Discriminative-Dictionary-Learning-Based Multilevel Point-Cluster Features for ALS Point-Cloud Classification

Zhenxin Zhang, Liqiang Zhang, Xiaohua Tong, Bo Guo, Liang Zhang, and Xiaoyue Xing

***Abstract*—Efficient presentation and recognition of on-ground objects from airborne laser scanning (ALS) point clouds are a challenging task. In this paper, we propose an approach that combines a discriminative-dictionary-learning-based sparse coding and latent Dirichlet allocation (LDA) to generate multilevel point-cluster features for ALS point-cloud classification. Our method takes advantage of the labels of training data and each dictionary item to enforce discriminability in sparse coding during the dictionary learning process and more accurately further represent point-cluster features. The multipath AdaBoost classifiers with the hierarchical point-cluster features are trained, and we apply them to the classification of unknown points by the heritance of the recognition results under different paths. Experiments are performed on different ALS point clouds; the experimental results have shown that the extracted point-cluster features combined with the multipath classifiers can significantly enhance the classification accuracy, and they have demonstrated the superior performance of our method over other techniques in point-cloud classification.**

***Index Terms*—Airborne laser scanning (ALS) point clouds, classification, discriminative dictionary learning, point clusters, sparse coding.**

I. Introduction

HIGHLY dense point clouds acquired by airborne laser scanning (ALS) have become a very important way for understanding a ground scene. In this context, efficient classification of ALS point clouds is one of the most challenging tasks in photogrammetry and remote sensing fields due to their large varieties, the complex geometry, and visual appearance [1]. For example, the point density of the ALS point cloud in an urban environment is not uniform because of the varying distance between the objects and the scanner. Even worse, severe occlusion between objects yields incomplete point clouds and clutter background. Clearly, extraction of discriminative and robust object representations from the point clouds with noise and clutters is very critical to achieve high classification precision. However, designing features for recognizing visual objects is time-consuming and requires a deep understanding of domain knowledge [2], [3]. The often used image feature descriptors such as histograms of oriented gradients (HOGs) [4] and scale-invariant feature transformation (SIFT) [5] are usually high dimensional. For a large scene, these descriptors are hard to be directly applied in the classifiers due to intensive computation and sensitivity to noise.

Currently, the feature representations for ALS point clouds are generally divided into the point-based [6], voxel-based [7]–[11], and object-based levels [12], [13]. Except for the features, classifiers also affect the classification performance. As noted in [14], the performance of the classifiers is dependent on the data. In this paper, we focus on obtaining discriminative cluster-based features to characterize the shape of objects in the ALS point clouds of urban environments, and then we utilize the multipath AdaBoost classifiers to enhance the classification results.

Sparse-representation-based and dictionary-learning-based methods have been successfully applied to image classification. For example, inspired by the idea behind the learning vector quantization [15], a dictionary learning algorithm is presented for the classification of hyperspectral images. The dictionary is optimized by minimizing the hinge loss of residual difference between competing classes [16]. Instead of vector quantization, image representation is computed based on the sparse codes of local descriptors such as HOG and SIFT. Sparse coding approximates an input signal (e.g., point clouds) by a linear combination of the bases in a dictionary. To find an appropriate set of bases, i.e., a dictionary, many efforts are devoted to dictionary learning. Learning the dictionary from the training samples for discriminative sparse coding can achieve impressive classification results [1], [17]–[20]. Some methods, such as the learned iterative shrinkage–thresholding algorithm [21] and the label consistent K-SVD (LC-KSVD) algorithm [22], have been developed to provide efficient optimization algorithms for sparse coding. However, the complexity of these algorithms grows dramatically when the number of categories is large;

Manuscript received December 28, 2015; revised April 21, 2016 and June 12, 2016; accepted August 2, 2016. Date of publication August 31, 2016; date of current version September 30, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 41371324, Grant 41325005, and Grant 41501499 and the Shenzhen Scientific Research and Development Funding Program (JCYJ20150625102531697). *(Corresponding author: Liqiang Zhang.)*

Z. Zhang, L. Zhang, L. Zhang, and X. Xing are with the State Key Laboratory of Remote Sensing Science, School of Geography, Beijing Normal University, Beijing 100875, China (e-mail: zhenxin066@163.com; zhanglq@bnu.edu.cn; yammer_0303@126.com; xyxing@mail.bnu.edu.cn).

X. Tong is with the School of Surveying and Geo-informatics, Tongji University, Shanghai 200092, China (e-mail: xhtong@tongji.edu.cn).

B. Guo is with the Key Laboratory for Geo-Environment Monitoring of Coastal Zone of the National Administration of Surveying, Mapping and GeoInformation and Shenzhen Key Laboratory of Spatial Smart Sensing, Shenzhen University, Shenzhen 518060, China (e-mail: guobo.szu@qq.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2016.2599163

thus, determination of sparse codes from large dictionaries is computationally expensive [22].

As noted in [23], utilizing the hierarchy to guide the model learning can bring improvement in classification efficiency and accuracy. In hierarchical data structures, determination of local neighborhoods has been conducted either with respect to the absolute size [24] or with respect to the scale parameter [25]. In order to capture the richness of data, a multiscale and hierarchical framework was presented to recognize terrestrial laser scanning (TLS) point clouds of cluttered urban scenes [7]. In this framework, the TLS point cloud is first resampled into different scales. Then, the resampled data set of each scale is aggregated into several HPCs. The point-cluster-based feature at each level is obtained by the LDA integrated with the bag of words (BoW). We know the BoW discards the spatial order of the local descriptors; thus, it limits the descriptive power to the original data. Another methodology that has been used in image classification is the super-pixel approach [26], [27], which aggregates the local neighboring points that have similar colors and texture statistics.

Brodu and Lague [28] classified TLS point clouds using a multiscale local dimensionality feature. The feature allows the best separation of different classes. By combining various scales, the method performs better than a single scale analysis and is robust to missing TLS data. Xu *et al.* [29] employed three types of entities, including single points, planar segments, and segments obtained by mean-shift segmentation, to classify the point clouds. In the above two methods, different scales are used for obtaining the context of the point cloud and the shape of the objects. In [30], the heterogeneous features of urban objects are extracted from spectral images and LiDAR data. Then, the features and multiple-kernel learning classifier based on two levels are utilized to classify urban scenes. A rule-based hierarchical semantic classification scheme that utilizes spectral information, geometry, and topology-related features was developed in [31]. Because multilevel structures are capable of representing the semantic intercorrelation or visual similarity among categories [23], hierarchical dictionary learning models are employed to enhance the classification performance [32]–[34]. To avoid losing all information about the spatial layout of the features, Zhou *et al.* [35] incorporated a multiresolution representation into a bag-of-features model. They partitioned an image into multiple resolutions and extracted local features from each of the multiresolution images with dense regions. Then, the representations of different-resolution channels were combined to reach a final decision using a support-vector-machine (SVM) classifier. Multilayer sparse coding networks [36]–[38] have been proposed to build feature hierarchies layer by layer using sparse codes and spatial pooling. Each layer in these networks contains a coding step and a pooling step. A dictionary is learned at each coding step which serves as a codebook for obtaining sparse codes from image patches or pooled features. Spatial pooling schemes group the sparse codes from adjacent blocks into common entities. The pooled sparse codes from one layer serve as the input to the next layer.

In this paper, we aim to investigate the shape features of objects and apply them to ALS point-cloud classification, i.e., the input is the ALS point cloud of an urban environment, and the output is the classification result of each object class (such as buildings and trees). Point-based features can represent the information of relatively small regions. Conversely, the features derived from large regions (e.g., cluster-based features) can capture more specific information. If we construct good features for each 3-D point, and extract discriminative features of the point clusters from the point-based features, the recognition ability to ALS point clouds can be significantly enhanced. Based on the above observations, we proposed a method for constructing the features of the hierarchical point clusters (HPCs).

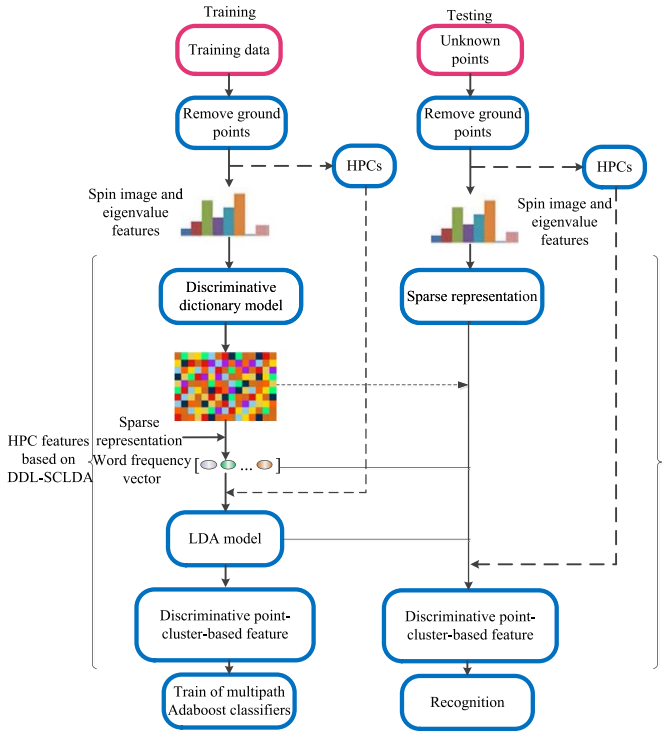


Fig. 1. Process for classifying ALS point clouds.

