An Efficient Class-Constrained DBSCAN Approach for Large-Scale Point Cloud Clustering

Hua Zhang, Zhenwei Duan, Nanshan Zheng, Yong Li, Yu Zeng, and Wenzhong Shi

***Abstract*—To better interpret the scene and facilitate the subsequent processing of large-scale point cloud, clustering is often implemented in the preprocessing stage. However, when the original density-based spatial clustering of application with noise (DBSCAN) approach is used for point cloud clustering, it is easy to categorize closely spaced vegetation points and nonvegetation points into the same cluster by mistake. Aiming at the problem, this article presents an improved DBSCAN by embedding a strategy of class constraint, which is called CC-DBSCAN. Specially, based on the RGB and label information of each point in the training samples, by using the logistic regression model, the logistic regression color index (LRCI) is calculated for each point in the clustering samples. Then, points to be clustered are classified as vegetation points and nonvegetation points through the LRCI. Furtherly, the class information of point is introduced as a constraint for ensuring the core point and its directly density-reachable points belong to the same class, thus, solving the problem that confusion cluster of the adjacent vegetation points and nonvegetation points. We evaluate our approach on the benchmark SensatUrban dataset, where Cambridge_28 scene dataset is taken as the training set and Cambridge_18 scene dataset is as the dataset to be clustered. Experimental results show that our method achieved 97.20% purity of point cluster, which outperforms the other DBSCAN methods. At the same time, it takes only 24.25 s for clustering 2 million points, which indicates that CC-DBSCAN has high computational efficiency and good practicability.**

***Index Terms*—Class constraint, color index, density-based spatial clustering of application with noise (DBSCAN), logical regression, point cloud.**

I. Introduction

POINT cloud data are widely applied in remote sensing, forest ecology, urban change detection, and autopilot technology, etc., for its rich geometric, shape, and scale information [1]. While effective interpreting the point cloud data, as a core step of the aforementioned applications, plays an important role. Due to that large-scale point cloud has the characteristics of many kinds of ground objects, complex types, and large amounts of data, the task of fast and precise interpretation of point cloud remains a great challenge. Thus, clustering methods are often implemented to preprocess point cloud of large scene, which is helpful to better understand the scene and facilitate subsequent processing, such as the monomerization of ground objects, the learning of overall geometric features, and the balanced sampling of scene points [2]. On this basis, this article focuses on the developing clustering method, which is used to preprocess the point cloud data.

Clustering is a technique of unsupervised clustering of similar data into a cluster based on internal attributes [3]. Many clustering algorithms, such as K-means [4], k nearest neighbor (KNN) [5], fuzzy c-means [6], expectation maximization [7], ISODATA [8], BIRCH [9], Sting [10], Mean shift [11], density-based spatial clustering of application with noise (DBSCAN) [12], [13], and their variations, have been exploited for unsupervised point cloud clustering. Amongst them, considering the characteristics of point cloud as described earlier, DBSCAN is the most suitable for point cloud clustering due to its robustness against noise and ability to detect arbitrary shapes of clusters in any dimension [12]. Despite its advantages, the original DBSCAN still has the following main shortcomings: 1) it is difficult to determine the appropriate values of input parameters, including scanning radius *eps* and density threshold *minPts*, 2) the computational complexity is high due to redundant distance computations. These drawbacks hinder the application of DBSCAN algorithm in large-scale point cloud clustering to some extent.

In order to solve the aforementioned shortcomings of traditional DBSCAN algorithm, many researchers have done large number of efforts to enhance its performance [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24]. Karami and Johansson [14] proposed a BDE-DBSCAN, which combines the binary differential evolution (BDE) and DBSCAN algorithm to automatically determine proper values for parameter *eps* and *minPts*. Khan et al. [15] presented an adaptive DBSCAN algorithm to determine appropriate values for parameter *eps* and *minPts* so that the algorithm can identify clusters with varying densities. Wang et al. [16] proposed an improved DBSCAN method for lidar data segmentation, in which the clustering parameters for DBSCAN can be estimated automatically based on the characteristics of data without any prior knowledge. Lai et al. [17] suggested a new method for determining the parameters of

Manuscript received 6 July 2022; revised 1 August 2022; accepted 23 August 2022. Date of publication 26 August 2022; date of current version 8 September 2022. This work was supported in part by the National Natural Science Foundation, China under Grant 41971400 and Grant 41974039 and in part by the Fundamental Research Funds for the Central Universities under Grant 2019ZDPY09. *(Corresponding authors: Zhenwei Duan; Nanshan Zheng.)*

Hua Zhang, Zhenwei Duan, and Nanshan Zheng are with the School of Environment and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China (e-mail: zhhua_79@163.com; duan_0607@163.com; znshcumt@163.com).

Yong Li is with the Sichuan Institute of Coal Field Geological Engineering Exploration and Designing, Chengdu 610072, China (e-mail: 59501359@qq.com).

Yu Zeng is with the Sichuan Institute of Coal Field Surveying and Mapping Engineering, Chengdu 610072, China (e-mail: 185517025@qq.com).

Wenzhong Shi is with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong (e-mail: lswzshi@polyu.edu.hk).

Digital Object Identifier 10.1109/JSTARS.2022.3201991

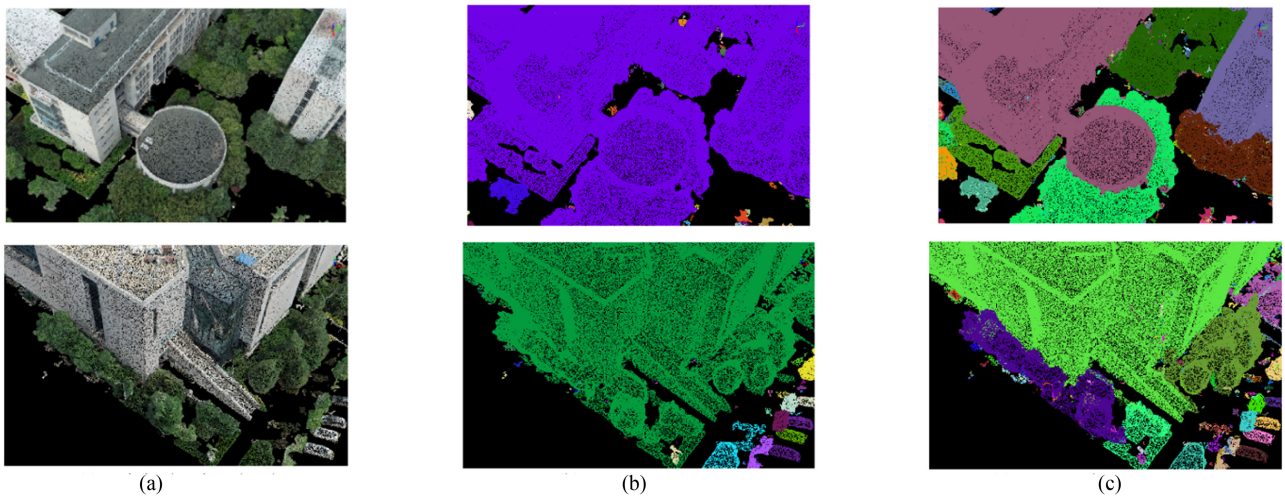

Fig. 1. Visualization of clustering results by DBSCAN and proposed method. (a) Original point cloud. (b) DBSCAN (*xyz*). (c) Proposed CC-DBSCAN.

