remote sensing

MDPI

Article

MSNet: Multi-Scale Convolutional Network for Point Cloud Classification

Lei Wang [1], Yuchun Huang [2,*], Jie Shan [2,3] and Liu He [4]

1 State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; wlei@whu.edu.cn
2 School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China; jshan@purdue.edu
3 Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA
4 Department of Geography, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3220, USA; liuhe95@unc.edu
* Correspondence: hycwhu@whu.edu.cn; Tel.: +86-180-6410-3611

Received: 15 March 2018; Accepted: 12 April 2018; Published: 17 April 2018

Abstract: Point cloud classification is quite challenging due to the influence of noise, occlusion, and the variety of types and sizes of objects. Currently, most methods mainly focus on subjectively designing and extracting features. However, the features rely on prior knowledge, and it is also difficult to accurately characterize the complex objects of point clouds. In this paper, we propose a concise multi-scale convolutional network (MSNet) for adaptive and robust point cloud classification. Both the local feature and global context are incorporated for this purpose. First, around each point, the spatial contexts of different sizes are partitioned as voxels of different scales. A voxel-based MSNet is then simultaneously applied at multiple scales to adaptively learn the discriminative local features. The class probability of a point cloud is predicted by fusing the features together across multiple scales. Finally, the predicted class probabilities of MSNet are optimized globally using the conditional random field (CRF) with a spatial consistency constraint. The proposed method was tested with data sets of mobile laser scanning (MLS), terrestrial laser scanning (TLS), and airborne laser scanning (ALS) point clouds. The experimental results show that the proposed method was able to achieve appreciable classification accuracies of 83.18%, 98.24%, and 97.02% on the MLS, TLS, and ALS data sets, respectively. The results also demonstrate that the proposed network has a strong generalization capability for classifying different kinds of point clouds under complex urban environments.

Keywords: point cloud; multi-scale convolutional network (MSNet); conditional random field (CRF); classification

1. Introduction

Point clouds are widely available now due to the progressive development of various laser sensors and dense image matching techniques. The efficient classification of point clouds is one of the fundamental problems in scene understanding for three-dimensional (3D) digital cities, intelligent robots, and unmanned vehicles. However, classifying point clouds under complex urban environments is not a trivial task, since they are usually noisy, sparse, and unorganized [1]. The density of point clouds varies with the sampling intervals and ranges of laser scanners. Moreover, severe occlusions between objects during scanning can lead to incomplete coverage of object surfaces. These problems present challenges for point cloud classification.

Point cloud classification can be accomplished via various approaches, such as region growing, energy minimization, and machine learning. A review of these approaches can be found in Nguyen

and Le [1]. Region growing is a segmentation method that partitions point clouds into disjoint homogenous regions. Based on the catastrophe and non-continuous points of curvatures, the seed regions are grown with a geometrical and topological continuity of the points. However, the process is sensitive to noise in the point cloud. One of the solutions to this problem is to grow the regions according to the local geometrical characteristics [2] so that the noise can be largely suppressed. Another way to solve this problem is to jointly use the edge and local region characters [3] to reduce the computational complexity and improve the classification accuracy. However, the region growing approach is susceptible to the initial seeds [4]; inaccurate seeds may lead to under or over-segmentation, and the growing process is difficult to stop when the transitions between two regions are smooth [5].

The energy minimization approach is a global solution framework that formulates the classification as an optimization problem [6]. The process starts by considering the point cloud as a graph. Each vertex corresponds to a point in the data, and the edges connect neighboring points [7,8]. By turning the segmentation into a min-cut/max-flow problem, the point cloud is segmented with graph cuts [9–11]. In addition, many studies on the graph approach also cast it into a probabilistic inference model, such as the conditional random field (CRF) [12,13], which regards the classification as a multi-labeling optimization problem. By constraining the fidelity of the data, the continuity of feature values, and the compactness of the segment boundaries [6], the energy function is minimized to ensure that the statistic characteristic difference is the minimum in the same class and the maximum between different classes [9,10]. To efficiently combine other constraints, the weights in the graph also can be computed as a combination of Euclidean distances, pixel intensity differences, and the angles between the surface normals, among others [14]. This global optimization approach is insensitive to noise, but its segmented results are usually piecemeal.

Machine learning aims to train an efficient classifier from enormous numbers of samples. To this end, a proper feature descriptor is important for the classification model. Generally, the currently available feature descriptors can be divided into three categories: (1) global, (2) local, and (3) regional [15]. Global descriptors describe the holistic statistical characteristics of a class of objects, and are useful in object retrieval and recognition. However, they are sensitive to incomplete object extraction, which is a common problem in point cloud processing. Local descriptors represent the local properties of objects, such as the surface normals, surface curvatures, and eigenvalues of the covariance matrix; however, they are sensitive to noise. Regional descriptors further introduce some neighboring region information, including texture [16], geometrical structure [17], topology [18], and contexture [19]. In addition, as objects often present different properties at different spatial scales, the multi-scale or multi-resolution spatial feature descriptors [20–22] can describe the objects across different scales. After the feature descriptors are extracted, the point cloud is classified with a machine learning classifier. The commonly used machine learning methods include support vector machine (SVM) [23,24], cascaded AdaBoost [25], and random forest [26,27]. However, the accuracy of these classifiers is similar and sensitive to the selection of the feature descriptors [25]. A good feature descriptor should be discriminative enough to adapt to complex situations. Nonetheless, the procedure of the aforementioned feature extraction techniques is "knowledge-driven", and highly relies on the human operator's a priori knowledge, which can be an overwhelming task with complex urban environments.

Over the last several years, deep learning has led to substantial gains in many areas, owing to its powerful feature learning ability. It simulates the cognitive processes of the human brain to learn the discriminative features from enormous amounts of data. As one of the deep learning networks, the convolutional neural network (CNN) uses convolution kernels to simulate the receptive fields of the vision system. CNN has become one of the most efficient methods for image classification, document analysis, voice detection, and video analysis [28,29], among others. For 3D object recognition [30,31], which aims to assign a reasonable label to a cluster of points, CNN has achieved promising results for discriminative feature extraction and representation [32,33].

As for 3D point cloud classification, each point is labeled separately. Some researchers [34] project the point cloud to a plane so that the standard two-dimensional (2D) CNN can be applied. Nevertheless,

due to the occlusion among objects, directly projecting the point cloud to imagery inevitably loses the depth and 3D spatial structure information. Another way to apply CNN to a point cloud is by voxelizing the entire unorganized 3D scene into a 3D array of point clouds. This would allow using some classical image semantic labeling networks such as FCN [35], SegNet [36], Deeplab [37], and DeconvNet [38] for point cloud classification by extending 2D convolution kernels into 3D ones. However, a point cloud is not actually the "3D data" of whole solid objects; rather, it is a recording of the objects' surfaces, which actually is a manifold that is embedded in the 3D space. Except for the objects' surfaces, the 3D space is filled with enormous null data. Simply voxelizing the entire 3D point cloud into a regular 3D array can lead to huge unnecessary computation. Therefore, an efficient network constructed directly on the point or voxel has increasingly become a topic of interest among research studies [39–41].

One of our arguments is that objects of various sizes exist in a point cloud. For objects that are small in size, a fine scale within a small neighboring region is enough, while large objects require a coarse scale containing a large region. To adapt to the varying sizes of objects, multi-scale voxelization is proposed in this paper to "observe" the small neighbors finitely and the large neighbors roughly. Instead of dividing the whole space into voxels with a fixed size, multi-scale voxelization divides a point cloud into voxels at multiple scales, thereby allowing the multi-scale features of objects of various sizes to be extracted based on those voxels. Also, the spatial context information at different scales is integrated during multi-scale feature extraction.

Based on multi-scale voxelization, we further propose a multi-scale convolutional network (MSNet), with the aim of efficient feature learning and class prediction. In our proposed method, only the position information (x, y, and z coordinates) of a point cloud was considered, as the intensity or RGB information is not always available. Among the neighboring voxels around a point, MSNet is proposed for the discriminative feature learning of its local context. The multi-scale features of different spatial resolutions are learned with convolutional networks and fused across different scales. Meanwhile, as a result of multi-scale voxelization, the spatial context with different sizes is captured at different scales by the 3D convolution kernels of MSNet. With this strategy, the conventional pooling operation is not necessary in MSNet to robustly capture the multi-scale features, so the structure of MSNet becomes concise to implement. However, the classification of point clouds with MSNet is voxel-level, and inevitably influenced by noise. As such, a CRF that fully considers the spatial consistency of a point cloud is applied to achieve a global optimization of the predicted class probabilities. Therefore, our method incorporates both local and global constraints for a highly accurate point cloud classification.