In this method, a discriminative-dictionary-learning-based sparse coding combined with LDA (DDL-SCLDA) is employed to derive the point-cluster-based features at each level from the point-based features. As illustrated in Fig. 1, in the training phase, the proposed method first aggregates the input ALS point cloud into multilevel point-cluster sets. Then, the discriminative dictionary learning model is built to learn a discriminative dictionary of the point-based features. In addition to using class labels of the point-based features in the training data, we associate label information with each dictionary item to enforce discriminability in sparse coding during the dictionary learning process. The optimal solution is efficiently solved by the K-SVD algorithm [41] to obtain the discriminative dictionary. The discriminative-dictionary learning-based sparse coding is integrated into the LDA to describe the point-cluster features of different levels. Finally, we exploit the hierarchical point-cluster features to train multipath AdaBoost classifiers, and the unknown points are classified by the heritance of recognition results under different paths.

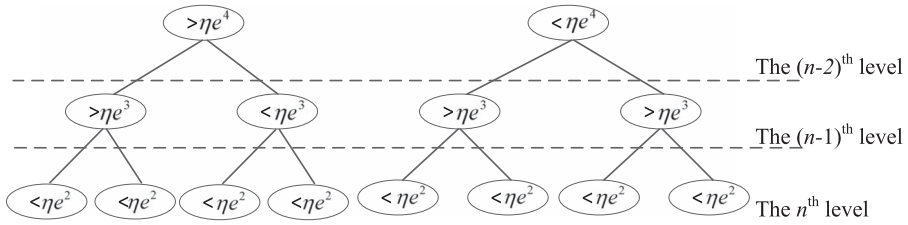


Fig. 2. Three-level point clusters.

The main contributions of this paper are summarized as follows.

1) A discriminative dictionary learning model with the label consistency constraint is introduced to represent the point-based features.
2) The DDL-SCLDA is proposed to construct the features of the point clusters from the point-based features.
3) The point-cluster features of different levels are used to train the multipath AdaBoost classifiers, and the final classification results of unknown points are obtained by means of the hierarchical category heritance under different paths.

II. Methodology

Here, we will first create hierarchical point-cluster sets from the input ALS point cloud. Afterward, we obtain the point-based features through discriminative dictionary learning. Next, construct the feature of each point cluster by using the DDL-SCLDA. Finally, the point-cloud classification is generated by employing the multipath AdaBoost classifiers on the point-cluster-based features of the HPCs.

A. Generation of HPCs

The process for generating HPCs consists of the following steps.

1) The terrain points and the isolated points are first removed using the method described in [43]. The removal of the terrain points helps to determine the connectivity of objects.
2) Considering that the nonterrain points are often unorganized and lack inherent structure, we utilize an undirected graph to organize them. Therefore, we search the k_1 closest points of each point and connect the point with its k_1 closest points by edges. In this way, an undirected graph $G(\mathbf{V}, \mathbf{E})$ is generated, where $\mathbf{V}$ is the point set, and $\mathbf{V}$ is the set of the corresponding edges. The Euclidean distance between two connected points is taken as the weight of the edge. After G is generated, all of the connected components of G can be found.
3) Because objects are often close together in cluttered urban scenes, a connected component can contain more than one object. In a connected component, a local maximum point may represent the top of an object. To further break the connected component into smaller pieces so that single objects can be isolated, a moving window algorithm is applied to search the local maximum points in a 2-D raster image, which represents the heights of the points in the connected component. The raster value is the maximum height of the points in each raster. When the local maximum points are found, the graph cut [44] is employed to segment the connected component, and the local maximum points are taken as seeds. After the graph cut is performed, the connected component is divided into several point clusters.
4) Each of the point clusters acquired by the steps 2 and 3 may still contain more than one object. To achieve discriminative cluster features, a cluster should contain only one single object (or a part of it). Furthermore, the distribution of points on the object surface should be as even as possible. Motivated by the fact that the normalized cut [45] can aggregate the points with uniform distribution into one cluster, it has been employed here to partition a large point cluster into two new clusters under the condition that the number of points in the cluster is larger than a predefined threshold δ_m. To ensure that a point cluster contains enough spatial or shape information, we define δ_m at different levels $\delta_m = \eta e^x$, where η is a parameter, and x is an integer related with the level number. Thus, the input point cloud is segmented into multilevel point-cluster sets.

The point clusters with the smallest size, i.e., $x = 2$, are the lowest level, namely, the nth level. In the jth $(j < n)$ level, $x = n + 2 - j$. Fig. 2 shows an example of the HPCs with three levels.

B. Discriminative Dictionary Learning

Since the input point cloud has been split into multilevels, the point-based features are correspondingly extracted in each level. The set of points that are the k closest neighborhoods of a point p is defined as the support region of p [7]. In this paper, we extract the feature of p in its three support regions with different sizes. In each support region, we compute the spin image descriptor [39] and eigenvalues [40] of p. A spin image can capture the majority of local shape information presented in a 3-D scene. It is a 2-D parameter space histogram. Each 3-D point can be projected onto a 2-D space with the x-axis and y-axis. We take the normal vector of a point as the rotation axis and set the size to 3×4 bins. Therefore, the spin image of p has 12 values. The 12 values of the spin image and the six values of the eigenvalues are combined into an 18-dimensional vector. The 54-dimensional vector obtained

from the three support regions is taken as the feature of p. Next, we will focus on constructing a discriminative dictionary learning model for deriving the features of the point clusters.

As the supervised learning approach has been shown beneficial to dictionary learning [18], we propose a supervised formulation for extracting discriminative dictionary to encode the point-based features. Assume the features of the training samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and $\mathbf{x}_i$ denotes the feature of the ith point. Given the dictionary $\mathbf{D}$, we aim to leverage the supervised information (i.e., labels) of the input feature to learn a discriminative dictionary. Each dictionary item is obtained to represent a subset of the training data from a single class, and then the dictionary item is associated with a particular label. Hence, there is an explicit correspondence between the dictionary items and the labels in our approach.

Next, we will construct a discriminative dictionary learning model. The model integrates a label consistency regularization term into the objective function. It is optimized by the K-SVD algorithm [41], and sparse representation of each point feature is obtained by using feature sign method [42] under the above optimized discriminative dictionary.

1) Discriminative Dictionary Learning Model: Similar to [22], we also apply the discriminative dictionary learning method to get a discriminative dictionary.

Assume $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_q]$ is the sparse representation of the input point features $\mathbf{X}$. The classification performance depends on the discriminability of the input sparse codes $\mathbf{U}$, which is closely related with the saliency of the dictionary $\mathbf{D}$, where $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K]$. For obtaining a discriminative $\mathbf{D}$, an objective function for extracting the dictionary is defined as

$$\begin{aligned} \langle \mathbf{D}, \mathbf{A}, \mathbf{U} \rangle = \underset{\mathbf{D},\mathbf{A},\mathbf{U}}{\arg\min} \left\{ \|\mathbf{X} - \mathbf{D}\mathbf{U}\|_F^2 + \alpha \|\mathbf{Q} - \mathbf{A}\mathbf{U}\|_F^2 \right\} \\ \text{s.t.} \quad \forall i, \|\mathbf{u}_i\|_0 \le T \end{aligned} \tag{1}$$

where $\mathbf{A}$ is a linear transformation matrix. α is a weight that controls the relative contribution between the reconstruction and label consistency regularization. T is a sparsity constraint factor (for each decomposed signal $\mathbf{u}_i$, the number of the nonzero items is less than T). $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_N] \in R^{K \times N}$ are the "discriminative" sparse codes of the input features $\mathbf{X}$ for classification. $\mathbf{q}_i = [q_i^1, \ldots, q_i^K]^t = [0 \ldots 1, 1, \ldots, 0]^t \in R^K$ is a "discriminative" sparse code corresponding to an input signal $\mathbf{x}_i$ if the nonzero values of $\mathbf{q}_i$ occur at those indexes where the input signal $\mathbf{x}_i$ and the dictionary item $\mathbf{d}_k$ share the same label. For example, if $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_5]$, and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_6]$, where $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{d}_2$ are from class 1, $\mathbf{x}_3$, $\mathbf{x}_4$, $\mathbf{d}_1$, and $\mathbf{d}_3$ are from class 2, and $\mathbf{x}_5$, $\mathbf{x}_6$, $\mathbf{d}_4$, and $\mathbf{d}_5$ are from class 3, $\mathbf{Q}$ can be defined as

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

We perform a linear transformation $g(\mathbf{u}, \mathbf{A}) = \mathbf{A}\mathbf{u}$, which makes the original sparse code $\mathbf{u}$ the most discriminative in the sparse feature space R^K. The term $\|\mathbf{Q} - \mathbf{A}\mathbf{U}\|_2^2$ represents the discriminative sparse code error, which enforces the transformed sparse codes $\mathbf{A}\mathbf{U}$ approximate the discriminative sparse codes $\mathbf{Q}$. It forces the features from the same class to have very similar sparse representations such as encouraging label consistency in the resulting sparse codes.

2) Initialization of the Discriminative Dictionary Learning Model: The parameters $\mathbf{D}^{(0)}$ and $\mathbf{A}^{(0)}$ are initialized for the discriminative dictionary learning model. For $\mathbf{D}^{(0)}$, we use several iterations of K-SVD [41] within each class, and then combine all the outputs (i.e., dictionary items learning from each class) of each K-SVD. The label of each dictionary item $\mathbf{d}^k$ is then initialized based on the corresponding class and remains fixed during the dictionary learning process. Words are uniformly allocated to each class with the number of the elements proportional to the dictionary size.

