Taguchi, Y.; Kamat, V.R. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 6218–6225.
29. Zhang, Z.; Zhang, L.; Tong, X.; Mathiopoulos, P.T.; Guo, B.; Huang, X.; Wang, Z.; Wang, Y. A multilevel point-cluster-based discriminative feature for ALS point cloud classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3309–3321. [[CrossRef](#)]
30. Xu, Y.; Yao, W.; Tüttas, S.; Hoegner, L.; Stilla, U. Unsupervised segmentation of point clouds from buildings using hierarchical clustering based on Gestalt principles. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2018**, *99*, 1–17. [[CrossRef](#)]
31. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd* **1996**, *96*, 226–231.
32. Liu, P.; Zhou, D.; Wu, N. VDBSCAN: Varied density based spatial clustering of applications with noise. In Proceedings of the 2007 International Conference on Service Systems and Service Management, Chengdu, China, 9–11 June 2007; pp. 1–4.
33. Li, M.; Sun, C. Refinement of LiDAR point clouds using a super voxel based approach. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 213–221. [[CrossRef](#)]
34. Rusu, R.B.; Bradski, G.; Thibaux, R.; Hsu, J. Fast 3d recognition and pose using the viewpoint feature histogram. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 2155–2162.
35. Aldoma, A.; Vincze, M.; Blodow, N.; Gossow, D.; Gedikli, S.; Rusu, R.B.; Bradski, G. CAD-model recognition and 6DOF pose estimation using 3D cues. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 585–592.
36. Lai, K.; Fox, D. Object recognition in 3D point clouds using web data and domain adaptation. *Int. J. Robot. Res.* **2010**, *29*, 1019–1037. [[CrossRef](#)]
37. Chen, D.; Peethambaran, J.; Zhang, Z. A supervoxel-based vegetation classification via decomposition and modelling of full-waveform airborne laser scanning data. *Int. J. Remote Sens.* **2018**, *39*, 2937–2968. [[CrossRef](#)]
38. 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](#)]
39. Wang, Z.; Zhang, L.; Zhang, L.; Li, R.; Zheng, Y.; Zhu, Z. A Deep Neural Network With Spatial Pooling (DNNSP) for 3-D Point Cloud Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4594–4604. [[CrossRef](#)]
40. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning, Williamstown, MA, USA, 28 June–1 July 2001; Volume 1, pp. 282–289.
41. Kohli, P.; Kumar, M.P.; Torr, P.H. P^3 & Beyond: Move Making Algorithms for Solving Higher Order Functions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 1645–1656.
42. Kohli, P.; Torr, P.H. Robust higher order potentials for enforcing label consistency. *Int. J. Comput. Vis.* **2009**, *82*, 302–324. [[CrossRef](#)]
43. Vineet, V.; Warrell, J.; Torr, P.H. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. *Int. J. Comput. Vis.* **2014**, *110*, 290–307. [[CrossRef](#)]
44. Fix, A.; Wang, C.; Zabih, R. A primal-dual algorithm for higher-order multilabel markov random fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1138–1145.
45. Guo, B.; Huang, X.; Zhang, F.; Sohn, G. Classification of airborne laser scanning data using JointBoost. *ISPRS J. Photogramm. Remote Sens.* **2015**, *100*, 71–83. [[CrossRef](#)]
46. Wang, Z.; Zhang, L.; Fang, T.; Mathiopoulos, P.T.; Tong, X.; Qu, H.; Xiao, Z.; Li, F.; Chen, D. A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2409–2425. [[CrossRef](#)]
47. Zhang, Z.; Zhang, L.; Tong, X.; Guo, B.; Zhang, L.; Xing, X. Discriminative-dictionary-learning-based multilevel point-cluster features for ALS point-cloud classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7309–7322. [[CrossRef](#)]
48. Zhang, Z. ALS Point Cloud Classification Based on Multilevel Point Cluster Features. Ph.D. Thesis, Beijing Normal University, Beijing, China, 2016.

49. Xia, S.; Wang, R. Extraction of residential building instances in suburban areas from mobile LiDAR data. *ISPRS J. Photogramm. Remote Sens.* **2018**, *144*, 453–468. [[CrossRef](#)]
50. Khoshelham, K.; Nardinocchi, C.; Frontoni, E.; Mancini, A.; Zingaretti, P. Performance evaluation of automated approaches to building detection in multi-source aerial data. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 123–133. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).