DBSCAN, where the multiverse optimizer algorithm is utilized to optimize the parameters through a special variable updating method. Chen et al. [18] proposed the APSCAN algorithm, in which the affinity propagation algorithm was utilized to generate a normalized density list, and then the density parameters in the normalized density are combined as the parameters of the DBSCAN. In [19], based on the clusters from DSets, the input parameters are automatically determined. The aforementioned methods aim to specify appropriate values for parameters *eps* and *minPts* in DBSCAN. In addition, to improve the computational efficiency of DBSCAN for dealing with large-scale data, many strategies have been put forward, Li [20] proposed an improve DBSCAN, where a nearest neighbor similarity based fast density algorithm and fast nearest neighbor search are presented for reducing the number of distance calculations, and then the clustering efficiency is improved. Jang and Jiang [21] presented the DBSCAN++ algorithm to improve the competitive performance and robustness by computing the densities for a chosen subset of points based on uniform and greedy k-center-based sampling strategies. In [22], a new KNN-BLOCK is presented for fast clustering, in which a fast approximate KNN algorithm is first utilized to detect three different blocks where all points have the same type, then a fast algorithm is proposed for merging CBs and assigning noncore points to proper clusters. Kumar and Reddy [23] proposed a fast DBSCAN clustering algorithm by accelerating the neighbor search operations based on a novel graph-based index structure. Chen et al. [24] suggested a NQ-DBSCAN algorithm, which reduced the unnecessary distance computations by using a novel local neighborhood searching technique. These methods have promoted the development and application of DBSCAN to some extent, however, there are still some problems exit when clustering outdoor large-scale point clouds. To illustrate this point, we visualize the clustering results of traditional DBSCAN and our proposed method.

As shown in Fig. 1(a), in the outdoor scene, the nonvegetation objects are generally discontinuous and not in contact with each other in geometric space, whereas the vegetation objects are usually prosperous and close to or even in contact with other types of objects (such as buildings). When the original DBSCAN algorithm is utilized to cluster the large-scale point clouds in the outdoor scene, the vegetation points and nonvegetation points are easily categorized into one cluster only based on distance-based judgment using the position information (x, y, z) due to that the vegetation points are close to nonvegetation points in space. Fig. 1(b) shows the clustering results by the DBSCAN algorithm based on the position (*xyz*) of points, we can find that a large number of building points and vegetation points are categorized into the same clusters by mistake while most of the points are correctly clustered using the proposed class-constrained DBSCAN (CC-DBSCAN) in this article.

To address the problems, this article presents an efficient CC-DBSCAN approach for large-scale point cloud clustering. First, the logistic regression color index (LRCI) is calculated for each point by using the logistic regression model based on the RGB and label information provided by the training samples. Then, points needing to be clustered are labeled as vegetation points and nonvegetation points through the calculated LRCI of each point. Finally, the class information of point is introduced as a constraint in the DBSCAN to reduce the number of distance calculations and prevent points close to each other but different classes. The main contributions of this article are listed as follows.

1) We propose a CC_DBSCAN algorithm for efficient and accurate large-scale point cloud clustering.
2) A novel color index (LRCI) is proposed and calculated for efficiently categorizing points as vegetation points and nonvegetation points.
3) A novel class constraint strategy is incorporated in CC_DBSCAN to reduce the number of distance calculations and prevent points close to each other but different classes. In addition, other color index can also be favorably adopted in our algorithm.
4) The proposed algorithm achieves 97.20% purity of point cluster, and high computational efficiency with 24.25 s for clustering 2 million points.

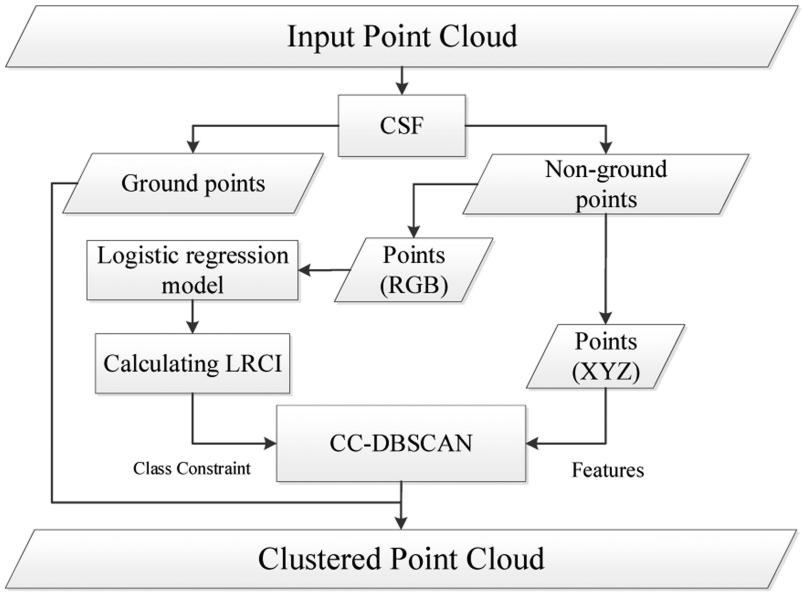

Fig. 2. Framework of CC-DBSCAN for point cloud clustering.

The rest of this article is organized as follows. Section II introduces the proposed CC-DBSCAN in details. Section III describes the experiments and analyzes the results. Finally, Section IV concludes this article.

II. Proposed Method

Fig. 2 shows the framework of CC-DBSCAN for the outdoor scene large-scale point cloud clustering. First, the cloth simulation filter (CSF) algorithm [25] is applied to classify the original points as ground points and nonground points, and the nonground points are furtherly processed through the following steps: based on the RGB and label information of each point in the training samples, the classification decision boundary between vegetation points and nonvegetation points is obtained by using the logistic regression model, then, on the basis of the boundary parameters obtained from the aforementioned classification decision boundary, the LRCI is calculated for every point and used to classify each point in the point cloud to be clustered as vegetation point or nonvegetation point. Furtherly, together with the position information (x, y, z), the class information of each point is embedded as a constraint for the DBSCAN to cluster the nonground points. Finally, the clustered results and the ground points are merged as the final clustering result.

A. Improved DBSCAN With Class Constraint (CC-DBSCAN)

The basic principles of DBSCAN and the DBSCAN-based algorithms [12] are as following: according to the setting scanning radius (*eps*) and the minimum number of included points (*minPts*), randomly selecting a point which is not visited as the starting point, and finding out its neighborhood points that are within a radius of *eps* (including *eps*). If the number of neighborhood points $\geq$ *minPts*, then the current point and its neighborhood points form a cluster, which is called directly density-reachable points set, and the current point is named as core point, the starting point is marked as visited. Recursively, all points in the cluster that are not marked as visited are processed in the same way, by this way, the cluster is expanded. From above, we can see that judging whether a point belongs to a cluster is only based on the distance between points without considering point's characteristics, such as land cover category, color, etc., which may lead to mistake. For example, as described earlier, the vegetation points are close to nonvegetation points for the vegetation cover, only distance-based judgment may lead to categorize them into one cluster by mistake. To address this problem, the strategy of class constraint is incorporated in the DBSCAN, which can be described as following: when judging whether the current point and its neighborhood points belong to cluster or not, if the nearby point has different class from the current point, the nearby point is not considered, thus, the false clustering can be prevented in a certain.