We employ the multivariate ridge regression model [46] to initialize $\mathbf{A}^{(0)}$, under the condition of the quadratic loss and Frobenius norm regularization. The model is as follows:

$$\mathbf{A} = \underset{\mathbf{A}}{\arg\min} \left\{ \|\mathbf{Q} - \mathbf{A}\mathbf{U}\|_F^2 + \beta \|\mathbf{A}\|_F^2 \right\} \tag{2}$$

which obtains the following solution:

$$\mathbf{A} = \mathbf{Q}\mathbf{U}^t (\mathbf{U}\mathbf{U}^t + \beta I)^{-1}. \tag{3}$$

Given the initialized $\mathbf{D}^{(0)}$, we apply the K-SVD algorithm to compute the sparse codes $\mathbf{U}$ of the training feature $\mathbf{X}$. Then, $\mathbf{U}$ is used to compute the initial $\mathbf{A}^{(0)}$.

3) Optimization: We use the K-SVD algorithm to find the optimal solution for all parameters simultaneously. Equation (4) can be rewritten as

$$\begin{aligned} \langle \mathbf{D}, \mathbf{A}, \mathbf{U} \rangle = \underset{\mathbf{D},\mathbf{A},\mathbf{U}}{\arg\min} \left\| \begin{pmatrix} \mathbf{X} \\ \sqrt{\alpha}\mathbf{Q} \end{pmatrix} - \begin{pmatrix} \mathbf{D} \\ \sqrt{\alpha}\mathbf{A} \end{pmatrix} \mathbf{U} \right\|_F^2 \\ \text{s.t.} \quad \forall i, \|\mathbf{u}_i\|_0 \le T. \end{aligned} \tag{4}$$

Let $\mathbf{X}_{\text{new}} = (\mathbf{X}^t, \sqrt{\alpha}\mathbf{Q}^t)^t$ and $\mathbf{D}_{\text{new}} = (\mathbf{D}^t, \sqrt{\alpha}\mathbf{A}^t)^t$. The matrix $\mathbf{D}_{\text{new}}$ is L_2 normalized columnwise. The optimization of (4) is equivalent to solving the following problems:

$$\begin{aligned} \langle \mathbf{D}_{\text{new}}, \mathbf{U} \rangle = \underset{\mathbf{D}_{\text{new}},\mathbf{U}}{\arg\min} \left\{ \|\mathbf{X}_{\text{new}} - \mathbf{D}_{\text{new}}\mathbf{U}\|_F^2 \right\} \\ \text{s.t.} \quad \forall i, \|\mathbf{u}_i\|_0 \le T. \end{aligned} \tag{5}$$

This is the standard problem that K-SVD [41] solves. Following K-SVD, $\mathbf{d}_k$ and its corresponding coefficients, the kth row in $\mathbf{U}$, which is denoted $\mathbf{u}_R^k$, are updated at a time. Let $\mathbf{E}_k = (\mathbf{X} - \sum_{j \ne k} \mathbf{d}_j \mathbf{u}_R^j)$, and $\widetilde{\mathbf{u}}_R^k$ and $\widetilde{\mathbf{E}}_k$ denote the result of discarding the zero entries in $\mathbf{u}_R^k$ and $\mathbf{E}_k$, respectively. $\mathbf{d}_k$ and $\mathbf{u}_R^k$ can be computed by

$$\left\langle \mathbf{d}_k, \tilde{\mathbf{u}}_R^k \right\rangle = \underset{\mathbf{d}_k, \tilde{\mathbf{u}}_R^k}{\arg\min} \left\{ \left\| \tilde{\mathbf{E}}_k - \mathbf{d}_k \tilde{\mathbf{u}}_R^k \right\|_F^2 \right\}. \tag{6}$$

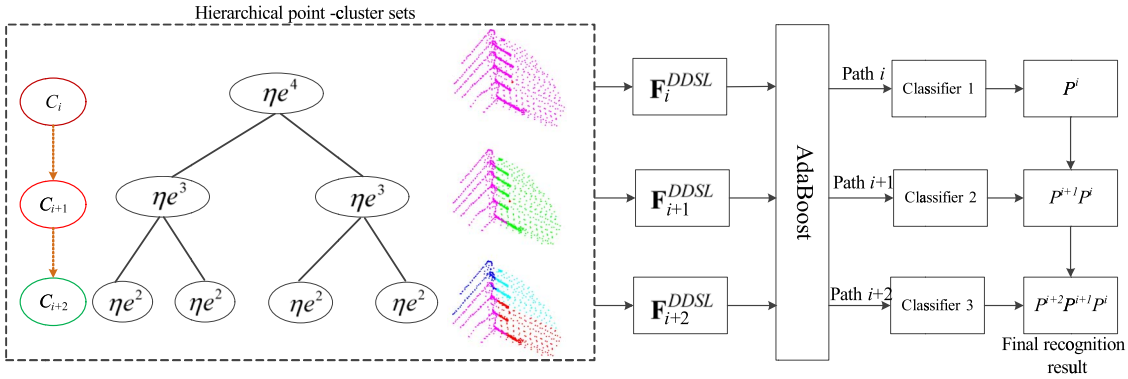


Fig. 3. Labeling unknown point clusters through multipath AdaBoost classifiers and inheritance.

An SVD operation is performed for $\tilde{\mathbf{E}}_k$, i.e., $\mathbf{B}\sum\mathbf{V}^t = \mathrm{SVD}(\tilde{\mathbf{E}}_k)$. Then, $\mathbf{d}_k$ and $\mathbf{u}_R^k$ are computed as

$$\mathbf{d}_k = \mathbf{B}(:,1) \qquad \mathbf{u}_R^k = \sum(1,1)\mathbf{V}(:,1). \tag{7}$$

The nonzero values in $\mathbf{u}_R^k$ are replaced with $\tilde{\mathbf{u}}_R^k$.

$\mathbf{D}$ and $\mathbf{A}$ are not simply obtained from $\mathbf{D}_{\mathrm{new}}$ since $\mathbf{D}_{\mathrm{new}}$ is the L_2-norm normalization constraint in the above algorithm, such as $\forall\, j,\ \|(\mathbf{d}_j^t, \sqrt{\alpha}\mathbf{a}_j^t)^r\|_2 = 1$. Then, the desired $\hat{\mathbf{D}}$ and $\hat{\mathbf{A}}$ can be calculated using

$$\hat{\mathbf{D}} = \left\{\frac{\mathbf{d}_1}{\|\mathbf{d}_1\|_2} \cdots \frac{\mathbf{d}_K}{\|\mathbf{d}_K\|_2}\right\} \qquad \hat{\mathbf{A}} = \left\{\frac{\mathbf{a}_1}{\|\mathbf{d}_1\|_2} \cdots \frac{\mathbf{a}_K}{\|\mathbf{d}_K\|_2}\right\}. \tag{8}$$

4) Sparse Representation: The following is applied to perform the sparse representation, where $\mathbf{D}$ is calculated from (1):

$$\mathbf{u}_i(\mathbf{x}_i, \mathbf{D}) = \arg\min_{\mathbf{V},\mathbf{U}} \left(\|\mathbf{x}_i - \mathbf{D}\mathbf{u}_i\|_2^2 + \lambda\|\mathbf{u}_i\|_1\right)$$
$$\text{s.t. } \|\mathbf{d}_m\|_2 \le 1 \quad \forall\, m = 1, 2, \ldots, K \tag{9}$$

where λ controls the sparsity. To make the sparse representation $\mathbf{u}$ more discriminative, the discriminative dictionary is used to compute $\mathbf{u}$. Sparse representation $\mathbf{u}$ is calculated by using the feature sign method [42].

C. Features of Point Clusters and Classification

The features of the point clusters take the spatial relationship among points and the shape of the point clusters at each level into account. To achieve this, the point-based features are first extracted. Next, the DDL-SCLDA is applied to construct the features of the point clusters from the point-based features. We define each point cluster as a document, and all point-cluster sets make up a document set. The dictionary derived by discriminative-dictionary-based sparse coding is taken as the dictionary of the LDA. The features of each 3-D point in a point cluster are set as a basic unit, and they are encoded by sparse coding. The appearance frequency of each word in a point cluster is computed to generate a word frequency vector with length K (K is the number of words in the dictionary).

1) Features of Point Clusters: Set the vocabulary $\mathbf{w} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K]$, with $\mathbf{w}$ having the same meaning as $\mathbf{D}$ in (1). The probability of the ith word in the sth document can be derived by

$$p(\mathbf{w}_i|\boldsymbol{\theta}, \boldsymbol{\beta}) = \sum_{k=1}^{r} u_i^k \tag{10}$$

where u_i^k is calculated by (9). r is the point number of the sth document. $\boldsymbol{\beta}$ is an $M \times N$ matrix, and M is the number of the topics. $\boldsymbol{\theta}$ is an M-dimensional Dirichlet random variable [47], namely, $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_M]$. θ_i represents the probability of the ith topic. Set the latent topic set $\mathbf{z} = [z_1, z_2, \ldots, z_N]$, and $\boldsymbol{\gamma}$ is a Dirichlet parameter. We build the LDA model [45] as follows:

$$p(\mathbf{w}|\boldsymbol{\gamma}, \boldsymbol{\beta}) = \frac{\Gamma(\sum_i \gamma_i)}{\prod_i (\Gamma(\gamma_i))} \int \left(\prod_{i=1}^{M} \theta_i^{\gamma_i - 1}\right) \times \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(\mathbf{w}_n|z_n, \boldsymbol{\beta})\right) d\theta. \tag{11}$$

In the training phase, $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are obtained using the expectation–maximization algorithm because $\mathbf{w}$ is taken as a variable. $\boldsymbol{\theta}$ and $\mathbf{z}$ are hidden variables. Finally, each latent topic probability in the point cluster can be achieved. The feature of the sth point cluster of the tth level is defined as $\mathbf{F}_{C_t^s}^{\mathrm{DDSL}}$, then

$$\mathbf{F}_{C_t^s}^{\mathrm{DDSL}} = [\theta_1, \theta_2, \ldots, \theta_M]. \tag{12}$$

2) Multipath AdaBoost Classifiers: We have extracted the feature of each point cluster. Next, we design multipath AdaBoost classifiers to classify the point-cluster sets into different categories.