The remainder of this paper is structured as follows. Section 2 starts with multi-scale point cloud voxelization. MSNet is then established for the discriminative local feature learning, and global label optimization with CRF is employed. Section 3 starts with the experimental data description, followed by a presentation of the individual and overall test results to demonstrate the solution procedure. Section 4 is a series of discussions where we compare our proposed MSNet with some state-of-the-art methods, and analyze the generalization capability of the proposed approach. Both quantitative and qualitative evaluations are presented. Section 5 consists of our concluding remarks on the properties of MSNet and our proposed future efforts.

2. Materials and Methods

As depicted in Figure 1, the proposed method consists of two complementary parts. In Part I of Figure 1, a point cloud is represented as multi-scale 3D voxels. Then, a corresponding MSNet is established for discriminative local feature learning to predict a class probability. In Part II of Figure 1, the point cloud is regarded as an edge-weighted graph, and a CRF with spatial consistency constraints is constructed to obtain the global context. Finally, global label optimization is used to combine the local feature and the global context for accurate classification of the point cloud.

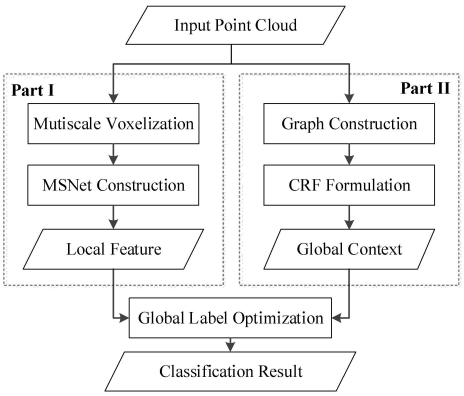


Figure 1. Flowchart of the proposed multi-scale convolutional network (MSNet).

2.1. Multiscale Voxelization

Humans perceive the object context of point clouds at multiple scales, including the scene at a coarse scale, and then objects, structures, edges, and points at a fine scale. It is a multi-scale observation process that considers information across different scales to enable comprehensive judgment. Similarly, to automatically understand point clouds across different scales for discriminative feature learning, the context of a point cloud is analyzed by multi-scale voxels that are centered at the point, which allows the network to closely observe at a fine scale, and a consider rough view at a coarse scale.

At each scale, for a given point $P(x, y, z)$, a neighboring cubic $[x - 0.5R, x + 0.5R] \times [y - 0.5R, y + 0.5R] \times [z - 0.5R, z + 0.5R]$ with length R is set up around it, as shown in Figure 2a. Then, the cube is subdivided into $n \times n \times n$ grid voxels [42] as a patch, and the side length of each voxel is $r = R/n$. R corresponds to the size of the neighboring region. The smaller the r is, the finer the scale. For small objects, a fine scale within a small neighboring region R is enough, whereas for large objects, a coarse scale including a large region is needed. On the contrary, observing small objects at coarse scale will omit details, while the processing of large objects with a fine scale may lead to high noise sensibility and large unnecessary computation. Therefore, dividing the point cloud with multiple scales is necessary to accommodate the various sizes of objects.

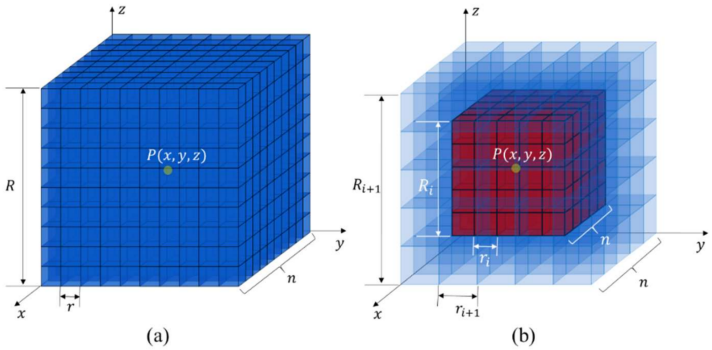


Figure 2. Point cloud voxelization at (**a**) uniform scale and (**b**) multiple scales. Around point $P(x, y, z)$, the spatial contexts of different sizes are partitioned as voxels of different scales, respectively. The points near $P(x, y, z)$ are more important, and are characterized with finer scales in several voxels than those far away from it.

According to the aforementioned analysis, we present the design of a multi-scale voxelization frame in this paper. The patch length n of different scales is fixed to an equal number that is as small as possible for computation efficiency. Then, a series of voxel side lengths $\{r_1, r_2, \ldots, r_S\}$ with increasing values are applied. With a larger $r \in \{r_1, r_2, \ldots, r_S\}$, the cubic side length R also will increase and yield a coarse

view in a large region, and vice versa. Instead of dividing the whole space into voxels of a fixed size, the multi-scale voxelization divides the individual context of the point cloud into voxels at multiple scales, and the spatial context information of different scales is well represented for each point. As shown in Figure 2b, with an equal patch length n of all of the scales, the spatial context with different sizes are obtained by changing the voxel length r_i $(i = 1, 2, \ldots, s)$ without compromising the computation efficiency.

Without loss of generality, the point density of each voxel, which is defined as the ratio of the point count within the voxel and its volume, is adopted as the representative value of the voxel. For fine voxel volume, the commonly-used occupancy value (i.e., if there is a point inside the voxel, the value is set as 1, otherwise it becomes 0) is reasonable, as the voxel is small, and there are only a few or no points that lie in it. Whereas, for the coarse voxel volume, the number of points that lie in the voxels may vary greatly, and cannot be simply approximated as 0 or 1. Compared with the occupancy value, the point density characterizes the degree of point occupancy within a voxel, and is invariant to scale.

2.2. Multi-Scale Convolutional Network of a 3D Point Cloud

Based on the multi-scale voxelization, MSNet is proposed for discriminative local feature learning and class probability prediction, as shown in Figure 3. With the multi-scale voxelization of point clouds, the multi-scale features of different spatial resolutions are learned with a series of convolutional networks (ConvNets) of shared weights that are fused directly across different scales. Due to multi-scale voxelization, the 3D convolution kernels of MSNet capture the spatial context with different sizes at different scales. Thus, the cascaded pooling operation is not necessary, and a concise structure of less model parameters is proposed in MSNet.

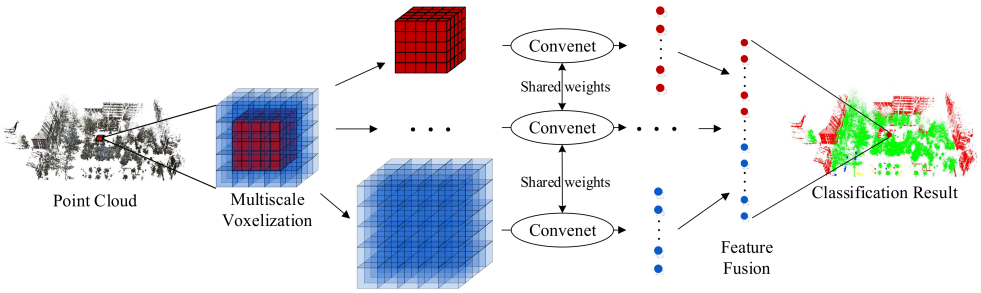


Figure 3. Multi-scale convolutional network.

2.2.1. Multi-scale Feature Extraction

Many excellent discriminative feature extraction methods have been proposed for point cloud classification [16–18]. However, most of them are "knowledge-driven", and are designed subjectively based on prior knowledge. Due to the influence of noise, occlusion, and various types and sizes of objects, these subjectively-designed features are difficult to use for characterizing the objects in a point cloud.

Owing to its convolution and pooling layer, CNN has recently been shown to have a powerful feature learning capability in the classification and semantic labeling of 2D images [28,29]. The kernels of the convolution layer simulate the receptive fields of human vision, while the pooling layer is applied for dimension reduction and an invariance guarantee of translation, rotation, and scale.