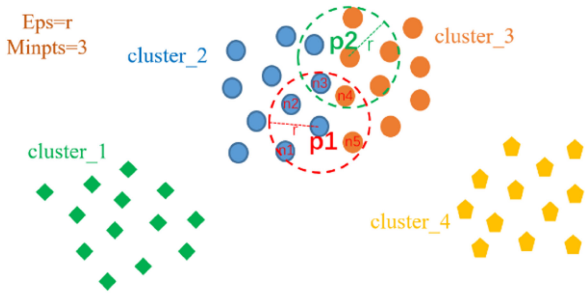

Fig. 3. Diagram of the DBSCAN algorithm.

A detailed description of the strategy is as following: as shown in Fig. 3, supposing there are four clusters, including cluster_1, cluster_2, cluster_3, and cluster_4, in which different color and shape indicate different clusters, *eps* is set as *r* and *minPts* is 3, the cluster_2 is adjacent to the cluster_3. p1 is selected as the starting point, according to the set scanning radius $eps = r$ and $minPts = 3$, its neighborhood points include n1, n2, n3, n4, and n5, and they are categorized into the same cluster, but the points

Algorithm 1: CC-DBSCAN.

Function: CC-DBSCAN (dataset *P*, *eps*, *minPts*)
Input: Dataset: *P*, *eps*, *minPts*
Output: Clusters *C*
Setting up an empty list of clusters *C* and an empty queue *Q* for the points that need to be checked
for $p_i \in P$
  if p_i is visited **then**
    continue
  end
  add p_i to the *Q*
  for $p_j \in Q$
    search neighborhood points set p_j^k of the p_j in a sphere with a radius of *eps*
      for $p_t \in p_j^k$
        if p_t is not visited and has the same category with p_j **then**
          add p_t to *Q*
        end
      end
  end
  number = the point number of *Q*
  if number $>=$ *minPts* **then**
    add *Q* to the clusters *C*
    for $p_j \in Q$ **do**
      mark p_j as visited
    end
    set *Q* as empty
  end
end
return *C*

n4 and n5 belong to the cluster_3 in fact, as a result, the cluster result is wrong. To improve it, the class constraint is incorporated in the DBSCAN, when judging whether the current point p1 and its neighborhood points n1, n2, n3, n4, and n5 belong to the same cluster or not, except for the radius *eps* and *minPts* constraints, the class constraint is added, if the neighborhood points have different classes from the current point p1, it will be not considered as the same cluster with the current point. By this way, points n4 and n5 will be clustered with point p1 into one cluster, then the points p1, n1, n2, and n3 are categorized into the same cluster. Recursively, the points n1, n2, and n3 are taken as the core points, and processed in the same way, by this way, the cluster is expanded, as a result, cluster_ 2 and cluster_ 3 are separated by two clusters. The optimized DBSCAN algorithm is as follows (Algorithm 1).

B. LRCI Value Calculated Based on Logistic Regression Model

From the aforementioned description of CC-DBSCAN, it is crucial to effectively distinguish whether the neighborhood points and the core point belong to the same class or not. And as described earlier, the vegetation points are close to nonvegetation points for the vegetation cover in the nonvegetation points set, it is easy to cluster the vegetation points and nonvegetation points into one by mistake. Thus, in this article, the nonground points are first divided into vegetation points or nonvegetation points before clustering. And the crucial problem becomes that the neighborhood points and the core point both belong to vegetation points or nonvegetation points. Inspired by the successful application of vegetation index in optical remote sensing classification, for the labeling of point cloud, nonvegetation points and vegetation points have very different spectral characteristics, so vegetation index may be used to preliminarily distinguish nonvegetation points and vegetation points. However, as for the calculation of traditional vegetation indices, the data of near-infrared band of vegetation are needed while the general point cloud does not have such band, but the RGB information can be obtained. Many classical vegetation indices based on RGB information had been proposed (see Table II), and they had been used to classify the point cloud into vegetation points and nonvegetation points in the study area, but the experimental results show unsatisfactory, the details are described in Section III-C. To address this problem, based on studying the distribution of RGB characteristics of point cloud, this article attempts to obtain the classification decision boundary of vegetation points and nonvegetation points in RGB space using the logical regression method, then, the LRCI is calculated for each point based on the boundary parameters of regression model, Finally, the point cloud is classified as vegetation points and nonvegetation points by LRCI.

As one of the most common algorithms in the field of machine learning, logical regression is usually used for classification [26]. In which, the probability regression is utilized to find the correlation between input features and output labels, and probability values are taken as the prediction results. It is a classical classification model with characteristics simple implementation, low computational cost, and highly efficient. In this article, the logical regression is introduced for binary classification (vegetation points and nonvegetation points), the probability distribution model of the logical regression is as (1)

$$\mathrm{P}(Y|X;W) = \left(h_W(X)\right)^Y \left(1 - h_W(X)\right)^{1-Y} \tag{1}$$

$$h_W(X) = \frac{1}{1 + e^{-W^T X}} \tag{2}$$

$$W^T X = \sum_{i=1}^{m} w_i f_i = w_0 + w_1 f_1 +, \ldots, +w_m f_m \tag{3}$$

where $Y = \{1, 0\}$ denotes the point is the vegetation point or nonvegetation point, respectively. $X = \{x_1, x_2, \cdots, x_N\}$ is the input sample points, and N is the number of sample points. $\mathrm{P}(1|X)$denotes the probability that point x is a vegetation point, and $\mathrm{P}(0|X)$ denotes the probability that point x is a nonvegetation point. $f = \{f_1, f_2, \cdots, f_m\}$ represents the features of input sample X, m is the feature number of input points, $W = \{w_1, w_2, \cdots, w_m\}$ represents the model parameters, and w_0 is the bias term. Equation (2) gives the prediction function for outputting the probability that the sample point belongs to the vegetation point, and the probability value is between 0 and 1, here, if $h_W(X) \geq 0.5$, then $Y = 1$, the current point belongs to vegetation class, otherwise, $Y = 0$, the current point belongs to nonvegetation class.

