IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 18, 2020, accepted December 17, 2020, date of publication December 21, 2020,
date of current version December 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3046313

An Efficient Voxel-Based Segmentation Algorithm Based on Hierarchical Clustering to Extract LIDAR Power Equipment Data in Transformer Substations

JIANLONG GUO[1], WEIXIA FENG[1], JIANG XUE[1], SHAN XIONG[1], TENGFEI HAO[1], RUIHENG LI[2], AND HUBEN MAO[2]

[1]Training and Evaluation Center, Guangdong Power Grid Company, Ltd., Guangzhou 510630, China
[2]China Southern Power Grid Digital Grid Research Institute, Guangzhou 510530, China

Corresponding authors: Ruiheng Li (liruiheng@cqu.edu.cn) and Huben Mao (17683754@qq.com)

This work was supported by the Science and Technology Project of Guangdong Power Grid Company, Ltd., under Grant 038700KK52170007.

ABSTRACT Light detection and ranging (LIDAR) scanning is a common method of substation scene modeling that extracts point clouds of electrical equipment from the point cloud scene of a substation. The extraction effect is limited by uncertainty regarding the noise level, nonuniform point cloud density, and the computational complexity. In this paper, we propose a point cloud extraction solution for electrical equipment models. First, a statistical analysis of substation ground elevation is performed to obtain the point clouds of devices at the feature height and remove large numbers of redundant underground point clouds. Second, based on the statistically derived power equipment feature heights, the point cloud data are sliced according to the featured elevation intervals. Based on voxelization, the point cloud slices are then clustered using horizontal hierarchical clustering. The clustering results at different elevation intervals are then reclustered using vertical hierarchical clustering. Finally, we use filters combined with the DBSCAN algorithm to perform fine segmentation on the point cloud data. The results show that our slice clustering approach reduces the computational burden involved in point cloud processing, and the comprehensive clustering strategy ensures the accuracy of the clustering results.

INDEX TERMS Clustering methods, point cloud segmentation, smart grid, substation modeling.

I. INTRODUCTION

A 3D model can be built by a light detection and ranging (LIDAR) system after obtaining 3D laser point clouds of electrical equipment in a substation. These models are widely used in the power industry during grid construction and operation [1]–[3]. In the substation modeling field, the solutions commonly used in production are often manually compared to the ground truth by combining individual field photos and CAD models, which is a burdensome and difficult task because of the complex structures. Fast segmentation is a key technology that can improve modeling efficiency [4], [5]. However, few studies on fully automatic segmentation methods [6] for point cloud scenes in electric substations have been reported.

Point cloud segmentation in substation scenes typically lacks large numbers of prior models [7], [8]. Density-based and distance-based clustering methods are commonly used methods for segmenting point clouds [9], [10]. The density-based clustering method can remove noise based on the density differences between the noisy point clouds and the target point cloud and can obtain the target point cloud cluster, which has a higher density [11]. However, when the density of the noise point cloud is close to or greater than the density of the target point cloud, the density-based clustering method does not work well [12]. In addition, the density-based clustering method carries a high computational burden because it requires calculating the density relationship between the local and overall point clouds [13]. Because substation scenes

The associate editor coordinating the review of this manuscript and approving it for publication was Arash Asrari.

result in point clouds with data volumes that can reach tens to hundreds of gigabytes, traditional desktop computers may suffer from memory overflow problems. Thus, density-based clustering is not suitable for performing fast point cloud segmentation with large amounts of data.

The distance-based clustering method is computationally more efficient than the density-based clustering method, making it more suitable for fast segmentation of point clouds with large amounts of data [9]. However, the basic conditions under which distance-based clustering methods can achieve better clustering results are limited to the following: the distance between clustering targets must be large, and there should be no point cloud noise interference between them [13], [14]. When noisy point clouds exist between the targets, the clustering effect will be greatly reduced [15]. Considering that the continuity of the spatial distribution of substation equipment point clouds is better than the continuity of the noise point cloud, the noise can be identified and removed based on this spatial discontinuity. This means that the denoising of the substation point cloud scene and segmentation of the target device point cloud can be completed simultaneously using the distance information.