However, it is difficult to directly utilize the conventional CNN for 3D point cloud classification. CNN needs an input of regular 2D or 3D array, but when the 3D point cloud is simply projected into 2D imagery, the 3D point cloud loses its 3D spatial context information. Dividing the point cloud into a regular 3D array cannot adaptively reflect the different sizes of the objects in the point cloud, and also will lead to large unnecessary computation on null values inside the object, even with the following pooling operation.

To address these problems, MSNet is proposed based on the 3D multi-scale voxelization of the point cloud. By simultaneously applying the ConvNets at multiple scales, the multi-scale contextual features

of the objects of different sizes in the point cloud are extracted with discriminative feature learning. The ConvNets at different scales operates within the spatial context of different region sizes, which acts as the cascaded pooling operations in normal CNN. Therefore, the pooling layer is not necessary in MSNet, and a less deep structure is achieved due to the simultaneous convolution at multiple scales.

At each point (x, y, z) in the 3D scene, we first construct the corresponding multiscale 3D voxels according to Section 2.1. Denote the patch of scale $s \in \{1, \ldots, S\}$ as $V_s \in R^{n \times n \times n \times q_0}$. In the superscript of R, the last dimension of V_s represents the number of features. In this paper, only the voxel's density is considered, leading to the last dimension $q_0 = 1$. For each scale s, the 3D Convnet F_s can be described as a sequence of linear transforms and non-linear activations. For the 3D Convnet F_s with L layers, the m-th output feature map of layer $l \in \{1, 2, \ldots, L\}$ can be represented as:

$$H_{s,l}^{(m)} = Relu\left(\sum_{t=1}^{q_{l-1}} H_{s,l-1}^{(t)} * W_{s,l}^{(t,m)} + b_{s,l}^{(m)}\right) \tag{1}$$

for all of $m \in \{1, 2, \ldots, q_l\}$ with $b_{s,l} \in R^{q_l}$ as a vector of bias; where $H_{s,0} = V_s$. $W_{s,l}^{(t)} \in R^{f_l \times f_l \times f_l}$ is the convolution kernel with a size of f_l, q_l is the feature number of hidden layer $H_{s,l}$, and $*$ represents the 3D convolution operator. $Relu(\cdot) = \max(\cdot, 0)$ is the activation function acting on each element of the input matrix, which leads to the non-linearity of the network and reduces the vanishing of the gradient and fast training. In addition, the contribution of the neighboring voxels to the center one is similar across different scales, and depends on their spatial relationship. To capture this character and improve the generalization capability of our MSNet, the weights of ConvNets are shared across different scales to reduce the number of model parameters, which also makes the MSNet concise.

The output of the 3D Convnet F_s is obtained as:

$$F_s(V_s) = \left[H_{s,L}^{(1)}, H_{s,L}^{(2)}, \ldots, H_{s,L}^{(q_L)}\right] \tag{2}$$

It is regarded as the feature of point (x, y, z) at scale $s \in \{1, \ldots, S\}$. A detailed convolution process at a single scale is provided in Figure 4.

Finally, the outputs of the S-scale ConvNets are flattened and fused to produce the final feature vector $\hat{F}$, which can be seen as the multi-scale feature around point (x, y, z):

$$\hat{F} = Relu(W_f[flatten(F_1), flatten(F_2), \ldots, flatten(F_S)] + b_f) \tag{3}$$

where $flatten(\cdot)$ is the flatten function to stretch the matrix to be a vector, W_f represents the full connection parameters, and b_f is the corresponding bias.

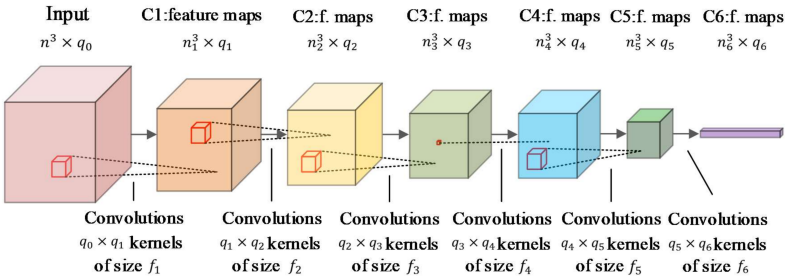


Figure 4. The three-dimensional 3D convolutional networks (Convnet) F_s with $L = 6$ layers. The input of the network is a patch with q_0 feature channels. A sequence of convolution kernels are applied for multi-layer feature learning (without padding for high computation efficiency), and the size of the output at l-th layer is denoted as $n_l = n_{l-1} - f_l + 1$, $l = 1, 2, \ldots, L$, and $n_0 = n$. The final output is a feature vector ($n_6 = 1$). To this end, the kernel size of the last layer is the same as the output of the former layer ($f_6 = n_5$).

2.2.2. Discriminative Feature Learning

With the fused multi-scale feature $\hat{F}$, our goal is to use it for class probability prediction. To this end, we apply softmax regression to predict the probability distribution p over each class as:

$$\hat{p}_{i,k} = softmax(\hat{F}_i) = \frac{e^{W_k^T \hat{F}_i}}{\sum_{c_j \in \mathbb{C}} e^{W_j^T \hat{F}_i}} \tag{4}$$

where $\hat{p}_{i,k}$ is the predicted probability for the i-th point belonging to class $c_k \in \mathbb{C}$, $\mathbb{C} = \{c_1, c_2, \ldots, c_C\}$ is denoted as the set of classes, and C represents the number of classes.

Next, we construct the loss function using cross entropy, which depicts the difference between the probability distribution $\hat{p}_{i,k}$ and the true probability distribution $p_{i,k}$ of class c_k:

$$Loss(\Theta) = -\sum_{i \in voxels} \sum_{k \in \mathbb{C}} p_{i,k} \ln\left(\hat{p}_{i,k}(\Theta)\right) \tag{5}$$

where the parameters $\Theta = \left\{W_1, b_1, W_2, b_2, \ldots, W_L, b_L, W_{flatten}, b_{flatten}\right\}$ can be learned by minimizing $Loss(\Theta)$ with a batch stochastic gradient descent algorithm. Once the network is learned, the loss function is no longer needed, and the predicted probabilities are used for further class label inference.

2.3. Global Label Optimization with CRF

Point cloud classification must assign each point a unique label that indicates its class. The simplest strategy for this end is to give each voxel a label with the argmax of the predicted probabilities (Equation (4)). Then, the label is assigned to the points in the corresponding voxel. However, such classification results are at the voxel level, and are inevitably influenced by noise and result in the spatial inconsistency of label prediction.

To address this issue, we use a CRF model with spatial consistency to globally optimize the class label of the point cloud. For this purpose, we construct a graph $G(V, E)$ with vertex $v \in V$ and edge $e \in E$. Each vertex is associated to a point, and the edges are added between the point and its K-nearest points of the point cloud.

Let random variable X_i be the label of point i. Random variable X consists of $X_1, X_2, \ldots, X_N$, where N is the total number of the points. We regard vertex V of the graph $G(V, E)$ as the random variable of label (i.e., $V = \{X_1, X_2, \ldots, X_N\}$). We can constitute the CRF model (P, X) based on the graph $G(V, E)$ of the point cloud, where P is the global observation of $G(V, E)$. $P = \{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_N\}$ corresponds to the predicted class probability of the point cloud, which is obtained with MSNet. The posterior probability of the point cloud that is assigned to label l, which consists of $l_1, l_2, \ldots, l_N$ $(l_i \in \mathbb{C} = \{c_1, c_2, \ldots, c_C\})$ under the global observation P, is then represented as below:

$$p(X = l|P) = \frac{1}{Z(P)} \exp\left(-E(l|P)\right) \tag{6}$$

where $Z(P)$ indicates the normalized index, and the energy of label l can be represented as:

$$E(l|P) = \sum_{i \in V} \varphi(\hat{p}_i, l_i) + \sum_{(i,j) \in E} \psi(l_i, l_j) \tag{7}$$

Label l maximizing the posterior probability $p(X = l|P)$ is the most appropriate label of the point cloud, whereas maximizing the posterior probability in Equation (6) is equal to minimizing the energy in Equation (6), which leads to a global optimization of the point cloud label.