TABLE I
COMPARISON OF THE NUMBER OF POINTS BEFORE AND AFTER DOWN SAMPLING IN CAMBRIDGE_18 AND CAMBRIDGE_28 SCENES

Scene	Down sample	Ground	Vegetation	Building	Wall	Bridge	Parking	Traffic road	Street furniture	Car	Footpath	Bike	Water	Rail
Cambridge_18	Before	2 19 6330	9 20 0753	13 2569 94	54 618	0	1 668 332	1 672 848	260 057	874 730	651 143	3817	0	0
	After	185 506	765 371	1 106 998	6071	0	130 654	133 792	22 300	71 006	51 823	297	0	0
Cambridge_28	Before	7 11 2143	1 483 4535	18 264 992	575 512	1 462 599	1 128 157	3 964 518	552 574	1 063 593	1 284 032	1413	1 526 111	0
	After	603 998	1 261 789	1 487 593	61 369	112 999	88 281	310393	47 162	83 556	100 728	120	157 801	0

TABLE II
CLASSICAL VEGETATION INDEX

Vegetation Index	Equation
RI [28]	$RI = G / B$
VARI [28]	$VARI = (G - R) / (R + G - B)$
ExG [29]	$ExG = 2g - r - b$
ExG-ExR [29]	$ExG - ExR = 3g - 2.4r - b$
NGBDI [30]	$NGBDI = (G - B) / (G + B)$
VDVI [30]	$VDVI = (2G - R - B) / (2G + R + B)$
LRCI	$LRCI = 0.0471R - 0.0924G + 0.0807B$

From (2), we can see that how to obtain the optimum model parameter W is vital for classification, the maximum likelihood method is introduced to solve the optimal parameters, and the likelihood function $L(W)$ can be constructed based on (1)

$$L(W) = \prod_{i=1}^{n} P(y_i|x_i; W) = \prod_{i=1}^{n} (h_W(x_i))^{y_i} (1 - h_W(x_i))^{1-y_i}. \quad (4)$$

According to the requirement of the maximum likelihood method, the probability of correct prediction of the corresponding category of each sample is as large as possible, that is, the optimal parameter W can be obtained through calculating the maximum point of $L(W)$. While gradient ascent is rarely used in machine learning task, therefore, in order to convert it into the gradient descent task, which is easy to be solved, the following transformation is often done [26]:

$$l(W) = -\frac{1}{n} \log L(W) = -\frac{1}{n} \sum_{i=1}^{n} (y_i \log h_W(x_i) + (1 - y_i) \log (1 - h_W(x_i))). \quad (5)$$

Thus, the problem is transformed into an optimization problem with the objective function $L(W)$. The optimal parameter W can be obtain when $L(W)$ reaches the minimum value. In which, the gradient descent method is used to update the model parameters step by step using (6), and α is the learning rate of the model

$$\mathrm{w}_m := w_m - \alpha \frac{1}{n} \sum_{i=1}^{n} (h_W(x_i) - y_i) x_i^m. \quad (6)$$

Based on the obtained best parameters W, the classification decision boundary can be determined as $W^T X = w_0 + w_1 f_1 +, \ldots, + w_m f_m = 0$, and the probability prediction classification can be carried out according to (2).

To make it easy to be understanded, the boundary is transformed into $\mathrm{w}_1 f_1 + w_2 f_2 + \ldots + w_m f_m = -w_0$. In this article, the features of input point include RGB information, thus the feature number m of input points is 3. Therefore, we let $\mathrm{LRCI}(R, G, B) = w_1 R + w_2 G + w_3 B$, and $\delta_{vi} = -w_0$, thus, the LRCI value of each point in the point cloud can be calculated by the function $\mathrm{LRCI}(R, G, B)$, and the point cloud is classified as vegetation point or nonvegetation point using (7)

$$F(R, G, B) = \begin{cases} 1, \mathrm{LRCI}(R, G, B) > \delta_{vi} \\ 0, \mathrm{LRCI}(R, G, B) \leq \delta_{vi} \end{cases} \quad (7)$$

where F is the function for classifying point cloud, the input is a set of points with RGB information $(\mathrm{R}, \mathrm{G}, \mathrm{B})$.

III. EXPERIMENTAL RESULTS

A. Experimental Environment and Dataset

All algorithms are implemented with Python 3.7.9, and are tested on an Intel(R) Core (TM) i5-8500 CPU (@3.00 GHz 3.00 GHz), 8.0-GB RAM. To verify the performance and effectiveness of the proposed method, the SensatUrban dataset is used [27], it is an urban-scale photogrammetric point cloud dataset with nearly three billion labeled points, and consists of large areas from three U.K. cities (Birmingham, Cambridge, and York), covering about 7.6 km^2 of the city landscape. In which, each point is labeled as one of 13 classes, including ground, vegetation, building, wall, bridge, parking, traffic road, street furniture, rail, car, footpath, bike, and water. In our experiments, taking the Cambridge_28 scene in SensatUrban as the training dataset to obtain the classification decision boundary function for classifying vegetation points and nonvegetation points in the RGB color space, and the Cambridge_18 scene as the test dataset for vegetation points and nonvegetation points classification and point cloud data clustering. Since the density of point cloud is too high, which requires high computer hardware, they are first down-sampled with a uniform grid of 0.2 m without affecting the experimental effects, Table I gives the description of the point cloud datasets before and after down-sampled in Cambridge_18 and Cambridge_28 datasets.

B. Calculating the Classification Decision Boundary

The CSF algorithm [25] is first applied to divide the training point cloud data of Cambridge_28 scene into ground points and nonground points, in the logistic regression model, we use the following settings: adaptive moment estimation (Adam) optimizer is adopted, and the initial learning rate is set to 0.001,

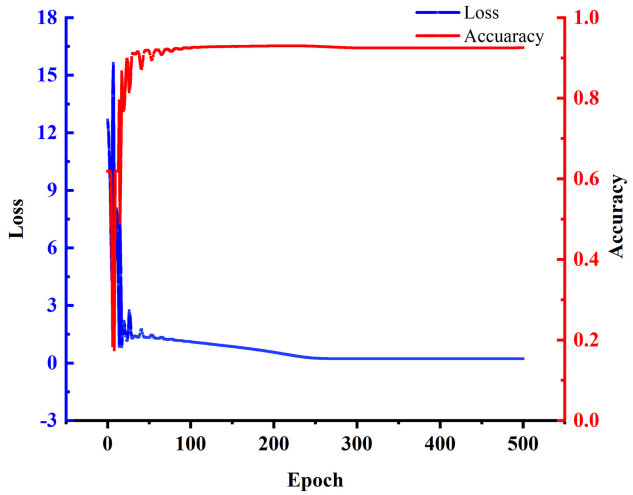

Fig. 4. Curves of loss and accuracy with different epochs.