To improve the efficiency of segmenting substation point clouds scenes, we divide the segmentation process into two stages. First, the distance clustering method is used to perform rapid preliminary clustering. In the rapid preliminary clustering stage, we project the 3D scene point cloud to 2D feature planes of different heights and integrate the 2D clustering results through hierarchical clustering. This operation not only improves the computational efficiency by reducing the dimensionality from a three-dimensional to a two-dimensional clustering process but can also use vertical continuity to remove noise. Second, the density method is used to perform fine clustering, which further improves the accuracy of the segmentation results.

In the remainder of this article, we first review the existing clustering methods in Section II and explain the characteristics of the distance clustering method and the density clustering method used in our clustering algorithm. In Section III, we present the characteristics of our measured substation dataset and provide an overview of our framework. The detailed steps of the method are listed in Section IV, and the results of each step are shown in Section V. The effects of the proposed method are discussed in Section VI, including its anti-interference ability, segmentation accuracy and calculation efficiency. Finally, Section VII concludes the paper.

II. RELATED WORKS

At present, point cloud segmentation and extraction methods can be roughly divided into four categories [16], [17]: edge-based methods, region-based methods, model-matching methods and machine learning methods. Edge-based detection is an old but effective method. Although edge-based detection methods offer fast segmentation, they can produce inaccurate results in cases with common noise and uneven density in point cloud data [18], [19].

A. REGION-BASED METHODS

The region-based methods are more robust to noise than are the edge-based methods [20]. These methods use neighborhood information to combine neighborhood points with similar properties and obtain isolated regions, thereby enabling the discovery of the differences between different regions [21]. Seed segmentation is a region-based method that does not require much prior knowledge to acquire regional nodes. In this method multiple seed points are first selected; then, using a predetermined similarity criterion, each region grows by adding neighborhood points [22]. The disadvantage of this method is that it is both highly sensitive to background noise and requires lengthy executions time [17].

To address the time consumption problem in segmentation, some studies divide segmentation into two stages: coarse segmentation and detailed subdivision [23]. The coarse segmentation method first extracts the main target from the scene; then, the detailed segmentation is used as a refinement process to extract more detailed information about the target component. This clustering algorithm has low time complexity [17], but its segmentation accuracy is highly dependent on selecting the correct seed points.

To address the problem of algorithms being excessively sensitive to background noise, some studies have used a combination of normal and curvature information to smooth the constraints between different regions [22], [24]. This approach reduces the impact of background noise on the clustering results to a certain extent. However, these methods are very sensitive to inaccurate estimations of the initial seed region.

Therefore, the region-based methods typically depend strongly on whether the selected seed points are correct. Moreover, selecting seed points and controlling the growth process are both time consuming.

B. MACHINE LEARNING METHODS

In machine learning methods, unsupervised learning does not require much prior knowledge. K-means clustering and K-NN are the two most common classification algorithms in unsupervised learning, but these methods require predefining the number of clusters to create, and the cluster value is often unknown in substation scenarios [25], [26].

Density-based spatial clustering of applications with noise (DBSCAN) is an unsupervised learning method that does not require a preset number of clusters and can remove noise while clustering [9], [11]. Although DBSCAN algorithm is well-established and has been widely used on sample-limited datasets, the computational burden will be very heavy if DBSCAN is applied indiscriminately to large data sets [11], [27]. However, DBSCAN performs clustering by considering only the density information. Therefore, if the point cloud densities of different devices vary substantially and the noise density is close to the device density, DBSCAN will have only a limited effect [28], [29].

Hierarchical clustering is another common unsupervised learning method for clustering that attempts to divide the

dataset at different levels to form a treelike clustering structure [30]. Datasets can be divided into ''bottom-up'' aggregation strategies or ''top-down'' split strategies [31]. Hierarchical clustering is similar to the region-based methods, but is more flexible when operating in high-dimensional data spaces. The advantage of hierarchical clustering is that it can control the clustering of data sets at different scales; however, its disadvantage is that it requires large amounts of calculation [32], [33].