The data cost term $\varphi(\hat{p}_i, l_i)$ penalizes the disagreement between a point and its assigned label. In this paper, the initial data cost of each point is calculated with its predicted probability in Section 2.2 as unary terms:

$$\varphi(\hat{p}_{i,k}, l_i) = \exp(-\hat{p}_{i,k})\delta(l_i \neq c_k) \tag{8}$$

where $\delta(\cdot)$ is an indicator function. The data cost enforces the value of label l close to the predicted probability.

The smooth cost term $\psi(l_i, l_j)$ penalizes the label inconsistency between neighboring points. The neighboring points are encouraged to assign similar labels. In this work, the K-nearest neighboring points are connected with the central point. The smooth cost is calculated according to the Euclidean distance between two points:

$$\psi(l_i, l_j) = \frac{1 - \exp(-d_{i,j})}{1 + \exp(-d_{i,j})}\delta(l_i \neq l_j) \tag{9}$$

where $d_{i,j}$ is the 3D Euclidean distance between points i and j. The smooth cost constrains the regularity and consistency of label l.

Finally, the energy function $E(l|P)$ is minimized with the α–$expansion$ [43–45] algorithm. A simple diagram of the optimization process is provided in Figure 5. The initial probabilities of each point are pre-predicted with MSNet, as described in Section 2.2. After several iterations, all of the class labels of the point cloud are globally optimized.

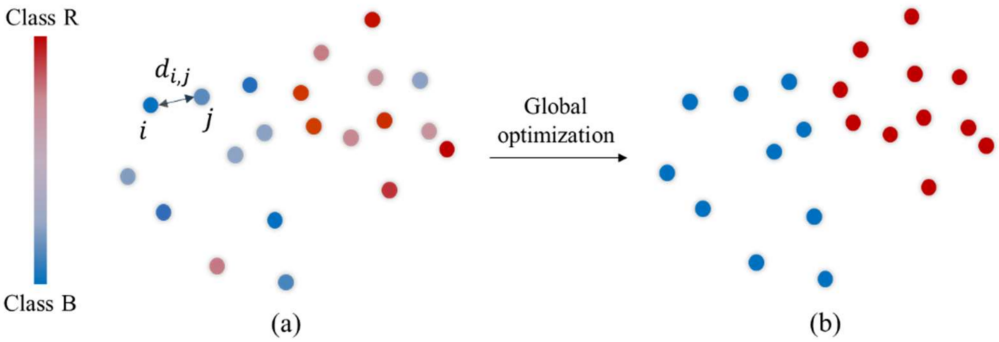


Figure 5. The global optimization aims to give each point a spatially consistent class label with the conditional random field (CRF). Assuming there are two classes R and B, the points in (**a**) describe the predicted probability of each class, which contributes to the initial data cost. The red and blue points totally belong to class R and class B, respectively. The color of the other points is mixed with the degree of red and blue, which is decided by the probability belonging to class R and class B, respectively. After global optimization, each point is assigned to be a certain class as shown in (**b**).

3. Results

3.1. Experimental Data

Both mobile laser scanning (MLS) and airborne laser scanning (ALS) point clouds were used to evaluate the proposed method, and included objects of different sizes and scanning densities. They were acquired from the same area of Wuhan University (WHU), China, and are available at https://github.com/wleigithub/WHU_pointcloud_dataset. An overview of the experimental area is shown in Figure 6, and the experimental data are provided in Figure 7. The properties of these data sets are summarized in Table 1.

The MLS point cloud with two blocks (block I and block II) was obtained with SICK LMS291 Laser Range Finder in March 2014. They are labeled into seven classes, which included vegetation (e.g., tree and grass), buildings, cars, pedestrians, lamps, and fences. The point density of the MLS

point cloud varied a lot with the different distances between the objects and scanners. Moreover, the point cloud of many of the objects often was incomplete due to mutual occlusion. The ALS point cloud (block III) was acquired in Wuhan, China in July 2014 by a Y5 plane carrying an H68-18 airborne laser radar system with a mean flight altitude of 800m above the ground. The point density in the experimental area was approximately 5–10 points/m^2. Compared with the MLS point cloud, the ALS point cloud was more sparse, and much more fragmented. Therefore, only three classes (i.e., vegetation, buildings, and cars) were recognizable in block III scanned with the ALS. Additionally, ground points, which were a large portion of the point cloud, were previously removed artificially in the experimental data sets. All of the data sets were divided into training and testing data by a randomly chosen plane. Human operators carefully labeled the data sets with the CloudCompare (http://www.cloudcompare.org/) tool. Figure 7 provides an overview of the three blocks (2,330,834, 5,023,784, and 804,836 points respectively).

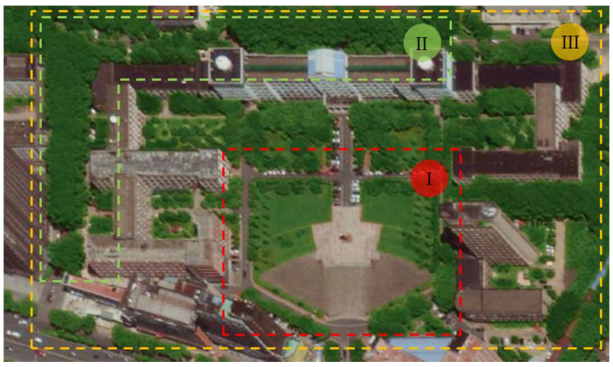


Figure 6. Study area. The three dashed boxes with different colors indicate different blocks for experiments. Blocks I and II were scanned with mobile laser scanning (MLS), while block III was scanned with airborne laser scanning (ALS) with a sparser point density.

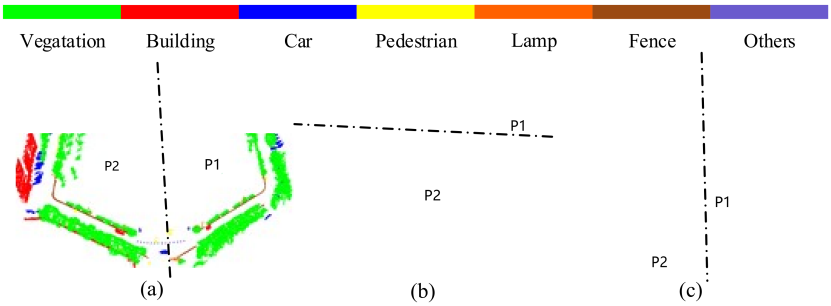


Figure 7. Experimental data overview. (**a**,**b**) are the MLS point cloud of block I and block II; (**c**) is the ALS point cloud of block III. Each block is divided into two parts, P1 and P2, by a vertical plane marked with a dash line. P1 and P2 are the training data and testing data, respectively.

Table 1. Properties of the experimental data.

Site	Block I	Block II	Block III
Scanning method	MLS		ALS
Equipment	LMS 291		H68-18
Acquisition data	March 2014		July 2014
Accuracy of measurement	10 mm		-
Total points	2,330,834	5,023,784	804,836
Points for training	1,049,782	2,213,511	398,987
Points for testing	1,281,052	2,810,273	405,876

3.2. Experimental Results and Assessment

At each point in 3D space, the corresponding multi-scale voxels were constructed among its neighbors. For the training samples, each voxel was assigned a unique label based on the majority of the neighboring labels in the voxel. For equal numbers of labels, a label was chosen randomly among them. Meanwhile, to simulate objects with different orientations, the training samples were randomly rotated around the voxels' central vertical axis. Additionally, k-fold cross validation was used for all of the experiments.

Considering both performance and efficiency, deeper layers would lead to higher accuracy, but more computation expense at the same time. $L = 6$ hidden layers were chosen and applied. In addition, n was fixed at 11 for computation efficiency, $f_1 = f_2 = f_3 = f_5 = f_6 = 3$ for feature learning, and $f_4 = 1$ for feature dimension reduction. In this way, the output of each scale corresponded to a feature vector. Moreover, the seven nearest neighbors of each point are searched to construct the graph $G(V, E)$ for global label optimization.