TABLE III
QUANTITATIVE COMPARISON OF OA, PRECISION, RECALL, AND F1 OF THE POINT CLOUD DATA IN THREE DATASETS

Data set	Vegetation Index	Threshold	OA (%)	Precision (%)	Recall (%)	F1 (%)
Cambridge_18	RI	1.142	88.73	83.01	85.53	84.25
	VARI	0.168	80.82	80.26	65.83	72.33
	VDVI	0.117	92.46	95.98	83.70	89.42
	ExG	0.170	92.35	96.11	83.29	89.24
	ExG-ExR	0.071	91.75	93.17	84.54	88.65
	NGBDI	0.169	91.81	**95.57**	82.31	88.45
	LRCI	0.976	**93.04**	93.56	**88.03**	**90.71**
Birmingham_12	RI	1.142	91.63	84.59	96.68	90.23
	VARI	0.168	62.88	52.07	90.59	66.12
	VDVI	0.117	88.70	87.63	83.53	85.53
	ExG	0.170	88.24	87.77	82.04	84.81
	ExG-ExR	0.071	88.12	85.96	84.03	85.00
	NGBDI	0.169	88.60	88.04	82.73	85.30
	LRCI	0.976	**93.88**	**88.46**	**97.41**	**92.72**
Cambridge_25	RI	1.142	94.69	92.57	96.64	94.56
	VARI	0.168	86.72	88.79	82.60	85.58
	VDVI	0.117	96.11	98.04	93.71	95.83
	ExG	0.170	95.97	**98.12**	93.35	95.67
	ExG-ExR	0.071	96.24	97.61	94.43	96.00
	NGBDI	0.169	94.54	97.81	90.59	94.06
	LRCI	0.976	**96.31**	94.00	**98.55**	**96.22**

betas is (0.9, 0.99), respectively. We conduct 500 epochs on the Cambridge_28 dataset. Besides, the binary classification cross-entropy loss is used as the loss function samely.

Fig. 4 illustrates the curves of loss and accuracy with different epochs during training, in which the loss value gradually stabilized after 50 epochs, and the accuracy value gradually stabilized after 100 epochs. The model parameters are determined when the accuracy reaches the highest, based on which, we obtain parameters $w_1 = 0.0471$, $w_2 = -0.0924$, $w_3 = 0.0807$, $w_0 = -0.976$, thus, the classification decision boundary deduced from the training dataset is described as $0.0471^*R - 0.0924^*G + 0.0807^*B - 0.976 = 0$, accordingly, the function $\mathrm{LRCI}(R, G, B)$ and the threshold value δ_{vi} can be described as

$$\mathrm{LRCI(R, G, B)} = 0.0471R - 0.0924G + 0.0807B, \delta_{vi} = 0.976. \tag{8}$$

Based on (7) and (8), the input points with RGB information are classified as vegetation points or nonvegetation points.

C. Classifying the Nonground Points Into Vegetation Points and Nonvegetation Points

As described earlier, LRCI will be used to classify the nonground points into vegetation points and nonvegetation points, in order to assess the performance of proposed LRCI, other six classic vegetation indices [28], [29], [30] (see Table II) are introduced for comparison. While the training points only have RGB information, thus, RGB information is utilized to calculate the color vegetation indices. Prior to calculate the indices, a color space normalization for each point is performed as following:

$$r = \frac{R}{R+G+B}\ g = \frac{G}{R+G+B}\ b = \frac{B}{R+G+B} \tag{9}$$

where R, G, and B stand for the actual RGB values from the point cloud based on each RGB channel of the point.

As the metrics to evaluate the performances of different indices, overall accuracy (OA), precision, recall, and F1 score are applied to assess the performance of LRCI and other compared indices. Equations are described as follows:

$$\mathrm{OA} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}} \tag{10}$$

$$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{11}$$

$$\mathrm{precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{12}$$

$$\mathrm{F1} = \frac{2 \times \mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{13}$$

where TP is the true positive prediction (correctly labeled vegetation points), FP is the false positive prediction (points that are mislabeled as vegetation points), TN is the true negative prediction (points that are correctly labeled as nonvegetation points), and FN is the false negative prediction (points that are mislabeled as nonvegetation points or labeled as missed vegetation points).

Based on the input point cloud data, the six classic vegetation indices for each point are calculated, and the histogram of each calculated index dataset is drawn, taking the value at the bottom position as the initial threshold, then the optimal threshold is obtained iteratively by the global optimal method for each index, the details threshold-deciding method can be referred from Zhou et al. [31]. The obtained optimal threshold of each index is used to classify point cloud data into vegetation points and nonvegetation, in each calculated index, if $index\ value > threshold$, then the corresponding point belongs to the vegetation point, otherwise the point belongs to the nonvegetation point. For the LRCI, the threshold had been provided by (8), the thresholds for the aforementioned indices are listed in Table III. Here, three different datasets, including Cambridge_18 scene, Cambridge_25 scene, and Birmingham_12 scene in SensatUrban, are applied to test the classification performance of the proposed LRCI. From Table III, the highest values are highlighted in bold. We can conclude that LRCI has made achievements in the accuracy