C. SUMMARY

The region-based methods have the highest computational efficiency when the noise level is small. However, without prior knowledge, a high noise level makes it very time consuming to select seed points and control the growth process. The biggest problem with unsupervised learning methods is the amount of calculation they require. In addition, when the density of the target point and the noise density are similar, the effect of this method is limited.

Based on the characteristics of these methods, we can first apply a hierarchical clustering method, which is similar to a region-based method, to reduce the dimensionality of the 3D scene point cloud data by limiting it to different 2D planes and then performing clustering to improve the computational efficiency. Point cloud noise can then be removed based on the vertical continuity via vertical hierarchical clustering. Then, because these preliminary clustering results reduce the noise level, the clustering results can be further processed using DBSCAN.

In view of the three problems with the above methods, we have carried out the following works:

(1) By analyzing the probability distribution of the ground point cloud, we obtain the ground elevation and remove a large number of redundant point clouds, which alleviate the problem of massive calculations.

(2) We use the vertical clustering method based on different feature heights to reduce the clustering from three-dimensional to two-dimensional clustering, which not only alleviates the massive computational burden but also removes the high-density point cloud noise and alleviates the uncertainty in the noise level.

(3) We use DBSCAN to further process the point cloud data to extract the point clouds of devices by clustering while removing the noise.

III. OVERVIEW OF OUR APPROACH

A. SUBSTATION SCENE OVERVIEW

1) FIELD DATASET

The point cloud of the substation scene selected in this paper comes from the 220 kV area of the Maohu Substation in Guangdong, China. The collection device is a Faro FocusS 70 laser scanner, and the observation method is ground observation. The accuracy of the point cloud is ± 1 mm. We use the Geomagic software (Faro software kit) to preprocess the data and register it with the data for each station to obtain the point cloud of the substation scene. This approach fully meets the requirements of substation modeling; the average accuracy of point cloud registration for the entire scene is 15 mm, which is well below 1% of the actual equipment size. A top view of the point cloud of the substation scene is shown in Figure 1a.

Electrical equipment can be divided into two categories: (1) isolated phase devices (Figure 2a-d) and (2) three-phase devices (Figure 2e-h). Isolated phase devices usually exist independent of other devices except for the wires at the top. Three-phase devices with connecting rods on a cement shelf exist between the different phase devices (Figure 2a, g, h).

The entire scene includes 235 devices, where the number of single-phase equipment is 190, the number of three-phase equipment is 45. We selected a subregion (Figure 1a) to show the detailed distribution of device point clouds and noise point clouds (Figure 1b). The yellow frame contains both three-phase and single-phase equipment, and there is a thin connecting rod in the middle of the three-phase equipment. The violet frame contains nine single-phase devices. The middle three are independent voltage transformers that are connected to other devices on both sides through the top wires.

2) POINT CLOUD DENSITY

We chose different neighborhood radii to determine the point cloud density (Figure 3). When the neighborhood radius is 0.03 m, the density threshold for distinguishing device point clouds from noise point clouds is approximately 200,000. A point cloud with a density greater than 200,000 occupies only a small part of the total device point cloud; thus, it is difficult for these point clouds to be used to correctly segment the device point cloud. Especially for the device point cloud in the yellow box, the neighborhood density of most points is below 10,000, which is similar to the neighborhood density of noise points.

When the neighborhood radius is 0.1 m, the density threshold for distinguishing the device point cloud from the noise point cloud is approximately 15,000. However, the continuity of the point cloud in the connection part of the three-phase device in the yellow frame is not good, and the three-phase device can easily become divided into three single-phase devices. Moreover, the wire point cloud density value on top of the 9 single-phase devices in the violet frame is greater than 20,000. If that is regarded as a valid device point cloud, these 9 single-phase devices will be connected to 3 three-phase devices, which leads to undersegmentation.