3.2.1. Classification of Point Clouds

The MLS point cloud in Figure 7a,b contains different objects, such as vegetation, buildings, cars, pedestrians, lamps, fences, and others (sculptures, roadblocks, and trash bins). To classify them, we used $S = 5$ scales for point cloud voxelization due to the wide range of object sizes, and n was fixed at 11 for computation efficiency. Based on the object sizes and the point cloud density in the space, the side length of the finest scale was set as r_1 =0.035 m (i.e., slightly larger than the average resolution of the point cloud). Additionally, similar to most multi-scale strategies, we set the side length of the next scale as two times that of the current scale, which was formulated as $r_{i+1} = 2r_i$, $i = 1, 2, \ldots, S-1$. Then, the side lengths of the other four scales were derived as $r_2 = 0.07$ m, r_3 = 0.14 m, $r_4 = 0.28$ m, and $r_5 = 0.56$ m, respectively. These scales allowed us to characterize the point cloud with five different scales. The finest scale was $r_1 = 0.035$ m, and the neighboring region around it was $0.385 \times 0.385 \times 0.385$ m, and focused on the details. The coarest scale was $r_5 = 0.56$ m, and the spatial context within the neighborhood was $6.16 \times 6.16 \times 6.16$ m^3.

For the ALS point cloud with lower density and larger objects, the side length of the finest scale was set as r_1 =0.07 m. Similar to the experiment of the MLS point cloud, five scales were applied for point cloud voxelization, and n was fixed as 11. The side length of the next scale was set as two times that of the current scale. Therefore, the side lengths of the other four scales were $r_2 = 0.14$ m, r_3 = 0.28 m, $r_4 = 0.56$ m, and $r_5 = 1.12$ m, and the side lengths of the neighboring regions at each scale were calculated as $R_1 = 0.77$ m, $R_2 = 1.54$ m, r_3 = 3.08 m, $R_4 = 6.16$ m, and $R_5 = 12.32$ m, with $R_s = nr_s$, $s = 1, 2, \ldots, 5$.

The ground truths and the corresponding classification results with the proposed method are shown in the first and last columns of Figure 8. It can be seen from the first two rows of Figure 8 that the MLS point clouds containing vegetation, buildings, cars, lamps, and fences were correctly classified, despite their different sizes and shapes. Due to the multi-scale and discriminative feature extraction capability of the proposed MSNet, the spatial context of the objects in Figure 8 were well characterized. Objects, such as lamps in vegetation, cars with incomplete shapes, and fences under trees in Figure 9, were also correctly classified, although they were partially occluded. For the sparse ALS point cloud, the proposed method also exhibited a satisfactory classification result. The predicted classification results without global label optimization are shown in the second column of Figure 8. The comparison results show that even though the CRF did not significantly improve the classification result, it efficiently suppressed the influence of noise, and guaranteed the smoothness of the classification result, which was beneficial for further object detection, reconstruction, etc.

However, there were some situations that still were difficult to classify. These situations involved uncommon structures (e.g., the gatehouse and flower bed in region A, and the spheroidal roof in region D), and insufficient sampling (e.g., glass refraction in region B and the distant scanning of

cars in region C). As shown in Figure 10, they were mistakenly classified due to the lack of sufficient training samples or scanning coverage.

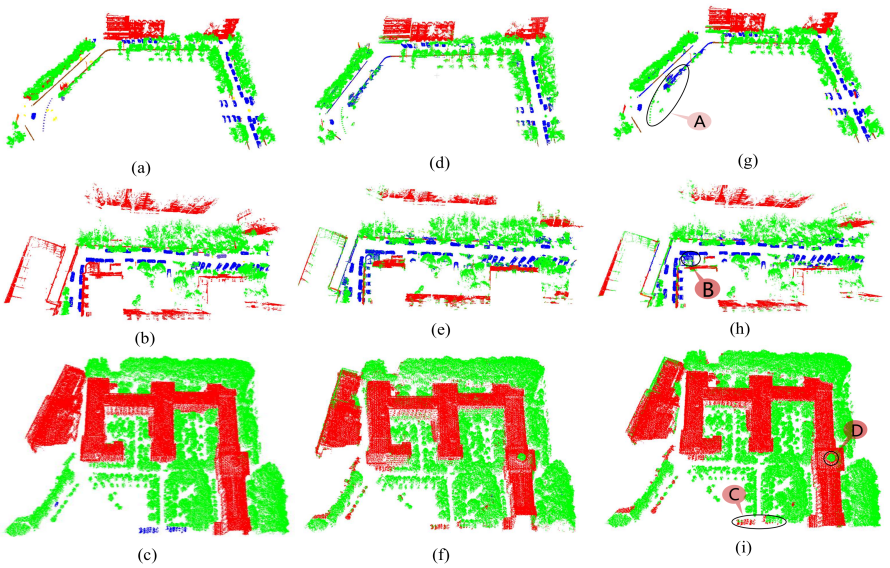


Figure 8. Classification results of test data. The first column shows the ground truth of the test data of blocks I, II, and III; the second column represents the predicted classification result without global label optimization; and the third column is the corresponding classification result with the proposed method. A, B, C, and D of the last column represent the typical cases that were mistakenly classified.

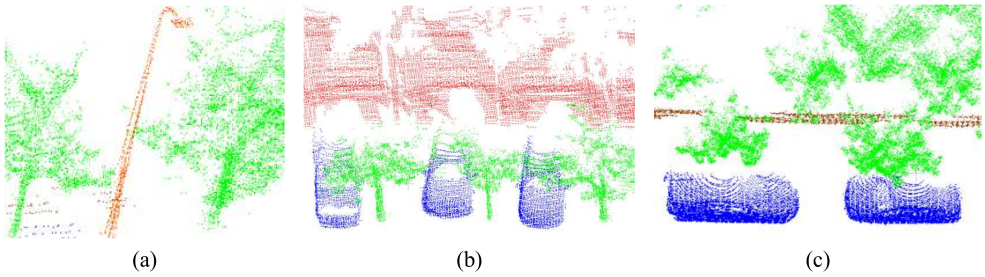


Figure 9. Details of some correctly classified objects. (**a**–**c**) show the lamp in the vegetation, the car with an incomplete shape, and the fence under the trees, respectively.

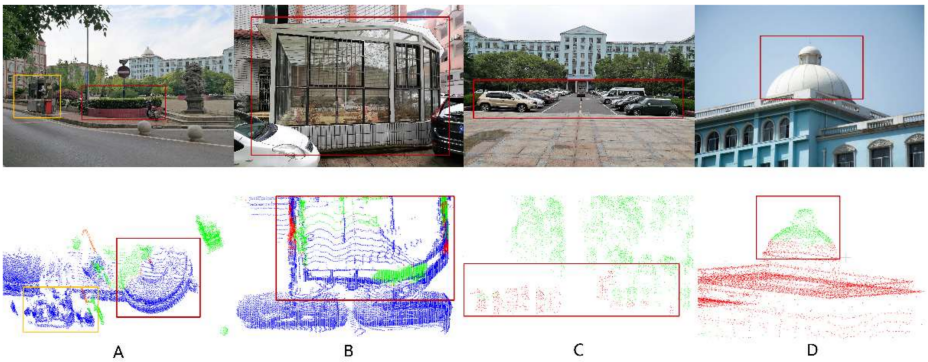


Figure 10. Actual situations corresponding to the typical cases that were mistakenly classified. The first row and the second row are the ground pictures and the point cloud details, respectively.

3.2.2. Assessment

Three metrics were used to quantitatively evaluate the performance of the proposed method. Precision is defined as the percentage of correctly classified points in the classification results, which is sensitive to the number of spurious points. Recall is defined as the percentage of correctly classified reference points, which is sensitive to the number of missed points. To further give a global assessment, the accuracy, which is the percentage of reference point cloud labels that are correctly predicted, was also considered in this paper. The three metrics are defined as:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$Accuracy = \frac{TP + TN}{TP + \mathrm{FP} + \mathrm{TN} + FN} \tag{12}$$

where TP is the number of true positives (i.e., the number of points both in the reference and classification), FP is the number of false positives (i.e., the number of classified points not found in the reference), FN is the number of false negatives (i.e., the number of reference points not found in classification), and TN is the number of true negatives (i.e., the number of points not found both in the reference and the classification). Similar to Li. et al. [11], precision/recall were used to represent the classification quality in this paper.