The training data are clustered into the HPCs, and DDL-SCLDA features are extracted. After all the features of the HPCs are derived, the one-versus-all AdaBoost classifiers of each path are trained, and different levels represent different paths (e.g., see Fig. 3). We set three categories to be classified as an example. For n point clusters, there are correspondingly $3n$ AdaBoost classifiers. The AdaBoost classifiers of each path are obtained in the training process. They are applied to classify the unlabeled point clouds. The unknown point-cluster sets are labeled by jointly determining different classifiers of the

TABLE I
EXPERIMENTAL DATA SETS

	Point number of the training data			Point number of the test data		
	Trees	Buildings	Cars	Trees	Buildings	Cars
Scene I	68,802	37,128	5,380	214,122	200,572	7,816
Scene II	12,895	184,732		21,393	495,129	

multipath. Following [27], the probability of assigning a label l_i to a specific cluster is mathematically expressed as

$$P_{\mathrm{num}}(l_i, \mathbf{F}^{\mathrm{DDSL}}) = \frac{\exp\left(H_{\mathrm{num}}(l_i, \mathbf{F}^{\mathrm{DDSL}})\right)}{\sum_i \exp\left(H_{\mathrm{num}}(l_i, \mathbf{F}^{\mathrm{DDSL}})\right)} \tag{13}$$

where $\mathbf{F}^{\mathrm{DDSL}}$ is the feature of each point cluster, num is an integer $(1 \leqslant num \leqslant s)$, and $H_{\mathrm{num}}(l_{\mathrm{i}}, \mathbf{F}^{\mathrm{DDSL}})$ is the output of the AdaBoost classifier for l_i.

During the training process, the training data are manually labeled, and each cluster only contains one specific object category. In the generalization process, the lowest point cluster only includes one or a part of an object, and the point clusters in each of the other levels may contain more than one object. Therefore, we only label the point-cluster set of the lowest level. The point cluster and its upper level of point clusters contain different point-cluster-based features; therefore, the unknown point-cluster sets are labeled by jointing the probabilities of assigning a label over point-cluster sets of different paths. The point clusters of the current level inherit the recognizing result of the previous level. As shown in Fig. 3, the probability of labeling l_i to a cluster C_i in a point-cluster set of the ith path is P^i. The probability of labeling l_i to a point cluster C_{i+1} in a point-cluster set of the $(i+1)$th path is P^{i+1}, and eventually, the classification result of C_{i+1} is $P^i \times P^{i+1}$, which is inherited from the probability of C_i. In the same way, the probability $P^i \times P^{i+1} \times P^{i+2}$ of the fine point cluster C_{i+2} belonging to each category can be obtained. The final probability for labeling l_i to a cluster in a point-cluster set can be mathematically obtained as

$$P_n^j(l_i) = \prod_{m=1}^{n} P^{m,\mathrm{num}}(l_i, \mathbf{F}^{\mathrm{DDSL}}) \tag{14}$$

where n denotes the number of paths, P_n^j denotes the probability of the jth point cluster attributing to the l_i category, and $P^{m,\mathrm{num}}$ denotes the probability of the mth point-cluster sets in the numth path attributing to the l_i category. Finally, all point clusters in the lowest level are labeled by the highest probability of the labels.

III. EXPERIMENTAL RESULTS

To validate the performance of our method, we perform both qualitative and quantitative evaluations on the ALS point clouds of two urban scenes.

TABLE II
FREE PARAMETERS RELATED TO OUR METHOD

Definition	Notation	Location
The number of words in **D**	K	Eq. (1)
Sparsity constraint factor	T	Eq. (1)
Weight of label consistency regularization	α	Eq. (1)
The neighborhood of a point	k	Section II-B
The number of multi-levels	n	Section II-A
The parameter of sparse representation	λ	Eq. (9)
Number of latent topics	N	Eq. (11)
Controlling parameter of point cluster generation threshold	η	Section II-A

A. Experimental Data Sets

We have captured the ALS point clouds of two scenes: One is the scene (Scene I) in Tianjin, China, and the other one is the scene (Scene II) in Toronto, Canada. Scene I represents residential area where buildings, trees, and cars exist. The point cloud of this scene was acquired in August 2010 by a Leica ALS50 system with a mean flying height of 500 m above the ground and a 45° field of view. The point density is approximately 20–30 points/m^2. The data set of Scene II was provided by the ISPRS Test Project on Urban Classification and 3-D Building Reconstruction [48]. It was acquired by Optech, Inc., and the point density is approximately 5–10 points/m^2. Scene II is a commercial district where there are many skyscrapers and a small number of trees. The related information is listed in Table I.

The algorithm runs on a computer with an Intel Core i7-4770K processor at 3.40 GHz and 8-GB RAM. It took approximately about 20.5 min to learn the DDL-SCLDA models and AdaBoost classifiers. It took approximately 10.8 and 15.2 min to classify the point clouds for Scenes I and II, respectively. During the processing, feature extraction and sparse representation cost about 71% of the whole computation time. However, most of the steps are parallelizable. Therefore, they can be implemented by using a parallel scheme to reduce the time consumption. In the experiment, the number of the weak classifiers in the AdaBoost is 2, the maximum possible depth of the tree is 2, and the weight trimming ratio is 0.95. To clearly show all free parameters related to our method, we list them in Table II.

TABLE III
MAIN CHARACTERISTICS OF THE FOUR METHODS

Methods	Feature representation	Extracting dictionary method	Feature representation	Point-cluster based?	Hierarchical point-cluster feature?
Our Method	DDL-SCLDA of point-cluster sets	K-SVD dictionary	Discriminative dictionary learning-based sparse representation	Yes	Yes
Method I	LDA of point-cluster sets	k-means method	Vector quantization	Yes	Yes
Method II	Point-based	No dictionary	No dictionary	No	No
Method III	Point-based	No dictionary	No dictionary	No	No

B. Comparisons With Other Methods

To validate the performance of our method, we compare it with other three methods. The first method (Method I) was the one described in [7]. This method uses a combination of BoW and LDA of point-cluster sets to classify the unknown data. The vocabulary is extracted using the k-means, and each point-based feature is represented using vector quantization. The second method (Method II) uses the point-based features. It directly applies the point-based features to classify point clouds without aggregating the point-based features into point clusters. The third method (Method III) is the one described in [48]. In this method, each point is associated with a set of defined features derived using geometric, multireturn, and intensity information, and features are selected using JointBoost to evaluate their correlations. In our method, the values of the used parameters are as follows: $K = 512$, $T = 30$, $\alpha = 16$, $k = 30$, $n = 4$, $\lambda = 0.15$, and $N = 10$. Table III shows the main characteristics of our method and the other three methods.

As shown in Fig. 4, the whole training data set taken from the scenes contains 308 937 points including 81 697 tree points, 221 860 building points, and 5380 car points.

Precision/recall is used to represent the classification quality. High precision means that an algorithm returned substantially more relevant results than irrelevant results, whereas high recall means that an algorithm returned most of the relevant results. Table IV shows precision/recall and accuracy of different methods in the learning stage. As shown in Table IV, the precision and recall of the classification results obtained by our method are the highest in classifying the three specific object classes. The classification precision of the cars obtained using our method is much higher than obtained by other three methods in Scenes I and II. Therefore, the features of point clusters generated by the DDL-SCLDA can better describe characteristics of the on-ground objects in the training data.

The classification qualities of the unknown data of the two scenes by using the above four methods have been tested. As shown in Table V, the classification results of most categories obtained by our method have higher accuracy. Except for a few points of the building corners and cars that are difficult to be distinguished, most of the points are correctly recognized by our method. The precision and recall of the cars obtained by our method are higher than the other three methods. Moreover, our method outperforms the other three methods in terms of the final classification accuracy levels.

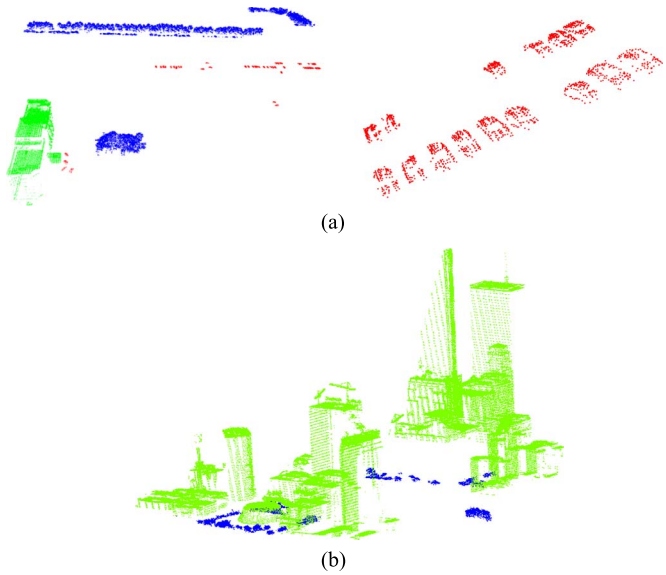


Fig. 4. Training data. Blue points represent trees, green points are buildings, and red points are cars. (a) Part of the training data set obtained from Scene I. (b) Training data set obtained from Scene II.