When the neighborhood radius is 0.5 m, the density threshold for distinguishing the device point cloud from the noise point cloud decreases to approximately 5,000. The continuity of the point cloud of the connection part of the three-phase equipment in the yellow frame is enhanced, but the oversegmentation problem it causes does not improve.

Therefore, the density of the point cloud in the connection part of the three-phase equipment is too similar to the density of the noise point cloud, and it is difficult to find a suitable neighborhood density threshold to separate them.

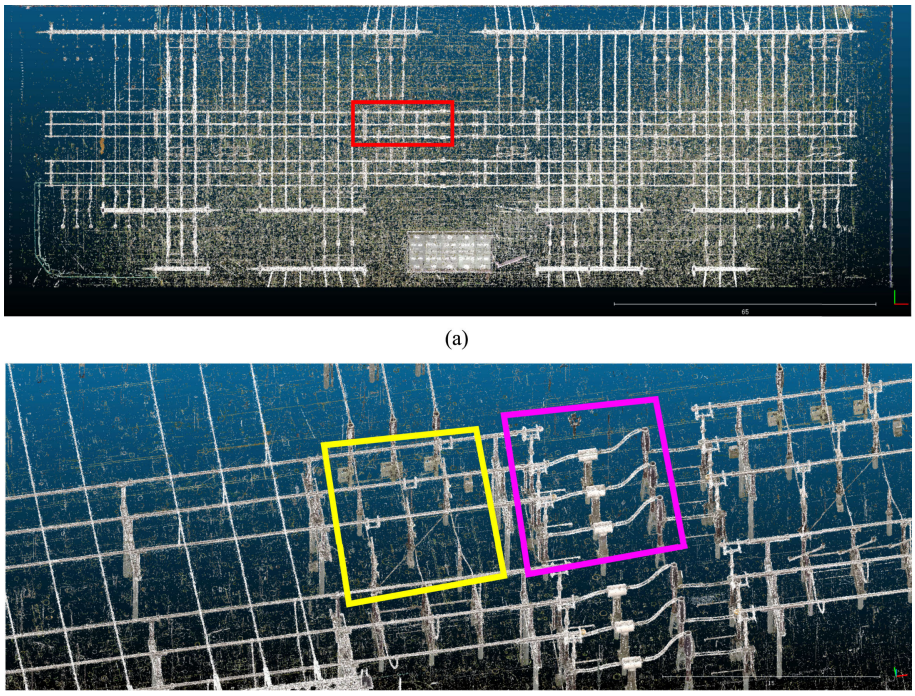

FIGURE 1. Measured point cloud data of the substation scene. (a) A top view of the point cloud of the substation scene. (b) The subregion delineated by the red frame in (a). The yellow frame contains both three-phase and single-phase equipment, and there is a thin connecting rod in the middle of the three-phase equipment. The violet frame contains nine single-phase devices.

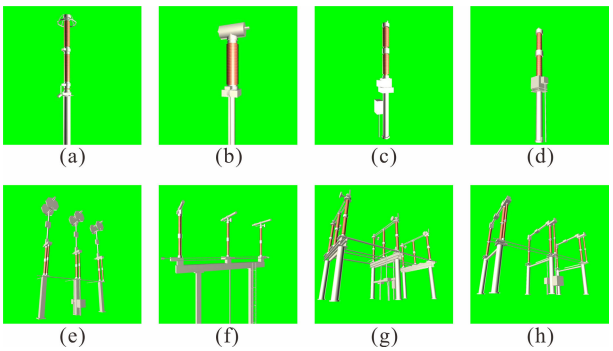

FIGURE 2. Model of electrical equipment. (a) Lightning rod, (b) Voltage transformer, (c)-(d) Current transformers, (e)-(h) Isolating switches.

This characteristic degrades the ability to accurately segment the point clouds of substation scenes.

B. SEGMENTATION WORKFLOW

The point cloud segmentation process of the substation scene is shown in Figure 4. The process can be divided into three steps: terrain correction, hierarchical clustering analysis and fine reclustering.