Quantitative evaluations of the experimental results are provided in Table 2. The classification accuracies of the MLS point clouds of blocks I and II were almost identical, at 83.18% and 82.98%, respectively; and the classification accuracy of the ALS point cloud of block III was relatively higher at 94.06%, due to its sparser point density and simpler object types (i.e., vegetation, buildings, and cars). Additionally, compared with the small-size objects (e.g., cars and lamps), the large-size objects (e.g., vegetation and buildings) were easier to classify. The incomplete objects caused by occlusion did not influence the classification of the large-size objects, while the shape of the small-size objects were easily obscured by the occlusion. Moreover, Table 2 shows that the recalls of vegetation and cars were higher than their precisions. Since the building walls and lamps were easily obscured by the surrounding trees, the precisions of the buildings and lamps were higher than their recalls.

Table 2. The precision/recall and accuracy of the test data with the proposed method.

	Test Data	Vegetation (%)	Buildings (%)	Cars (%)	Lamps (%)	Fence (%)	Others (%)	Accuracy (%)
With CRF	Block I	88.88/96.21	93.11/52.13	56.5/93.57	85.02/38.21	99.8/58.77	0/0	83.18
	Block II	76.84/93.87	99.64/73.59	65.77/98.52	11.93/31.44	-	0/0	82.98
	Block III	92.38/97.81	96.55/89.88	0/0	-	-	-	94.06
Without CRF	Block I	88.55/95.11	92.46/50.21	56.28/91.03	27.16/37.51	93.55/60.87	13.06/0.5	82.11
	Block II	76.24/91.62	99.76/71.53	61.40/95.86	3.51/33.04	-	0/0	80.81
	Block III	92.38/96.92	95.47/89.97	0/0	-	-	-	93.60

- Represents that there is no such kind of object in current block.

To evaluate the advantage of multi-scale voxelization over single-scale, three single-scale voxelizations (finest, middle, and coarsest scales) with the neighboring sizes of 0.385 m, 1.54 m^3, and 6.16 m respectively, were compared and tested using the point cloud of block I. The quantitative assessments of the experimental results are shown in Table 3. It can be seen that different voxelizations had different classification successes for objects that were of different sizes. For the finest voxelization, small-size objects, such as lamps, were classified satisfactorily, while it was difficult to distinguish buildings and vegetation. Coarser voxelization was more appropriate for extracting the distinctive feature of large-size objects. Comparing Table 3 with Table 2, it was concluded that multi-scale voxelization adaptively characterized and classified all of the objects better, regardless of their types and sizes.

Table 3. The precision/recall and accuracy of block I with single-scale voxelization.

Scale	Vegetation (%)	Buildings (%)	Cars (%)	Lamps (%)	Fence (%)	Others (%)	Accuracy (%)
Finest	92.60/1.06	21.30/96.87	33.94/73.52	66.34/35.02	95.57/6.85	0/0	25.79
Middle	81.07/91.50	86.64/38.65	51.34/72.52	38.43/5.36	91.78/65.26	30.18/21.61	76.31
Coarsest	82.05/96.19	92.73/45.40	56.50/88.48	13.83/10.20	34.74/5.27	2.01/0.36	76.69

4. Discussion

To further determine the performance of the proposed MSNet, comparison experiments with other state-of-the-art methods and generalization capability analysis were conducted to accomplish these experiments; another two data sets (terrestrial laser scanning [11,15] (TLS-Wang) and ALS [21] (ALS-Zhang)) point clouds were utilized. The TLS-Wang point cloud was obtained with a single terrain scanner, in which the majority of the objects were buildings, trees, cars, and pedestrians. The ALS-Zhang point cloud was acquired by a Leica ALS50 system with a mean flying height 500 m above ground, which contained three kinds of objects (vegetation, buildings, and cars). The details of the TLS-Wang and ALS-Zhang data sets can be found in Wang et al. [15] and Zhang et al. [21], respectively. Each data set included two scenes (i.e., scene I and scene II), as shown in Figure 11. Scene I was used for comparison experiments, and scene II was used for generalization capability analysis. Additionally, their ground points were removed prior to the analysis.

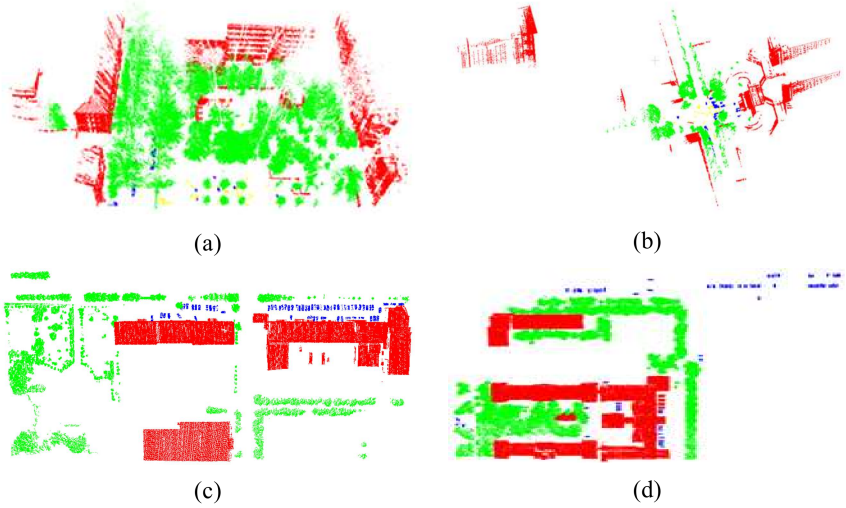
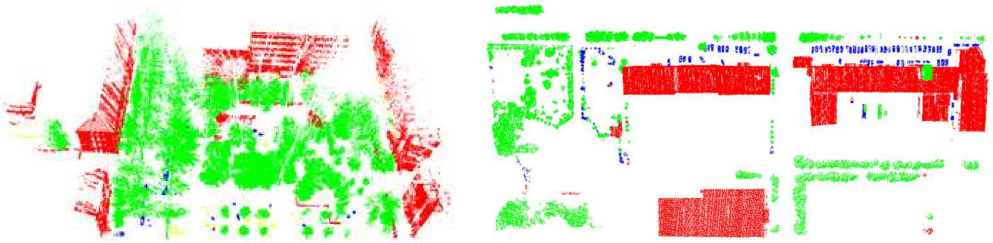


Figure 11. Ground truth of the terrestrial laser scanning (TLS-Wang) and ALS-Zhang point clouds. (**a**,**b**) are TLS-Wang scene I and II, while (**c**,**d**) represent ALS-Zhang scene I and II respectively.

4.1. Comparision with Other Methods

Considering the similarity of point cloud density and object size, the parameter settings for the TLS-Wang and ALS-Zhang point clouds in this section were the same as for the experiments on the MLS (MLS-WHU) and ALS (ALS-WHU) point clouds respectively. The number and size of the multi-scale voxelization were also the same. Additionally, there were only a few training samples in the data sets, which were insufficient for network training. To enrich the diversity of the samples, we randomly selected 50,000 samples, which accounted for 25% of the total amount, and rotated them with two arbitrary angles around a vertical axis. Together with the original samples in Wang et al. [15] and Zhang et al. [21], a total of 400,000 training samples were ultimately used for both the TLS-Wang and ALS-Zhang point clouds. The classification results of scene I are shown in Figure 12, and their corresponding ground truths are provided in Figure 11a,c. It can be seen that the proposed method successfully classified most of the objects in the TLS-Wang and ALS-Zhang point clouds.



(a) (b)

Figure 12. Classification results of comparative test data. (**a**,**b**) are the classification results of TLS-Wang scene I and ALS-Zhang scene I with the proposed method, respectively.

For the TLS-Wang point cloud classification, we compared the proposed method to other state-of-the-art methods (sLDA model [46], LDA model [15], object-oriented decision tree [11], and PointNet++) [41]. The precision/recall of each kind of object and the overall accuracy are listed in Table 4. In general, the proposed method achieved the highest accuracy at 98.24%, which was far higher than for the other methods. Moreover, its classification accuracy was also higher than the experimental results of the WHU data, as the TLS-Wang scene was relatively simpler. Similar to the experiments of the WHU point cloud, large-size objects (vegetation and buildings) were relatively easier to classify, as they were not sensitive to part of the occlusion and incompleteness.