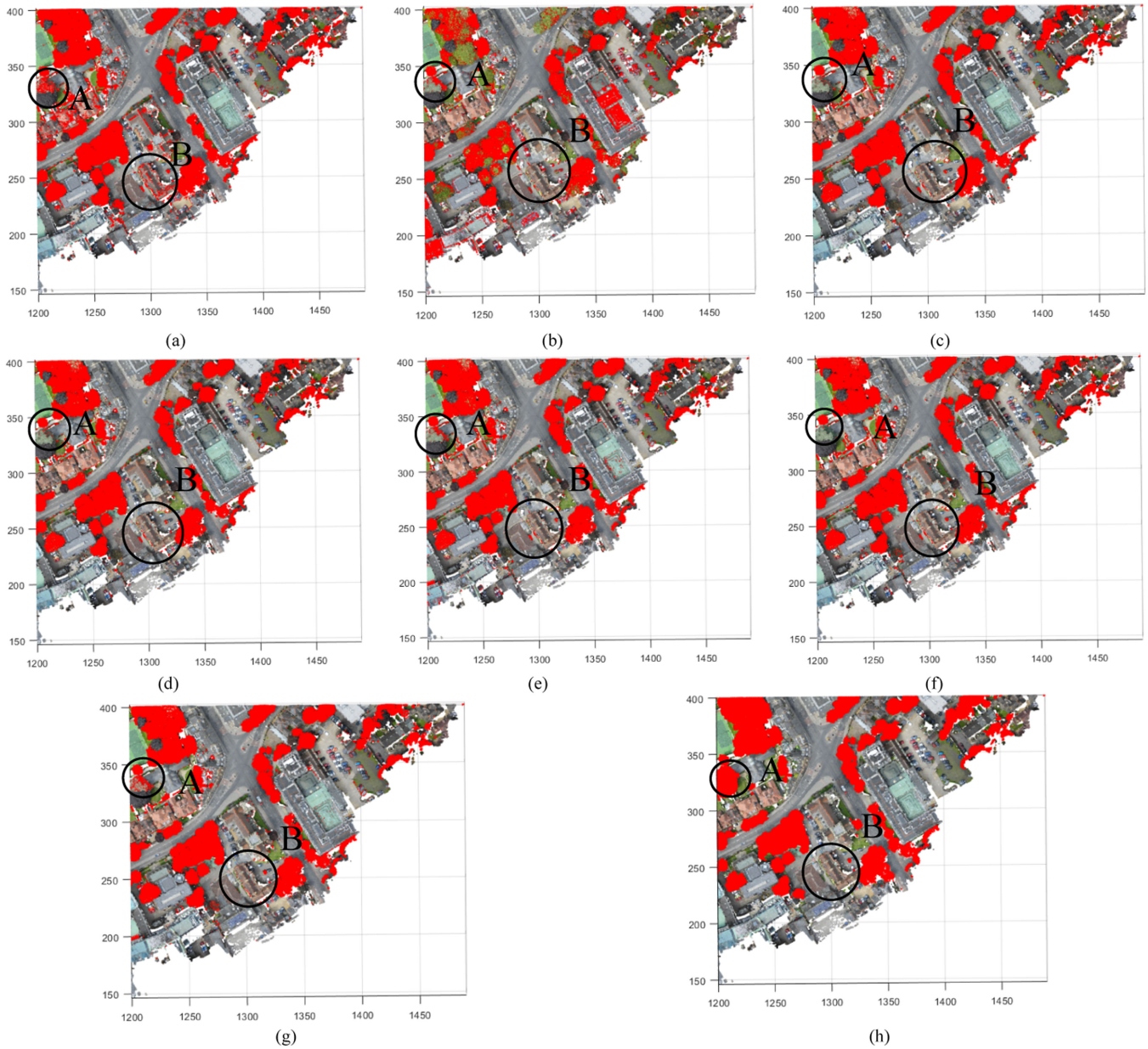

Fig. 5. Visualization of classification of vegetation points and nonvegetation points by different index in the Cambridge_18 scene. (a)–(g) Classification results by RI, VARI, VDVI, ExG, ExG-ExR, NABDI, and LRCI, respectively. (h) Ground truth of points (red points represent the vegetation points). (a) RI. (b) VARI. (c) VDVI. (d) ExG. e) ExG-ExR. (f) NABDI. (g) LRCI. (h) Ground Truth.

compared with the other six indices on the three datasets. On the Cambridge_18 scene, LRCI achieves the highest OA, recall, and F1 score, and gets higher precision. The classification effects of RI and VARI are poor, the reasons may be that the vegetation index histograms of vegetation and nonvegetation points overlap, which makes the optimum threshold difficult to be determined. Other classic indices have slightly better results, but recall rates are low, mostly below 85%. For example, the accuracy and precision of the ExG index are as high as 92.35% and 96.11%, respectively, but the recall rate is only 83.29%. The similar conclusions can be drawn from the other two datasets. We also visualized the classification results. Fig. 5 shows classification results of vegetation points and nonvegetation points by different index on the Cambridge_18 scene. Seen from the visualization results, we can clearly notice that vegetation points are misclassified as nonvegetation points (red) in area A and mislabeled nonvegetation points as vegetation points in area B in the six classic indices, and while LRCI corrects this error to a certain extent, most points labeling of the vegetation are closer to the real value. To sum up, the LRCI has the best performance of classifying the points into vegetation and nonvegetation points quantitatively and visually.

D. Point Cloud Clustering Results

Through Section III-C, the points in the Cambridge_18 scene had been classified as vegetation and nonvegetation points based on the LRCI, the classification results will be taken as the

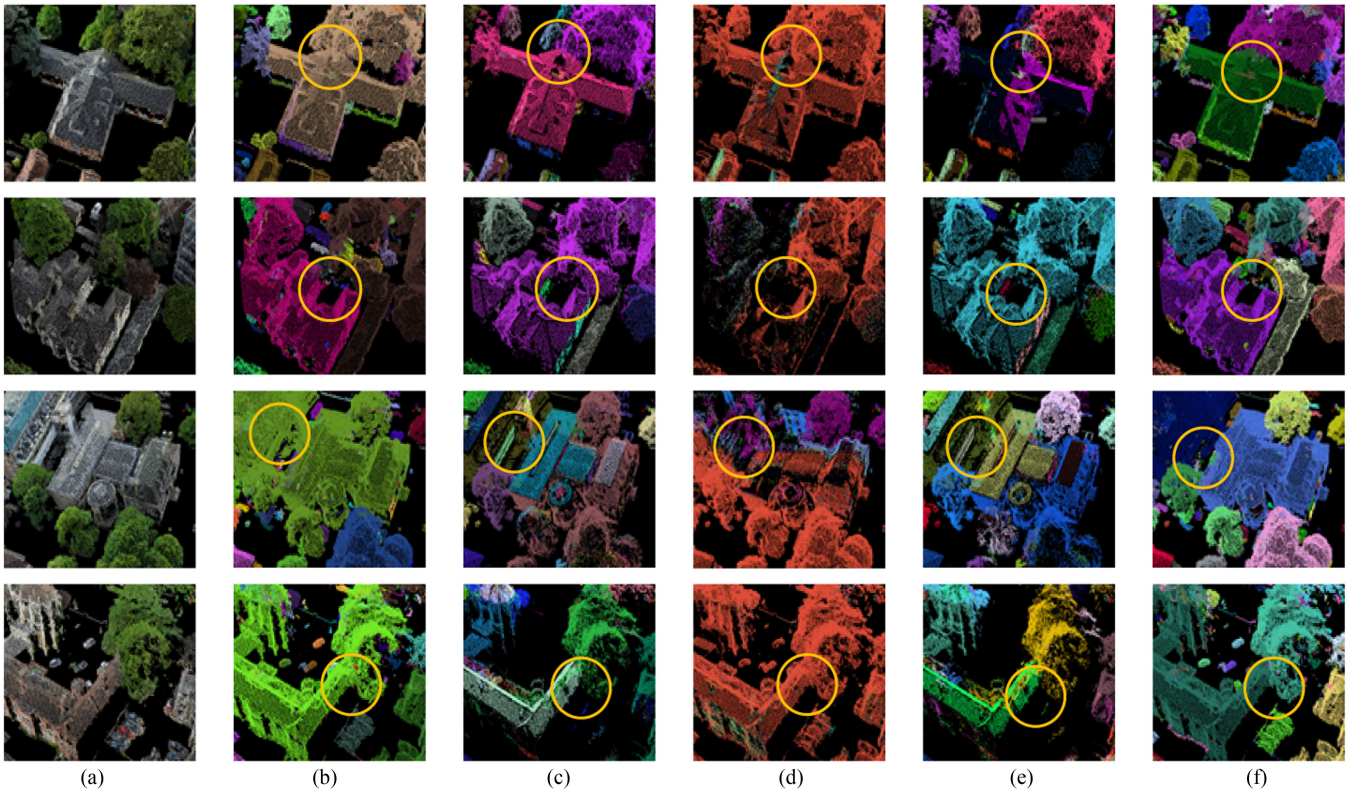

Fig. 6. Visualization of clustering results by different strategies of Cambridge_18 scene. (a) Original point cloud. (b) and (c) Results of DBSCAN clustering using xyz and xyz + RGB features, respectively. (d) and (e) Results of DBSCAN clustering by using PCA to reduce the xyz + RGB features to two-dimension and five-dimension, respectively. (f) Results of CC_DBSCAN clustering based on XYZ features using LRCI as a constraint.

class constraint embed in the CC-DBSCAN (as described in Algorithm 1). To assess the effectiveness of proposed method, its performances are analyzed and compared with other approaches through the following experiments. The purity is used to quantitatively evaluate the effect of clustering different classes of points. Purity refers to the ratio of the sum of major class points in each cluster to the total number of points in all clusters and is calculated as follow:

$$\text{purity}(C, L) = \frac{1}{N} \sum_{k} \max_{j} |c_k \cap l_j| \tag{14}$$

where $C = \{c_1, c_2, \cdots, c_k\}$ is the set of clusters, k is the number of clusters, and $L = \{l_1, l_2, \cdots, l_j\}$ is the set of classes, j is the number of classes, and N represents the number of points in all clusters. The higher the cluster purity, the better the separate clustering effect. In addition, the computational efficiency is another important index for evaluating the clustering strategy.