In the terrain correction stage, to obtain the ground elevation, we divide the entire substation scene into small grids and obtain the ground elevation corresponding to each small grid based on a statistical histogram. Then, invalid underground point clouds can be eliminated according to the ground elevation.

The hierarchical clustering analysis step can be divided into two steps: horizontal hierarchical clustering and vertical hierarchical clustering. First, point cloud slices at different feature elevation intervals are clustered horizontally. The clustering process uses voxel units to mesh the point cloud, which avoids the inefficiency of performing large numbers of point cloud operations. Then, we use vertical hierarchical clustering to reintegrate the results of each horizontal cluster.

In the fine reclustering stage, we apply DBSCAN to remove residual noise while extracting and segmenting fine-scale device point clouds with complex shapes.

IV. METHODOLOGY

A. TERRAIN CORRECTION

Assume that there are N points in the point cloud $\mathbf{P}$ of the substation scene; then, P is expressed as follows:

$$\mathbf{P} = \{x_i, y_i, z_i, c_i | i = 1, 2, 3, \cdots, N\}, \tag{1}$$

where x, y, z represent the pbosition coordinates, and c represents the color information of the point. We project the cloud $\mathbf{P}$ of the substation site on the Z axis to form a dataset $\mathbf{P}_z$ that can be expressed as

$$\mathbf{P}_z = \{z_i | i = 1, 2, 3, \cdots, N\}, \tag{2}$$

where the horizontal coordinate information (x and y) and color information are ignored in $\mathbf{P}_z$, and the number of points

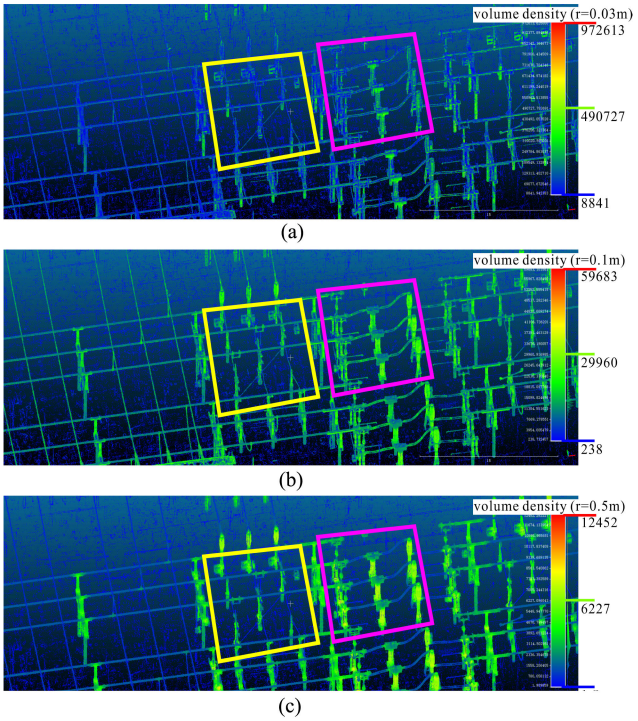

FIGURE 3. Measured point cloud data and its point cloud density in the subregion shown in Figure 2b. The point cloud density is the volume density, calculated by the CloudCompare [34]. The r is used to estimate the radius of the volume.

in $\mathbf{P}$ and $\mathbf{P}_z$ is the same as N. The histogram of the dataset $\mathbf{P}_z$ is calculated. The width of the statistical equal-width bins is dz. Histogram statistics can identify the bin with the largest number of points as the ground height interval because the ground point clouds are usually concentrated in a certain height interval.

This global statistical method directly counts the ground height of the entire substation area when the terrain of the substation is flat and the ground height remains unchanged. However, when the substation has undulating terrain, the ground height of the entire area does not truly reflect the ground height of all locations. In such cases, it is necessary to acquire more accurate statistics at various locations.