TABLE IV
PRECISION/RECALL AND ACCURACY OF DIFFERENT METHODS IN THE LEARNING STAGE

	Scene	Trees (%)	Buildings (%)	Cars (%)	Accuracy (%)
Our method	I	97.2/96.8	93.9/95.4	94.8/88.8	96.0
	II	98.3/82.4	98.4/99.9		98.8
Method I	I	96.7/95.2	94.9/95.4	72.5/84.9	94.8
	II	95.2/80.5	98.7/99.7		98.5
Method II	I	89.7/95.6	92.6/84.5	79.1/59.5	90.2
	II	89.4/73.3	98.2/99.4		97.7
Method III	I	95.9/97.7	96.1/96.5	85.7/63.5	95.6
	II	73.7/69.0	97.9/98.33		96.5

TABLE V
PRECISION/RECALL AND ACCURACY OF THE CLASSIFICATION RESULTS

Scene I	Trees (%)	Buildings (%)	Cars (%)	Accuracy (%)
Our method	93.1/**96.0**	**95.2**/**92.6**	**73.3**/62.2	**93.7**
Method I	**94.8**/93.8	93.5/92.3	41.2/**66.7**	92.6
Method II	85.7/92.9	92.0/83.8	56.9/54.7	87.9
Method III	89.7/**98.1**	**97.9**/89.1	65.2/46.6	92.9
Scene II	Trees (%)	Buildings (%)		Accuracy (%)
Our method	**93.0**/**86.8**	**99.4**/**99.7**		**99.2**
Method I	86.5/61.2	98.3/99.6		98.0
Method II	52.3/56.4	98.1/97.8		96.1
Method III	69.3/63.6	98.5/98.8		97.4

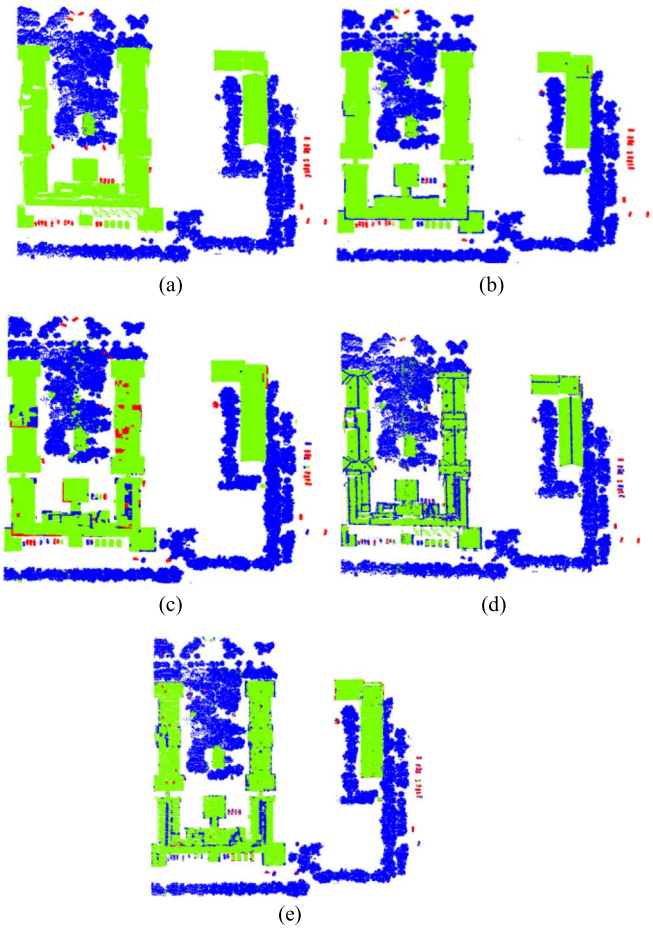


Fig. 5. Classification results obtained by different methods in a part of Scene I. (a) Ground truth. (b) Classification results obtained by using our method. (c) Classification results obtained by using Method I. (d) Classification results obtained by using Method II. (e) Classification results obtained by using Method III. The points on trees, buildings, and cars are colored in blue, green and red, respectively. Among the four methods, our method can correctly classify most of the points, and our method outperforms the other three methods.

Figs. 5 and 6 visually show the classification results. Compared with the other classification results, our method can more accurately classify most of the points into the correct categories, such as the recognition of the points in dashed boxes 1, 2, and 3 of Fig. 5 and in dashed boxes 1 and 2 of Fig. 6. The classification result of Method II is the worst. The reason is that the point-based features do not well distinguish the objects as the basic cluster unit is helpful for improving the classification results. Method I achieves a better classification performance than Method I, which means that the use of the point-cluster-based features as the basic unit can enhance the performance of the classification results. However, Method I cannot effectively recognize the points of trees, buildings, and cars from the cluttered point clouds. Although Method III achieves good classification results, the classification quality of cars need to be further improved. In our method, discriminative dictionary learning and feature representation with sparse coding can accurately express the characteristics of the objects. Furthermore, the multipath Adaboost classifiers on the features of the point clusters obtained by the DDL-SCLDA generate much higher classification accuracy.

C. Classification Results Obtained by Using the SVM Classifier

To compare the classification performance achieved by different classifiers, we use an SVM classifier [49] to reclassify the point clouds in Scenes I and II. The classification results have been listed in Table VI. From this table, we notice that the precision/recall and classification accuracy is high using the SVM classifier trained on the point-cluster-based features. However, the classification performance is much better using the AdaBoost classifiers.

D. Sensitivity of the Parameters to the Classification Accuracy

Here, we analyze the impact of the parameters on the classification results. The F_1 measure is used to represent the classification accuracy of Scenes I and II as follows:

$$F_1 = \frac{2(\text{recall} \times \text{precision})}{\text{recall} + \text{precision}}. \tag{15}$$

1) Sparsity Constraint Factor T: A large T increases the number of nonzero elements in the sparse representation, and meanwhile, it may introduce some redundant elements. On the contrary, a small value of T refines the sparse representation, although it may ignore some essential features. To determine the proper values of T, we set five different values of T, i.e., 26, 28, 30, 32, and 34, to test the influences on the classification in

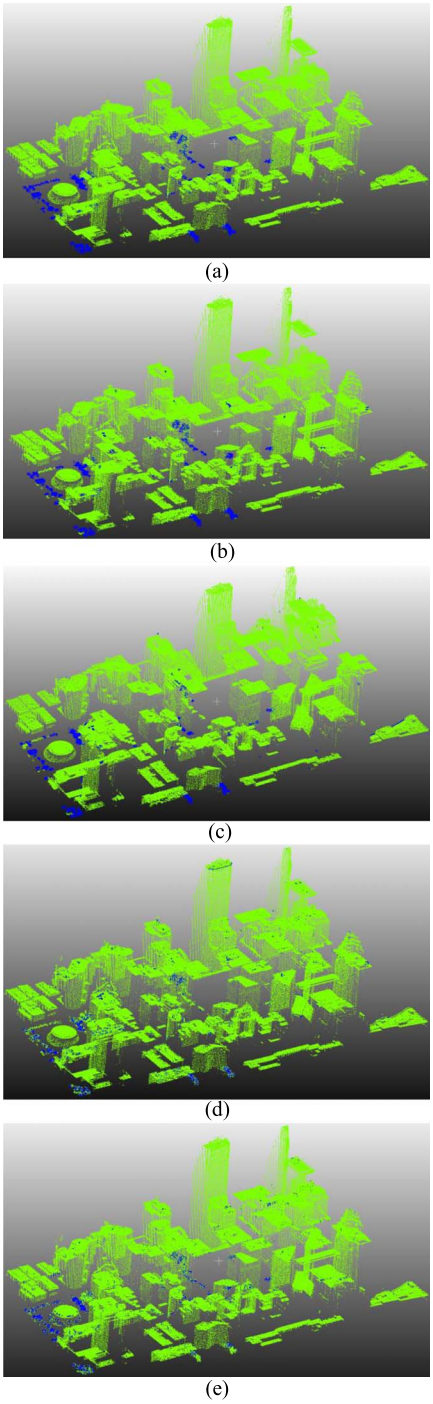


Fig. 6. Classification results obtained by different methods in a part of Scene II. (a) Ground truth. (b) Classification results obtained by using our method. (c) Classification results obtained by using Method I. (d) Classification results obtained by using Method II. (e) Classification results obtained by using Method III. The points on trees, buildings, and cars are colored in blue, green and red, respectively. Among the four methods, our method can correctly classify most of the points, and outperforms the other three methods.

TABLE VI
PRECISION/RECALL AND ACCURACY OF THE CLASSIFICATION RESULTS OBTAINED BY USING THE ADABOOST CLASSIFIERS AND THE SVM CLASSIFIER

Scene I	Trees (%)	Buildings (%)	Cars (%)	Accuracy (%)
AdaBoost classifiers	93.1/96.0	95.2/92.6	73.3/62.2	93.7
SVM classifier [49]	90.5/91.2	90.3/90.5	66.3/49.9	90.0
Scene II	**Trees (%)**	**Buildings (%)**	**Accuracy (%)**	
AdaBoost classifiers	93.0/86.8	99.4/99.7	99.2	
SVM classifier [49]	96.1/76.9	98.4/99.9	98.3	