For the ALS-Zhang point cloud classification, some other methods, including Guo et al. [47], Zhang et al. [21], and PointNet++ [41] were compared with the proposed method. Table 5 shows the precision/recall for each kind of object and the overall accuracy. It is noted that our proposed method did not only work well on a TLS point cloud, it also achieved the highest accuracy for ALS point cloud classification. The classification accuracy with our proposed method was 97.02%, which was far higher than the results achieved by other methods. For all of the comparison methods, cars were the most difficult to classify, due to the discretize error of the insufficient sampling effect. Moreover, some piecemeal and low vegetation was mistakenly classified, because the overhead view of the shape was confused with the cars and buildings.

Table 4. The precision/recall and accuracy of TLS-Wang scene I [15] with different methods.

	Cars (%)	Vegetation (%)	Pedestrians (%)	Buildings (%)	Accuracy (%)
sLDA model	42.9/18.1	91.9/97.7	68.4/33.5	84.1/80.7	89.7
LDA model	52.9/45.4	95.4/98.3	82.9/62.7	89.9/86.7	93.4
Object-oriented decision tree	91.38/93.76	96.86/97.47	80.37/67.82	90.46/93.68	95.17
PointNet++ [41]	86.87/71.56	96.30/94.61	91.99/69.28	79.14/89.66	92.59
MSNet	75.07/89.75	98.68/99.32	91.86/82.5	99.46/96.64	98.24

Table 5. The precision/recall and accuracy of ALS-Zhang scene I [21] using different methods.

	Cars (%)	Vegetation (%)	Buildings (%)	Accuracy (%)
Guo et al. [47]	44.1/34.8	86.8/91.2	96.81/95.5	92.2
Zhang et al. [21]	53.9/60.5	94.7/94.5	98.1/97.7	95.5
PointNet++ [41]	38.49/1.21	98.43/96.13	80.99/96.13	95.16
MSNet	70.32/99.98	96.71/94.53	98.7/98.04	97.02

4.2. Genegralization Capability Analysis

This section primarily focuses on the generalization capability of the proposed MSNet. In our previous experiments, it was necessary to collect training samples before the classification step. However, to manually collect enough samples for each classification would be cumbersome and unacceptable. Therefore, the cross-scene generalization capability of the proposed MSNet,

which measures the applicability of the pre-trained MSNet over one scene to other scenes of point clouds, was also an important aspect for network assessment.

For the TLS point cloud, we tested the TLS-Wang scene I with two different MSNets, which were trained with the MLS-WHU and TLS-Wang scene II point clouds, respectively. The classification results are provided in Figure 13a,b. Although there were some incorrect classifications (marked in black circles), both achieved satisfactory classification results. The incorrect classifications were attributed to the lack of similar samples in the training step.

For the ALS point cloud, we tested the ALS-Zhang scene I with two different MSNets that were trained with the ALS-WHU and ALS-Zhang scene II point clouds, respectively. The classification results are shown in Figure 14a,b. Part of the cars were mistakenly classified, because the point density of WHU and Zhang et al. [21] were different (the ALS-Zhang point cloud was about three times denser than the WHU point cloud). Besides the density, the collected types of vegetation also varied with the experimental scenes, and led to classification errors regarding unknown vegetation types.

Besides the cross-scene tests of the same kind of point cloud, the ALS-Zhang scene I point cloud was also tested with the MSNet that was trained with the MLS-WHU data. In this case, the training and testing point clouds had totally different densities, perspectives, objects, and occlusions. As shown in Figure 14c, most of the buildings that were relatively large in size were correctly identified, while some small cars and piecemeal vegetation were not. Therefore, we concluded that the discriminative feature learned by the MSNet was sufficiently robust.

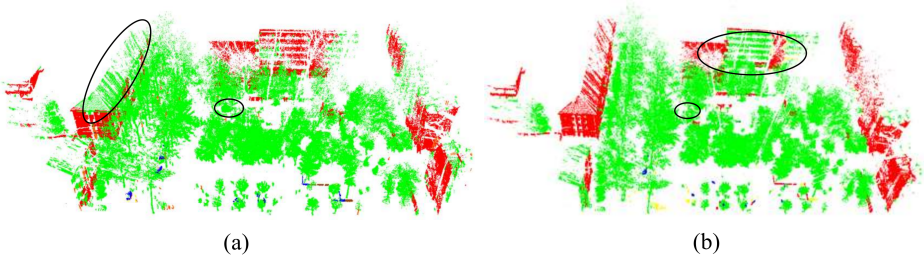


Figure 13. Classification results of TLS-Wang scene I using MSNet trained with other data sets. (**a**,**b**) are the classification results using MSNet trained with the MLS-WHU and the TLS-Wang scene II point cloud, respectively.

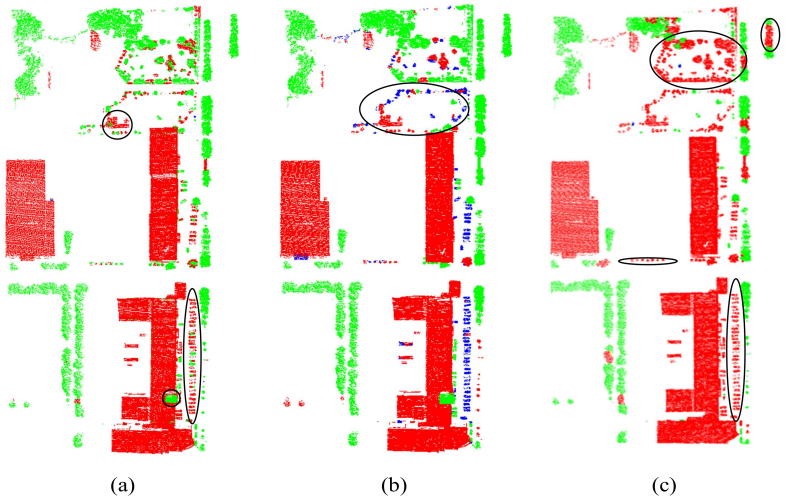


Figure 14. Classification results of ALS-Zhang scene I using MSNet trained with other data sets. (**a**–**c**) are the classification results using MSNet trained with the ALS-WHU, the ALS-Zhang scene II, and the MLS-WHU point cloud respectively.

Table 6 lists the quantitative precision/recall and accuracy of the aforementioned three tests. Generally, for similar scenes with equivalent densities, the pre-trained MSNet performed well. The accuracy of each of the tests was higher than 83%, and the best accuracy of approximate density was 92.74%. The accuracy with objects that were small in size, such as cars and pedestrians, was relatively lower, as small objects are more sensitive to noise and occlusions in point clouds. Due to the multi-scale voxelization and the weight sharing across different scales, similar context features are learned at different scales. It is beneficial to the generalization capability of the proposed method to classify point clouds of different resolutions such as MLS and ALS. However, for different resolutions, the size of voxels should be set according to the size of objects that are to be classified in the point cloud.

Table 6. The precision/recall and accuracy of different scenes with different models.

Data	Training Data for MSNet	Cars (%)	Vegetation (%)	Pedestrians (%)	Buildings (%)	Accuracy (%)
TLS-Wang scene I [15]	MLS-WHU point cloud	45.15/28.93	90.65/99.35	12.45/0.27	99.27/70.62	90.27
	TLS-Wang scene II [15]	73.47/30.37	91.99/99.78	70.88/23.21	98.84/78.57	92.74
ALS-Zhang scene I [21]	ALS-WHU point cloud	0.0/0.0	83.88/83.88	-	90.41/93.21	88.3
	ALS-Zhang scene II [21]	51.53/70.73	94.31/82.37	-	94.27/98.34	92.74
	MLS-WHU point cloud	0.0/0.0	98.76/55.92	-	80.95/99.98	83.8

- Represents that there is no such kind of object in current scene.

5. Conclusions

The method proposed in this paper provides an efficient point cloud classification approach, which consists of two complementary parts. In the first part, the point cloud is represented as multi-scale 3D voxels, and a corresponding MSNet is proposed to learn the multi-scale discriminative local features and predict the class label of each point. In the second part, the coarse classification results of MSNet are globally optimized using CRF with a spatial consistency constraint on the point cloud.

Compared with the existing point cloud feature extraction methods, which mainly focus on designing and extracting features subjectively, the feature extraction in our method is adaptive and learning-based. With the proposed multi-scale voxelization of MSNet, the multi-scale discriminative feature of a point cloud is adaptively extracted and fused to comprehensively characterize the local spatial context of each point in a concise way.