We divide the horizontal direction of the entire area into $n_x \times n_y$ small grids. There are N' points in the grid with sequence numbers (m_x, m_y). The dataset $\mathbf{P}'$ of these points can be expressed as follows:

$$\begin{aligned} \mathbf{P}'_{m_x,m_y} &= \{x_j, y_j, z_j, c_j\} \quad j = 1, 2, 3, \cdots, N' \\ m_x &= 1, 2, 3, \cdots, n_x \quad m_y = 1, 2, 3, \cdots, n_y, \end{aligned} \tag{3}$$

where n_x is the number of grids in the X direction and n_y is the number of grids in the Y direction. By projecting the point cloud $\mathbf{P}'$ onto the Z axis and ignoring the horizontal coordinate and color information, we can obtain $\mathbf{P}'_z$:

$$\mathbf{P}'_{z|m_x,m_y} = \{z_j | j = 1, 2, 3, \cdots, N'\}. \tag{4}$$

The data from different grids are then calculated as histogram statistics. The width of equal-width bins in statistics is still represented by dz. In the statistical results of the projected dataset $\mathbf{P}'_{z|mx,my}$ of each grid, the bin with the largest count can be identified as the ground height corresponding to the grid. We then traverse all the grids to obtain the ground height corresponding to the different grids. In some grids, the sparse ground point cloud will count the gantry height as the ground height. This wrong ground height is more than ten meters higher than the correct ground height in other areas, and can be regarded as outlier in the entire area. We replace the outlier by the interpolation result of the surrounding area. A flowchart of this terrain correction is shown in Appendix Table 3.

For calculating the ground height of a point cloud data as shown in Equation 2, the time complexity is $O(n)$. When we analyze the ground height according to the fine statistics of the voxelized grid, we need to

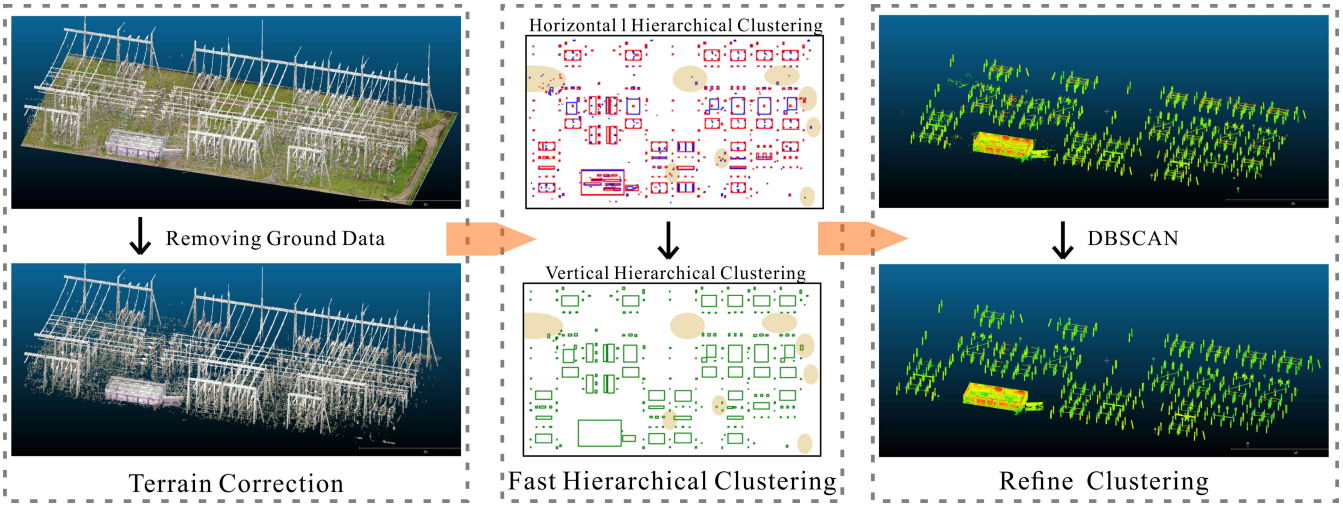

FIGURE 4. Workflow for extracting point clouds of each electrical equipment in the substation.

scan $Nx \times Ny$ the point clouds shown in Equation 3, and the time complexity is $O((Nx \times Ny)n)$. Considering that the processing of outliers will add c (constant) calculation steps with $O(1)$, the time complexity is $O((Nx \times Ny)(n + c))$, which is approximately equal to $O(n)$.