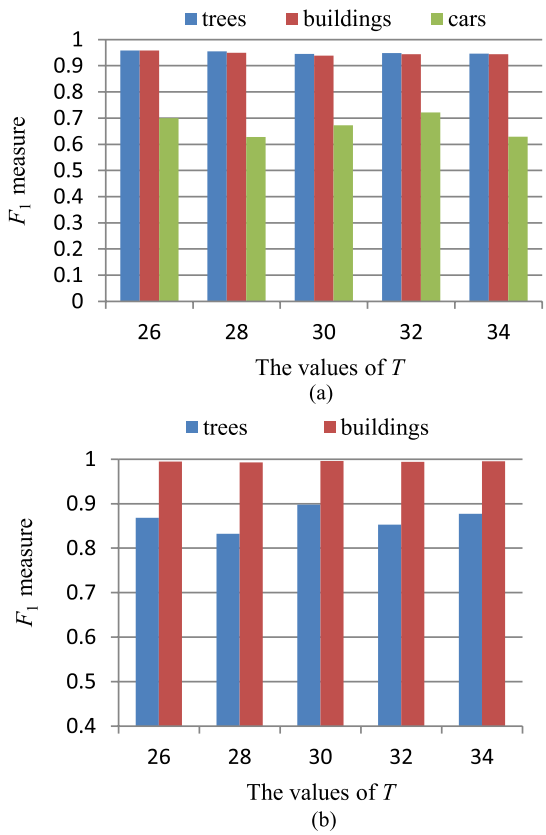


Fig. 7. Impacts of different T values on the classification results. (a) Influences of different T values on the classification results in Scene I. (b) Influences of different T values on the classification results in Scene II.

the two scenes. Other parameters are set as follows: $N = 10$, $\alpha = 16$, $\lambda = 0.15$, $k = 30$, $K = 512$, and $\eta = 14$. As shown in Fig. 7(a), the F_1 values of trees and buildings are larger than 0.9, and that of cars is smaller than those of buildings and trees, which is attributed to the less training samples; even so, the F_1 value of cars is no less than 0.6. When $T = 32$, the F_1 value of cars reaches the highest. From Fig. 7(b), it is noted that the F_1 value of buildings almost reaches 1. The F_1 value of trees is more than 0.8 even when the training samples of trees are far less than those of buildings. As shown in Fig. 7, the values of T slightly influence the classification quality, and our method can

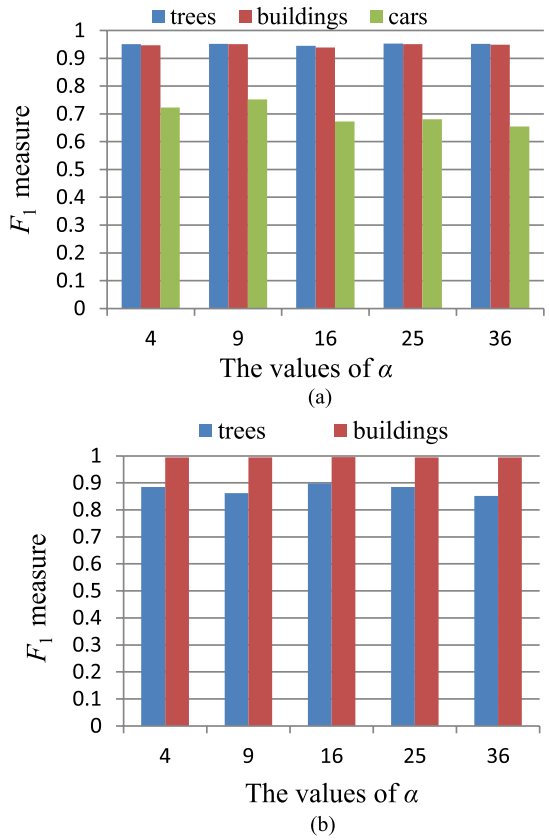


Fig. 8. Impacts of α on the classification results. (a) Impacts of α on the classification results in Scene I. (b) Impacts of α on the classification results in Scene II.

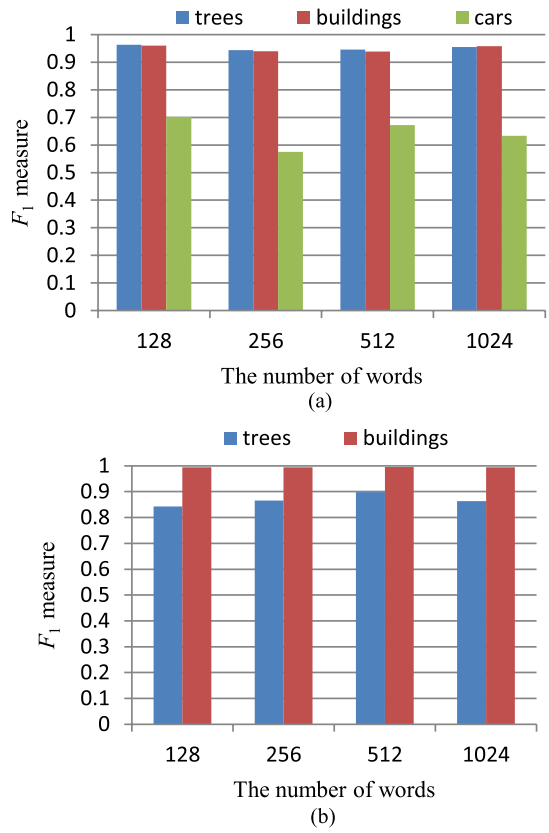


Fig. 9. Impacts of different word numbers on the classification results. (a) Influences of different word numbers on the classification results in Scene I. (b) Influences of different word numbers on the classification results in Scene II.

generate better classification performance when the values of T range from 30 to 32.

2) Weight for the Label Constraint Term: The influences of different values of the weight α in (1) on classification results are also tested. The parameters are set as follows: $N = 10$, $T = 30$, $\lambda = 0.15$, $k = 30$, $K = 512$, and $\eta = 14$. F_1 Measure values under different values of α are shown in Fig. 8. For Scene I, F_1 measure values of trees and buildings are approximate equal and larger than 0.9. With the change of α, fluctuation of F_1 measures on trees, buildings, and cars are not large. When $\alpha = 9$, the classification of the three categories can obtain better performance. For Scene II, F_1 measure values of buildings almost reaches 1, and F_1 measure values of trees range from 0.8 to 0.9 under the condition of less training tree data. With the increase in α, the classification results remain stable, and F_1 measure is the highest when α ranges from 9 to 16.

3) Number of Words: We also investigate the effect of the dictionary size (the number of words) on the classification results. Fig. 9 shows the classification quality of Scenes I and II corresponding to the different numbers of words. The parameters are set as follows: $N = 10$, $T = 30$, $\alpha = 16$, $\lambda = 0.15$, $k = 30$, and $\eta = 14$. We select 128, 256, 512, and 1024 words to show the impact of different word numbers on the classification quality, respectively. Fig. 9 shows that the numbers of the words have little influence on the classification quality of the three categories in the two scenes. The increase in the dictionary size enriches the encoded properties, which makes the feature representation discriminative. As shown in Fig. 9(a) and (b), the F_1 value have a better scalability on the result of classification with the increase in the number of words, and when the number of words is equal to 512, we can get better classification accuracy.

4) Influences of Different Multilevels on Classification Accuracy: To validate the influence of the multilevel point clusters on the classification results, we segment the input point cloud into the data set with one level (i.e., the original point cloud), the data set with two levels, the data set with three levels, the data set with four levels, and the data set with five levels, respectively. The corresponding parameters are as follows: $N = 10$, $T = 30$, $\alpha = 16$, $\lambda = 0.15$, $K = 512$, and $k = 30$. Fig. 10 illustrates the classification results obtained using the data set with different levels. In Fig. 10(a), we notice that the F_1 measures of buildings and trees using the data set with three levels are approximately equal to those obtained using the data set with four levels or five levels. However, the F_1 measure of cars is the highest using the data set with three levels. The F_1 measure of each category using the data set with one level is the lowest. With the level number increasing, the classification results keep high accuracy. In Scene II, there are only two categories. Correspondingly, there are many points on each object. From Fig. 10(b), it is noted that we can better distinguish trees and buildings using four levels. At the same

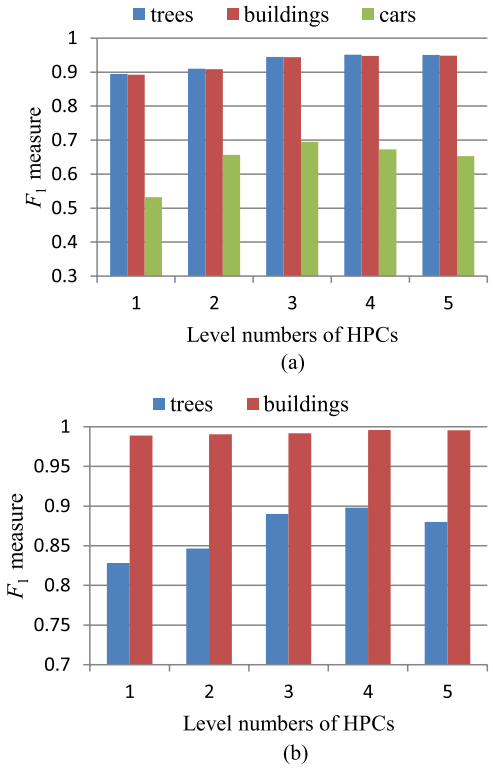


Fig. 10. Impacts of the level numbers of HPCs on the classification results. (a) Influences of the level numbers of HPCs on the classification results in Scene I. (b) Influences of the level numbers of HPCs on the classification results in Scene II.