To address the MSNet classification inconsistency of one object cluster, which is caused by the point-wise class prediction, CRF with spatial consistency is constructed based on the graph of the point cloud to achieve a global optimization for all of the predicted class labels.

The experimental results show that the proposed method not only works well for MLS point clouds, it also achieved a much higher classification accuracy on ALS and TLS point clouds compared with the state-of-the-art methods, at 97.02% and 98.24%, respectively, thereby demonstrating the strong generalization capability of the proposed network for point cloud classification under complex urban environments.

However, the proposed solution also has its limitations. Although the multi-scale voxelization of point clouds substantially reduced the computation expense compared with a traditional CNN, further improvement is possible for the point-wise classification method. Therefore, a new convolution kernel with angle parameters, which can adopt the manifold structure and efficiently handle the point cloud within the linear computation, will be considered in our future work. Additional experiments on larger data sets are also a possibility in the future.

Acknowledgments: Work described in this paper is jointly supported by the Natural Science Foundation of China Project (No. 41671419, No. 41271431), the inter-disciplinary research program of Wuhan University (No. 2042017kf0204), the Collaborative Innovation Center of Geospatial Technology, the Fundamental Research Funds for the Central Universities (2042017KF0235).

Author Contributions: Lei Wang, Jie Shan and Yuchun Huang conceived and designed the framework of this research; Lei Wang and Liu He performed the experiments; Yuchun Huang and Jie Shan supervised this research; Lei Wang, Yuchun Huang and Jie Shan wrote the paper.

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

References

1. Nguyen, A.; Le, B. 3D point cloud segmentation: A survey. In Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics (RAM), Manila, Philippines, 12–15 November 2013; pp. 225–230.
2. Rabbani, T.; van den Heuvel, F.A.; Vosselman, G. Segmentation of point clouds using smoothness constraint. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2006**, *36*, 248–253.
3. Lin, X.; Zhang, J. Segmentation-based filtering of airborne LiDAR point clouds by progressive densification of terrain segments. *Remote Sens.* **2014**, *6*, 1294–1326. [CrossRef]
4. Awwad, T.M.; Zhu, Q.; Du, Z.; Zhang, Y. An improved segmentation approach for planar surfaces from unstructured 3D point clouds. *Photogramm. Rec.* **2010**, *25*, 5–23. [CrossRef]
5. Sampath, A.; Shan, J. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 1554–1567. [CrossRef]
6. Yan, J.; Shan, J.; Jiang, W. A global optimization approach to roof segmentation from airborne lidar point clouds. *ISPRS J. Photogramm. Remote Sens.* **2014**, *94*, 183–193. [CrossRef]
7. Golovinskiy, A.; Funkhouser, T. Min-cut based segmentation of point clouds. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops, Kyoto, Japan, 27 September–4 October 2009; pp. 39–46.
8. Shapovalov, R.; Velizhev, A.; Barinova, O. Non-Associative Markov Networks for 3D Point Cloud Classification. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2010**, 103–108.
9. 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.* **2012**, *3*, 263–268. [CrossRef]
10. Lozes, F.; Hidane, M.; Elmoataz, A.; Lezoray, O. Nonlocal segmentation of point clouds with graphs. In Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing (GlobalSIP 2013), Austin, TX, USA, 3–5 December 2013; pp. 459–462.
11. Li, Z.; Zhang, L.; Tong, X.; Du, B.; Wang, Y.; Zhang, L.; Zhang, Z.; Liu, H.; Mei, J.; Xing, X.; et al. A Three-Step Approach for TLS Point Cloud Classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 5412–5424. [CrossRef]
12. Niemeyer, J.; Rottensteiner, F.; Soergel, U.; Heipke, C. Hierarchical higher order crf for the classification of airborne lidar point clouds in urban areas. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 655–662. [CrossRef]
13. Kalogerakis, E.; Hertzmann, A.; Singh, K. Learning 3D mesh segmentation and labeling. *ACM Trans. Gr.* **2010**, *29*. [CrossRef]
14. Strîmbu, V.F.; Strîmbu, B.M. A graph-based segmentation algorithm for tree crown extraction using airborne LiDAR data. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 30–43. [CrossRef]
15. 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]
16. Niemeyer, J.; Rottensteiner, F.; Soergel, U. Contextual classification of lidar data and building object detection in urban areas. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 152–165. [CrossRef]
17. Weinmann, M.; Schmidt, A.; Mallet, C.; Hinz, S.; Rottensteiner, F.; Jutzi, B. Contextual classification of point cloud data by exploiting individual 3D neigbourhoods. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *2*, 271–278. [CrossRef]
18. Richter, R.; Behrens, M.; Döllner, J. Object class segmentation of massive 3D point clouds of urban areas using point cloud topology. *Int. J. Remote Sens.* **2013**, *34*, 8408–8424. [CrossRef]
19. Anand, A.; Koppula, H.S.; Joachims, T.; Saxena, A. Contextually guided semantic labeling and search for three-dimensional point clouds. *Int. J. Robot. Res.* **2013**, *32*, 19–34. [CrossRef]
20. Chen, G.; Maggioni, M. Multiscale geometric dictionaries for point-cloud data. In Proceedings of the International Conference on Sampling Theory and Applications, Singapore, 2–6 May 2011; pp. 1–4.
21. 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]

22. Hackel, T.; Wegner, J.D.; Schindler, K. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 177–184. [CrossRef]
23. Zhang, J.; Lin, X.; Ning, X. SVM-Based classification of segmented airborne LiDAR point clouds in urban areas. *Remote Sens.* **2013**, *5*, 3749–3775. [CrossRef]
24. Ghamisi, P.; Höfle, B. LiDAR Data Classification Using Extinction Profiles and a Composite Kernel Support Vector Machine. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 659–663. [CrossRef]
25. Lodha, S.K.; Fitzpatrick, D.M.; Helmbold, D.P. Aerial Lidar Data Classification using AdaBoost. In Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling, Montreal, QC, Canada, 21–23 August 2007; pp. 435–442. [CrossRef]
26. 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]
27. Niemeyer, J.; Rottensteiner, F.; Soergel, U. Classification of urban LiDAR data using conditional random field and random forests. In Proceedings of the Urban Remote Sensing Event, Sao Paulo, Brazil, 21–23 April 2013; Volume 856, pp. 139–142. [CrossRef]
28. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 4489–4497.
29. Rana, S.; Gaj, S.; Sur, A.; Bora, P.K. Detection of fake 3D video using CNN. In Proceedings of the 2016 IEEE 18th International Workshop on Multimedia Signal Processing (MMSP), Montreal, QC, Canada, 21–23 September 2016.
30. Maturana, D.; Scherer, S. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928. [CrossRef]
31. Li, Y.; Pirk, S.; Su, H.; Qi, C.R.; Guibas, L.J. FPNN: Field probing neural networks for 3D data. In *Advances in Neural Information Processing Systems*; NIPS: Barcelona, Spain, 2016; pp. 307–315.
32. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 945–953.
33. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D ShapeNets: A deep representation for volumetric shapes. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
34. Yu, Y.; Li, J.; Guan, H.; Jia, F.; Wang, C. Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 709–726. [CrossRef]
35. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef] [PubMed]
36. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
37. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *40*, 834–848. [CrossRef] [PubMed]
38. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1520–1528.
39. Riegler, G.; Ulusoy, A.O.; Geiger, A. OctNet: Learning Deep 3D Representations at High Resolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
40. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, 21–26 July 2017.
41. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space Supplementary Material. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 8 December 2017.

42. Huang, J.; You, S. Point cloud labeling using 3D Convolutional Neural Network. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancún, México, 4–8 December 2016; pp. 2670–2675.
43. Boykov, Y.; Veksler, O.; Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1222–1239. [CrossRef]
44. Boykov, Y.; Kolmogorov, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1124–1137. [CrossRef] [PubMed]
45. Kolmogorov, V.; Zabih, R. What Energy Functions Can Be Minimized via Graph Cuts? *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 147–159. [CrossRef] [PubMed]
46. Mcauliffe, J.D.; Blei, D.M. Supervised topic models. In *Advances in Neural Information Processing Systems*; NIPS: Vancouver, BC, Canada, 2008; pp. 121–128.
47. 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]