The equipment height usually ranges from 10 to 20 m [35]. To make the accuracy range of the ground height close to 1% of the height of the device, a floating error of ± 0.15 m is allowed. The statistical height interval dz is set according to the absolute floating error interval, which is 0.3 m. The upper and lower boundaries of the bins in the histogram are the maximum and minimum values of z in the statistical dataset. Since the ground slope in the substation does not exceed 5 degrees [35], an error of 0.15 m in the longitudinal direction corresponds to a range of 3 m in the transverse direction. We set the size of $n_x \times n_y$ small grids to 3×3 m^2.

B. FAST HIERACHICAL CLUSTERING STRATEGY

1) VOXELIZED CLUSTERING

To speed up the clustering process, the fewer points that are involved in the clustering process, the better. We use voxels to represent the point cloud of a region. This process is similar to the downsampling process. When these voxels are sufficient to represent the point cloud of the overall substation scene, the clustering results are more accurate. Thus, selecting an accurate voxel size is essential for acquiring the positions of electrical equipment in the substation space when using the clustering algorithm.

The center position of each voxel is regarded as the position of the voxel unit, and the voxel grid can be regarded as a set of points. We use distance-based clustering under a specific distance threshold (Figure 5). The distance threshold in this clustering algorithm is the desired minimum distance between clusters. Figure 5 shows that when the threshold distance is large, discrete points can easily be mistaken for continuous points. The goal is to include as few discrete noise signals in the continuous device point cloud as possible when clustering a substation scene point cloud. Consequently, the logical approach is that the smaller the distance threshold is, the better. However, at the same time, we need to prevent occlusions in the point cloud collection from causing discontinuities in the device point cloud, even if the possibility of such occlusions is small. Therefore, based on comprehensive consideration, we set the threshold distance to twice the size of a voxel unit.

All the voxels in the point cloud are regarded as valid voxel units and will be counted. The voxel size is set to 0.2 m. The lateral span of electrical equipment is usually between 5 and 20 m [35], which is dozens of times the size of the voxel unit. The device with the smallest lateral span is an insulating pillar with a lateral size of 1.5-2 m; even that is at least 5 times the size of a voxel unit. This means that a 0.2 m voxel size is sufficient to determine the locations of devices in space.

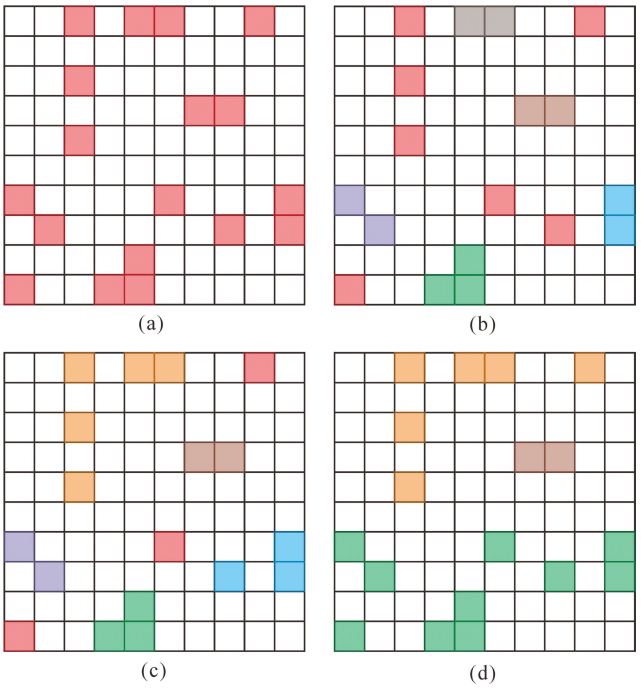

FIGURE 5. Clustering results when searching adjacent voxels: (a) voxels containing point clouds; (b) clustering results when the adjacent distance is $\sqrt{2}$ l; (c) clustering results when the adjacent distance is 2 l; and (d) clustering results when the adjacent distance is 3 l. Elements of the same color belong to the same cluster.