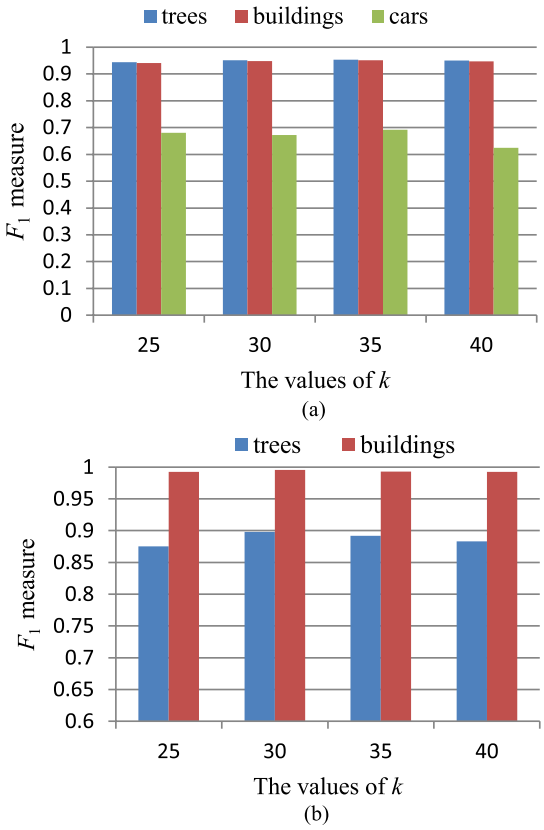


Fig. 11. Impacts of the number of points in the neighborhood on the classification results. (a) Influences on the classification result in Scene I. (b) Influences on the classification result in Scene II.

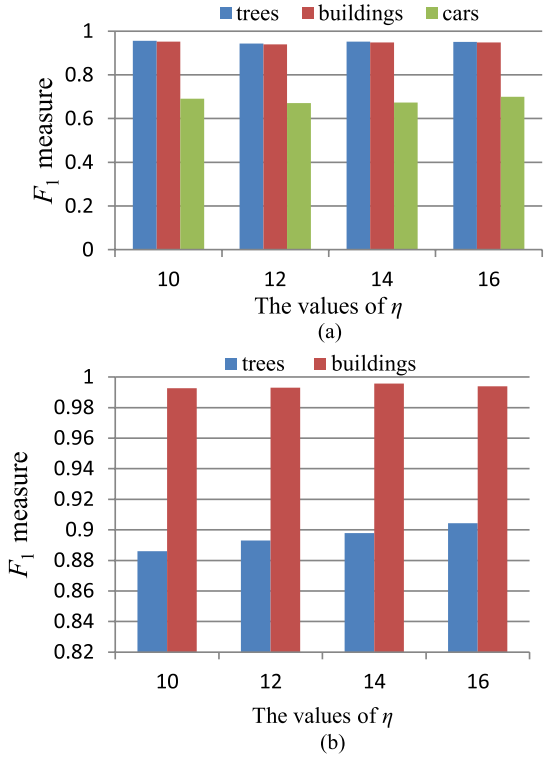


Fig. 12. Influences of different η on the classification results. (a) Influences of different η on the classification results in Scene I. (b) Influences of different η on the classification results in Scene II.

TABLE VII
CONFUSION MATRIX OF THE CLASSIFICATION RESULTS IN SCENE I

	Trees	Buildings	Cars	Recall
Overall accuracy: 94.8%				
Trees	206,921	6,811	390	0.966
Buildings	11,524	188,523	525	0.940
Cars	2043	514	5,259	0.673
Precision	0.938	0.963	0.852	

TABLE VIII
CONFUSION MATRIX OF THE CLASSIFICATION RESULTS IN SCENE II

	Trees	Buildings	Recall
Overall accuracy: 99.2%			
Trees	18,565	2828	0.868
Buildings	1,394	493,735	0.992
Precision	0.930	0.994	

time, the classification accuracy obtained using multiple levels is obviously higher than that using one level. In general, we can achieve high-quality classification results when the level number is equal to 3 or 4.

5) Influences of the Neighborhood Sizes on Classification Accuracy: The neighborhood size is determined by the number k of the points in the support region. We set k to 25, 30, 35, and 40, respectively. The other parameters are as follows: $N = 10$, $T = 30$, $\alpha = 16$, $\lambda = 0.15$, $K = 512$, $\eta = 14$, and $n = 4$.

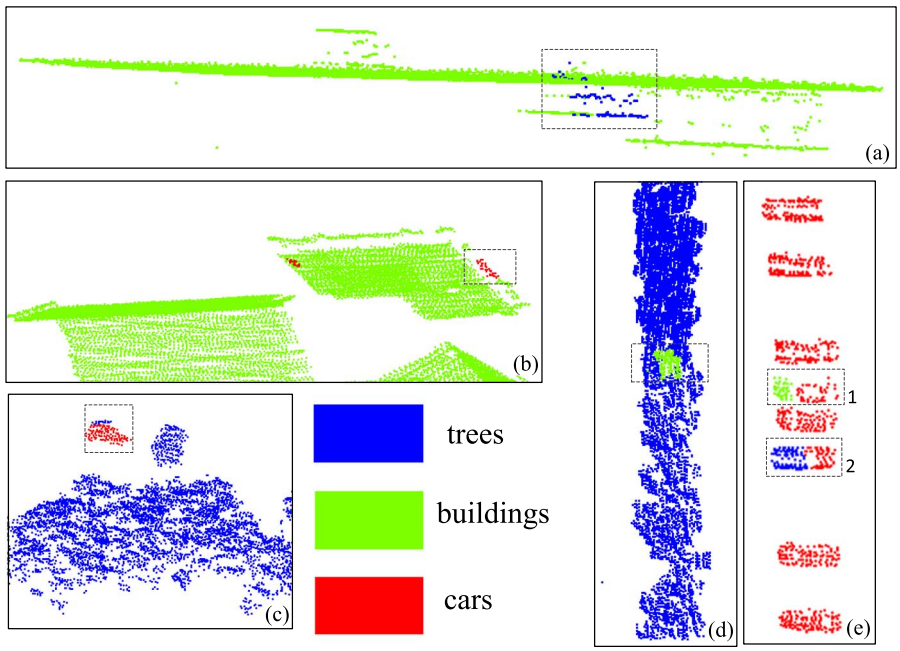


Fig. 13. Typical misclassification errors. (a) Points of the concave exits on the buildings are misclassified as trees. (b) Points on the building edge are misclassified as trees. (c) Tree points are misclassified as cars. (d) Tree points are misclassified as buildings. (e) Car points are misclassified as trees and buildings.

Fig. 11 illustrates the classification results under different values of k. As shown in Fig. 10(a), for the four different values of k, F_1 measures of trees and buildings all are above 0.9 and remain stable. F_1 measure of cars is larger than 0.6, and it is about equal to 0.7 as $k = 35$. In Fig. 10(b), we observe that F_1 measures of buildings almost reach 1, and those of trees are more than 0.87. We find the classification results have higher accuracy when k is between 30 and 35, as shown in Fig. 11(a) and (b).

6) Point-Cluster Generation Threshold: We test the influences of different point-cluster generation thresholds δ_m on the classification results. δ_m is controlled by η in $\delta_m = \eta e^x$. Let $\eta = 10$, 12, 14, and 16, respectively, and the other parameters are as follows: $N = 10$, $T = 30$, $\alpha = 16$, $\lambda = 0.15$, $K = 512$, $k = 30$, and $n = 4$. As shown in Fig. 12, we notice that different values of δ_m keep high and stable classification performances, and have little influences on classification accuracy.

E. Error Analysis

We analyze the classification results of Scenes I and II under the condition that $K = 512$ words, 30 topics, $\lambda = 0.15$, $\eta = 14$, and $\alpha = 16$. Tables VII and VIII list the confusion matrices of the two scenes. Some concave exits exist on some buildings; thus, the points on the inner side of exits are easily misclassified into trees [see the dashed box in Fig. 13(a)] as they are similar to trees in shapes. The points on the prominent eaves are often wrongly classified into car points [the dashed box in Fig. 13(b)]. For the error analysis of tree classification, on the one hand, the generated point cluster of the single tree crown [the dashed box in Fig. 13(c)] may be wrongly classified into cars as its spatial distribution is similar to the point cluster of cars; on the other hand, the tree points are misclassified into building roofs as some point clusters are nearly flat [see the dashed box in Fig. 13(d)]. Due to the inhomogeneity of the input point cloud, the points on some cars are presented in a clump as shown in the dashed box 1 in Fig. 13(e), which are misclassified into tree points. Some points on cars are scattered so that the recognition of the points is often wrong, such as the points in the dashed box 2 in Fig. 13(e).

IV. Conclusion

In this paper, a discriminative feature based on multilevel point clusters has been presented. At the first step, the input ALS point cloud is segmented into a set of point clusters, and then the point clusters are divided into different levels. Afterward, a dictionary learning approach is introduced to present the point-based features. At the second step, the feature of each point cluster is constructed from the point-based features by the DDL-SCLDA. Finally, the points in the ALS point cloud belonging to the specific categories are recognized by employing the multipath AdaBoost classifiers on the point-cluster-based features. We have performed the experiments on different complex ALS point clouds. The experimental results show that our presented method outperforms other state-of-the-art methods such as those in [7] and [50]. Meanwhile, the setting of the parameters in our method has little influences on the classification performance, which means that the method is robust to recognizing different point clouds.

In future work, we will improve the efficiency of our method. Moreover, inspired by [38], we plan to combine dictionary learning through several pathways, utilizing multiple point clusters [51] and encoding each point cluster through multiple paths, to learn features through multiple paths to further enhance the classification accuracy.

REFERENCES

[1] A. Fawzi, M. Davies, and P. Frossard, "Dictionary learning for fast classification based on soft-thresholding," *Int. J. Comput. Vis.*, vol. 114, no. 2, pp. 306–321, Sep. 2015.

[2] H. V. Nguyen, H. T. Ho, V. M. Patel, and R. Chellappa, "Joint hierarchical domain adaptation and feature learning," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5479–5491, Dec. 2015.

[3] S. Mei, M. He, Y. Zhang, Z. Wang, and D. Feng, "Improving spatial–spectral endmember extraction in the presence of anomalous ground objects," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4210–4222, Nov. 2011.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. 1, pp. 886–893.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[6] J. Niemeyer, C. Mallet, F. Rottensteiner, and U. Soergel, "Conditional random field for the classification of lidar point clouds," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 38, (Part 4/W19) (on CD-ROM), Aug. 2012.