The CC-DBSCAN is compared with the traditional DBSCAN with different input features. Table IV shows the comparison of results on the Cambridge_18 scene. In the "features" column, xyz denotes the input points' features only contains position information $(\mathrm{x}, \mathrm{y}, \mathrm{z})$, xyz + RGB denotes input points' features contains position information $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ and color information $(\mathrm{R}, \mathrm{G}, \mathrm{B})$, *xyz* + RGB (2-D) denotes that input points' features contains the first two PCs of the results produced by principal component analysis (PCA) method based on the position information $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ and color information $(\mathrm{R}, \mathrm{G}, \mathrm{B})$, and xyz + RGB (5D) denotes that input points' features contains the first five PCs of the results produced by the PCA method based on the position information $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ and color information $(\mathrm{R}, \mathrm{G}, \mathrm{B})$. As we known, parameters *eps* and *minPts* are particularly important for the DBSCAN clustering algorithm. In this experiment, the optimal parameters of all strategies are determined by the method [31], and the results are listed in Table IV, and the best records are marked with bold.

TABLE IV
QUANTITATIVE COMPARISON OF DIFFERENT CLUSTERING STRATEGIES OF THE POINT CLOUD DATA IN CAMBRIDGE_18 SCENE

Strategy	Features	*eps*	*minPts*	Purity (%)	Computational cost (s)
DBSCAN	xyz	0.253	3	37.19	**22.50**
	xyz+RGB	4.807	28	46.55	107.34
	xyz+RGB (5-D)	4.339	29	43.40	105.51
	xyz+RGB (2-D)	0.704	41	30.12	32.40
CC_DBSCAN	xyz	0.253	3	**97.20**	24.25

The quantitative results are listed in Table IV. As seen from Table IV, amongst all strategies, CC_DBSCAN obtains the greatest purity and produces a value of 97.20%, with gains of 60.01%, 50.65%, 53.8%, and 67.08% over DBSCAN (xyz), DBSCAN (xyz + RGB), DBSCAN (xyz + RGB (5-D)), and DBSCAN (xyz + RGB (2-D)), respectively. And it is clear that time consumption increases with the added dimensions of input features, whereas CC_DBSCAN achieves the much higher

purity than that of DBSCAN (xyz) while the time required is a little more.

The visualization of clustering results by different strategies of Cambridge_18 scene is shown in Fig. 6(b)–(f). Due to the vegetation cover and only distance-based judgment used in the cluster, DBSCAN (xyz + RGB (2-D)) produces a result with plenty of wrong cluster points and shows the weakest performance amongst the five strategies. Only based the position features xyz, DBSCAN (xyz) obtains a higher accuracy than that of DBSCAN (xyz + RGB (2-D)) for the added input features, but many wrong clustered points remain in the result for only distance-based judgment is considered, which may lead to categorize them into one cluster by mistake for that the vegetation points are close to nonvegetation points. With the color information RGB is incorporated into input features, DBSCAN (xyz + RGB), DBSCAN (xyz + RGB (5-D)), and CC_DBSCAN produce accurate clustering results. As we known, the position information xyz and color information RGB belong to different metric spaces, thus, it can be seen from the clustering results of DBSCAN (xyz + RGB) and DBSCAN (xyz + RGB (5-D)), although the degree of clustering points belongs to different classed into the same cluster is alleviated to a certain, but some noise points are produced and there are still many mistaken clustering points. The CC_DBSCAN achieves the most satisfactory result, almost of the points are categorized into one cluster according to theirs real classes. The reason may be that the strategy of class constraint is incorporated in the DBSCAN, the false clustering can be prevented to a certain extent.

IV. Conclusion

In this article, an improved DBSCAN, namely CC-DBSCAN, is proposed for large-scale point cloud clustering. The proposed algorithm can overcome the drawbacks of the well-known DBSCAN by incorporating the strategy of class constraint in the DBSCAN. The CC-DBSCAN is effective in clustering large-scale point cloud with high accuracy and computational efficiency. This advantage is based on the definition of a new class constraint, which can prevent the core point and neighborhood points with different classes from being wrongly categorized into the same cluster. Experiments on the benchmark Sensat-Urban dataset were conducted to demonstrate the effectiveness of CC-DBSCAN. Compared with DBSCAN using different input features, it achieved the highest purity of point cluster, and reaches a value of 97.20%, furtherly, it takes only 24.25 s for clustering 2 million points. Therefore, the proposed CC-DBSCAN is an effective clustering method for large-scale point cloud clustering.

Generally, the research provides a new approach for large-scale point cloud clustering. Currently, the optimal parameters of CC-DBSCAN are determined by the method [31], we will conduct more research works on how to determine the optimal parameters. Moreover, our experiments are implemented on point cloud clustering, and we will further use the clustered results in the classification of point cloud, to achieve automatic interpretation of point cloud in future work.

Acknowledgment

The authors would like to thank the editor and anonymous reviewers whose insightful suggestions have significantly improved this article.

References

[1] R. B. Zhao, M. Y. Pang, and J. D. Wang, "Classifying airborne lidar point clouds via deep features learned by a multi-scale convolutional neural network," *Int. J. Geographical Inf. Sci.*, vol. 32, no. 5, pp. 960–979, Feb. 2018.

[2] Y. L. Guo, H. Y. Wang, Q. Y. Hu, H. Liu, L. Liu, and M. Benhamou, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Jun. 2021.

[3] D. K. Xu and Y. J. Tian, "A comprehensive survey of clustering algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Aug. 2015.

[4] K. Krishna and M. N. Murty, "Genetic K-means algorithm," *IEEE Trans. Syst. Man Cybern.*, vol. 29, no. 3, pp. 433–439, Jun. 1999.

[5] A. Mahdaoui and E. H. Shai, "3D point cloud simplification based on k-nearest neighbor and clustering," *Adv. Multimedia*, vol. 2020, 2020, Art. no. 8825205.

[6] Y. Yang, M. Li, and X. Ma, "A point cloud simplification method based on modified fuzzy c-means clustering algorithm with feature information reserved," *Math. Probl. Eng.*, vol. 2020, 2020, Art. no. 5713137.

[7] S. K. Lodha, D. M. Fitzpatrick, and D. P. Helmbold, "Aerial lidar data classification using expectation maximization," in *Proc. SPIE Electron. Imag.*, San Jose, CA, USA, 2007, pp. 1–9.

[8] P. Zhang, J. H. Li, X. Yang, and H. H. Zhu, "Semi-automatic extraction of rock discontinuities from point clouds using the ISODATA clustering algorithm and deviation from mean elevation," *Int. J. Rock Mech. Mining Sci.*, vol. 110, pp. 76–87, Oct. 2018.

[9] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Rec.*, vol. 25, no. 2, pp. 103–114, Jun. 1996.

[10] W. Wang, J. Yang, and R. R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *Proc. 23rd Int. Conf. Very Large Data Bases*, 1997, pp. 186–195.