[7] Z. Wang *et al.*, "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2409–2425, May 2015.

[8] E. H. Lim and D. Suter, "Multi-scale conditional random fields for over-segmented irregular 3D point clouds classification," in *Proc. IEEE Comp. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Anchorage, AK, USA, 2008, pp. 1–7.

[9] E. H. Lim and D. Suter, "3D terrestrial LIDAR classifications with super-voxels and multi-scale conditional random fields," *Comput. Aided Des.*, vol. 41, no. 10, pp. 701–710, Oct. 2009.

[10] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Segmentation based classification of 3D urban point clouds: A super-voxel based approach with evaluation," *Remote Sens.*, vol. 5, no. 4, pp. 1624–1650, Mar. 2013.

[11] L. Truong-Hong, D. F. Laefer, T. Hinks, and H. Carr, "Combining an angle criterion with voxelization and the flying voxel method in reconstructing building models from LiDAR data," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 28, no. 2, pp. 112–129, Feb. 2013.

[12] H. B. Kim and G. Sohn, "Random forests-based multiple classifier system for power-line scene classification," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 38, (Part 5/W12) (on CD-ROM), 2011.

[13] M. Lehtomaki *et al.*, "Object classification and recognition from mobile laser scanning point clouds in a road environment," *IEEE Trans. Geosci. Rem. Sens.*, vol. 54, no. 2, pp. 1226–1239, Oct. 2015, doi: 10.1109/TGRS.2015.2476502.

[14] S. K. Lodha, D. M. Fitzpatrick, and D. P. Helmbold, "Aerial lidar data classification using adaboost," in *Proc. Int. Conf. 3-D Digital Imaging Model.*, Montreal, QC, Canada, 2007, pp. 435–442.

[15] T. Kohonen, "Improved versions of learning vector quantization," in *Proc. IJCNN*, 1990, vol. 1, pp. 545–550.

[16] Z. Wang, N. Nasrabadi, and T. Huang, "Spatial–spectral classification of hyperspectral images using discriminative dictionary designed by learning vector quantization," *IEEE Trans. Geosci. Remote. Sens.*, vol. 52, no. 8, pp. 4808–4822, Aug. 2013.

[17] J. Yang, K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3517–3524.

[18] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[19] N. Zhou, Y. Shen, J. Peng, and J. Fan, "Learning inter-related visual dictionary for object recognition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3490–3497.

[20] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification," *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 209–232, 2014.

[21] G. Karol and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 1–8.

[22] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.

[23] L. Shen, G. Sun, Q. Huang, Z. Lin, and E. Wu, "Multi-level discriminative dictionary learning with application to large scale image classification," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3109–3123, Oct. 2015.

[24] M. Weinmann, B. Jutzi, and C. Mallet, "Feature relevance assessment for the semantic interpretation of 3D point cloud data," in *Proc. ISPRS Ann. Photogramm., Remote Sens., Spatial Inf. Sci.*, 2013, vol. II-5/W2, pp. 1–6.

[25] M. Weinmann, S. Urban, S. Hinz, B. Jutzi, and C. Mallet, "Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas," *Comput Graph.*, vol. 49, pp. 47–57, Jun. 2015.

[26] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proc. IEEE Comp. Soc. Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, 2006, pp. 1605–1614.

[27] J. Xiao and L. Quan, "Multiple view semantic segmentation for street view images," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kyoto, Japan, 2009, pp. 686–693.

[28] N. Brodu and D. Lague, "3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology," *ISPRS J. Photogramm. Remote Sens.*, vol. 68, pp. 121–134, Mar. 2012.

[29] S. Xu, G. Vosselman, and S. Elberink, "Multiple-entity based classification of airborne laser scanning data in urban areas," *ISPRS J. Photogramm. Remote Sens.*, vol. 88, pp. 1–15, Feb. 2014.

[30] Y. Gu, Q. Wang, X. Jia, and J. A. Benediktsson, "A novel MKL model of integrating LiDAR data and MSI for urban area classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5312–5326, Oct. 2015.

[31] J. Rau, J. Jhan, and Y. Hsu, "Oblique aerial images for land cover and point cloud classification in an urban environment," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1304–1319, Mar. 2015.

[32] S. Bengio, J. Weston, and D. Grangier, "Label embedding trees for large multiclass task," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 163–171.

[33] T. Gao and D. Koller, "Discriminative learning of relaxed hierarchy for large-scale visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2072–2079.

[34] R. Salakhutdinov, A. Torralba, and J. Tenenbaum, "Learning to share visual appearance for multiclass object detection," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1481–1488.

[35] L. Zhou, Z. Zhou, and D. Hu, "Scene classification using a multi-resolution bag-of-features model," *Pattern Recognit.*, vol. 46, pp. 424–433, Jan. 2013.

[36] L. Bo, X. Ren, and D. Fox, "Hierarchical matching pursuit for image classification: Architecture and fast algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2011, pp. 2115–2123.

[37] K. Yu, Y. Lin, and J. Lafferty, "Learning image representations from the pixel level via hierarchical sparse coding," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1713–1720.

[38] L. Bo, X. Ren, and D. Fox, "Multipath sparse coding using hierarchical matching pursuit," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 660–667.

[39] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[40] H. Gross and U. Thoennessen, "Extraction of lines from laser point clouds," in *Proc. Photogramm. Image Anal.*, Bonn, Germany, 2006, pp. 87–91.

[41] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 4311–4322, Nov. 2006.

[42] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 801–808.

[43] D. Chen, L. Zhang, Z. Wang, and H. Deng, "A mathematical morphology-based multi-level filter of LiDAR data for generating DTMs," *Sci. Chin. Inf. Sci.*, vol. 56, no. 10, pp. 1–14, Oct. 2013.

[44] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[45] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[46] G. Golub, P. Hansen, and D. O'leary, "Tikhonov regularization and total least squares," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 1, pp. 185–194, 1999.

[47] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[48] M. Cramer, "The DGPF test on digital aerial camera evaluation—Overview and test design," *Photogramm.-Fernerkundung-Geoinf.*, vol. 2, pp. 73–82, Jan. 2010.

[49] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE CVPR*, Jun. 2009, pp. 1794–1801.
[50] B. Guo, X. Huang, F. Zhang, and G. Sohn, "Classification of airborne laser scanning data using JointBoost," *ISPRS J. Photogramm. Remote Sens.*, vol. 100, pp. 71–83, Feb. 2015.
[51] Z. Zhang *et al.*, "A multi-level point cluster-based discriminative feature for ALS point cloud classification," *IEEE Trans. Geosci. Remote. Sens.*, vol. 54, no. 6, pp. 3309–3321, Jun. 2016.

Zhenxin Zhang is currently working toward the Ph.D. degree in geoinformatics with the State Key Laboratory of Remote Sensing Science, School of Geography, Beijing Normal University, Beijing, China.

His research interests include light detection and ranging data processing, quality analysis of geographic information systems, and algorithm development.

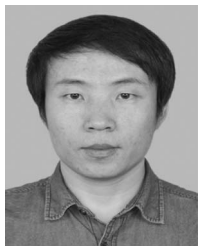
Liqiang Zhang received the Ph.D. degree in geoinformatics from the Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 2004.

He is currently a Professor with the State Key Laboratory of Remote Sensing Science, School of Geography, Beijing Normal University, Beijing, China. His research interests include remote sensing image processing, 3-D urban reconstruction, and spatial object recognition.

Xiaohua Tong received the Ph.D. degree from Tongji University, Shanghai, China, in 1999.

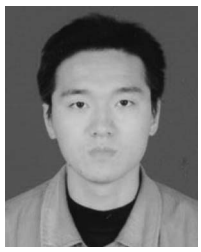
Between 2001 and 2003, he worked as a Postdoctoral Researcher with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China. In 2006, he was a Research Fellow with The Hong Kong Polytechnic University, Hong Kong. Between 2008 and 2009, he was a Visiting Scholar with the University of California, Santa Barbara, CA, USA. He is currently with the School of Surveying and Geo-informatics, Tongji University. He is a Chang-Jiang Scholar Chair Professor appointed by the Ministry of Education, China. He is the author of more than 40 publications in international journals. His current research interests include remote sensing, geographic information systems, trust in spatial data, image processing for high resolution, and hyperspectral images.

Dr. Tong serves as the Vice Chair of the Commission on Spatial Data Quality of the International Cartographical Association and the Co-Chair of the ISPRS Working Group (WG II/4) on Spatial Statistics and Uncertainty Modeling. He received the State Natural Science Award (Second Place) from the State Council of the Peoples' Republic of China in 2007 and the National Natural Science Funds for Distinguished Young Scholar in 2013.



Bo Guo received the Ph.D. degree from Wuhan University, Wuhan, China, in 2014.

He is currently a Postdoctoral Researcher with the Key Laboratory for Geo-Environment Monitoring of Coastal Zone of the National Administration of Surveying, Mapping and GeoInformation and Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen, China. His research interests include applications of laser scanning point cloud on photogrammetry and computer vision.



Liang Zhang is currently working toward the Ph.D. degree with the State Key Laboratory of Remote Sensing Science, School of Geography, Beijing Normal University, Beijing, China.

His research interests include remote sensing imagery processing and 3-D urban modelling.

Xiaoyue Xing is currently working toward the Bachelor's degree with the State Key Laboratory of Remote Sensing Science, School of Geography, Beijing Normal University, Beijing, China.

Her research interests include terrestrial laser scanning point-cloud processing.