[11] M. Thomas, "Non-parametric segmentation of ALS point clouds using mean shift," *J. Appl. Geophys.*, vol. 1, no. 3, pp. 159–170, Nov. 2007.

[12] J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm DBSCAN and its applications," *Data Mining Knowl. Discov.*, vol. 2, pp. 169–194, 1998.

[13] J. Shen, X. Hao, Z. Liang, Y. Liu, W. Wang, and L. Shao, "Real-time superpixel segmentation by DBSCAN clustering algorithm," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5933–5942, Dec. 2016.

[14] A. Karami and R. Johansson, "Choosing DBSCAN parameters automatically using differential evolution," *Int. J. Comput. Appl.*, vol. 91, no. 7, pp. 1–11, Apr. 2014.

[15] M. M. R. Khan, M. A. Siddique, R. Arif, and M. Oishe, "ADBSCAN: Adaptive density-based spatial clustering of applications with noise for identifying clusters with varying densities," in *Proc. 4th Int. Conf. Elect. Eng. Inf. Commun. Technol.*, 2018, pp. 101–111.

[16] C. Wang, M. Ji, J. Wang, W. Wen, T. Li, and Y. Sun, "An improved DBSCAN method for LiDAR data segmentation with automatic Eps estimation," *Sensors*, vol. 19, no. 1, pp. 1–26, Jan. 2019.

[17] W. Lai, M. Zhou, F. Hu, K. Bian, and Q. Song, "A new DBSCAN parameters determination method based on improved MVO," *IEEE Access*, vol. 7, pp. 104085–104095, 2019.

[18] X. Chen, W. Liu, H. Qiu, and J. Lai, "APSCAN: A parameter free algorithm for clustering," *Pattern Recognit. Lett.*, vol. 32, no. 7, pp. 973–986, May 2011.

[19] J. Hou, H. Gao, and X. Li, "DSets-DBSCAN: A parameter-free clustering algorithm," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3182–3193, Jul. 2016.

[20] S. S. Li, "An improved DBSCAN algorithm based on the neighbor similarity and fast nearest neighbor query," *IEEE Access*, vol. 8, pp. 47468–47476, 2020.

[21] J. Jang and H. Jiang, "DBSCAN++: Towards fast and scalable density clustering," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 3019–3029.

[22] Y. Chen et al., "KNN-BLOCK DBSCAN: Fast clustering for large-scale data," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 6, pp. 3939–3953, Jun. 2021.

[23] K. Mahesh Kumar and A. Rama Mohan Reddy, "A fast DBSCAN clustering algorithm by accelerating neighbor searching using groups method," *Pattern Recognit.*, vol. 58, pp. 39–48, Oct. 2016.

[24] Y. Chen, S. Tang, N. Bouguila, C. Wang, J. Du, and H. Li, "A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data," *Pattern Recognit.*, vol. 83, pp. 375–387, Nov. 2018.
[25] W. Zhang et al., "An easy-to-use airborne lidar data filtering method based on cloth simulation," *Remote Sens.*, vol. 8, no. 6, pp. 1–22, Jun. 2016.
[26] C. J. Paciorek, "Computational techniques for spatial logistic regression with large data sets," *Comput. Statist. Data Anal.*, vol. 51, no. 8, pp. 3631–3653, May 2007.
[27] Q. Y. Hu, B. Yang, S. Khalid, W. Xiao, N. Trigoni, and A. Markham, "Towards semantic segmentation of urban-scale 3D point clouds: A dataset, benchmarks and challenges," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4977–4987.
[28] A. Ö. Ok, "Robust detection of buildings from a single color aerial image," in *Proc. GEOBIA*, Calgary, AB, Canada, 2008, Art. no. 6.
[29] G. E. Meyer and J. C. Neto, "Verification of color vegetation indices for automated crop imaging applications," *Comput. Electron. Agriculture*, vol. 63, no. 2, pp. 282–293, Oct. 2008.
[30] M. Du and N. Noguchi, "Monitoring of wheat growth status and mapping of wheat yield's within-field spatial variations using color images acquired from UAV-camera system," *Remote Sens.*, vol. 9, no. 3, pp. 1–14, Mar. 2017.
[31] Z. P. Zhou, J. F. Wang, S. W. Zhu, and Z. W. Sun, "An improved adaptive and fast AF-DBSCAN clustering algorithm," *CAAI Trans. Intell. Syst.*, vol. 11, no. 1, pp. 93–98, Feb. 2016.

Hua Zhang received the Doctoral degree in cartography and geographical information engineering from the China University of Mining and Technology, Xuzhou, China, in 2012.

He is currently an Associate Professor in GIS and remote sensing with the School of Environment and Spatial Informatics, China University of Mining and Technology. He has authored or coauthored more than 40 peer-reviewed articles in international journals, such as IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING and *ISPRS Journal of Photogrammetry and Remote Sensing*. His current research interests include multi/hyperspectral and high-resolution remotely sensed images processing, uncertainty in classification, pattern recognition, and remote sensing applications.

Zhenwei Duan received the B.S. degree in geomatics engineering from Nanjing Tech University, Nanjing, China, in 2020. He is currently working toward the master's degree in geomatics engineering with the School of Environment and Spatial Informatics, China University of Mining and Technology, Xuzhou, China.

His research interests include deep learning, object detection, and point cloud segmentation.

Nanshan Zheng received the Ph.D. degree in urban environment from Kyoto University, Kyoto, Japan, in 2009.

He is currently a Professor with the School of Environment and Spatial Informatics, China University of Mining and Technology, Xuzhou, China. He has authored or coauthored more than 60 peer-reviewed journal or conference articles. His current research interests include GNSS-R, signal processing, and remote sensing of environment and disaster.

Yong Li received the B.S. degree in surveying engineering from the China University of Mining and Technology, Xuzhou, China, in 2001.

He is currently a Senior Engineer in GIS and remote sensing with the Sichuan Institute of Coal Field Geological Engineering Exploration and Designing, Chengdu, China. He has authored or coauthored more than ten peer-reviewed articles in international journals. His current research interests include high-resolution remotely sensed images processing and remote sensing applications.

Yu Zeng received the B.S. degree in surveying engineering from the China University of Mining and Technology, Xuzhou, China, in 2001.

She is currently a Senior Engineer in GIS and remote sensing with the Sichuan Institute of Coal Field Surveying and Mapping Engineering, Chengdu, China. She has authored or coauthored more than eight peer-reviewed articles in international journals. Her current research interests include spatial data quality control and remote sensing applications.

Wenzhong Shi received the Doctoral degree in philosophy from the University of Osnabrück, Vechta, Germany, in 1994.

He is currently a Professor in GIS and remote sensing with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong. He has authored or coauthored more than 400 research articles (including more than 100 SCI papers) and 10 books. His current research interests include GIS and remote sensing, uncertainty and spatial data quality control, and image processing for high-resolution satellite images.

Prof. Shi was the recipient of the State Natural Science Award from the State Council of China in 2007 and The Wang Zhizhuo Award from the International Society for Photogrammetry and Remote Sensing in 2012.