2) HORIZONTAL HIERARCHICAL CLUSTERING

The voxel grid clustering results may include many special clusters. Elements in special clusters are distant from each other (Figure 5c, d); however, the bounding boxes of these clusters may be adjacent or even contained in another bounding box (Figure 6a, b). For this cases, we recombine the special clusters into one cluster. This recombination process can be regarded as hierarchical clustering based on the original distance clustering procedure (Figure 5c, d). The hierarchical clustering strategy merges the clusters whose outer bounding boxes intersect, are adjacent, or have a containing relationship.

We use the following recombination algorithm (Table 1) to cluster the bounding boxes with a hierarchical strategy. This hierarchical clustering strategy (Figure 6c, d) is an extension of the voxel clustering strategy that also reduces the number of clusters. The shapes of the bounding boxes conforms to the distribution law of electrical equipment in a substation scene.

The cross condition for determining whether box $A = \{x_{min}, x_{max}, y_{min}, y_{max}\}$ intersects with box $B = \{x'_{min}, x'_{max}, y'_{min}, y'_{max}\}$ in a certain way can be expressed as Equation 5.

The horizontal hierarchical clustering repeatedly cross-validates the boxes in the box set. The determination of cross-validation in the loop is shown in Equation 5, and the time complexity is $O(1)$. The repeated cross-validation process is a nested loop (Table 1). As shown in Table 1, the outer loop is shown in step 2, and the sub loop is shown in step 2.2. The number of sub-loops decreases as the collection

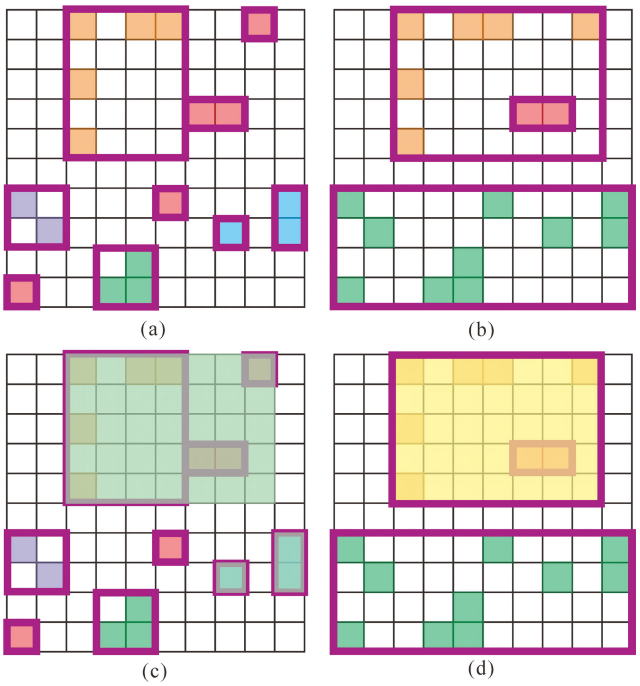

FIGURE 6. Clustering frames when searching adjacent voxels: (a) cluster bounding boxes for the adjacent distance are 2 l; (b) merging bounding boxes for the adjacent distance are l, a small box is contained in a larger one; (c) cluster bounding boxes for the adjacent distance are 3 l; (d) merged boxes where one or more boxes were contained by another.

TABLE 1. Recombination algorithm.

Input: boxes $B=\{B_i = 1,2,3,\ldots,N\}$, $B_j=\{x_{i\_min}, x_{i\_max}, y_{i\_min}, y_{i\_max}\}$
Output: boxes $B'=\{B'_j = 1,2,3,\ldots,M\}$, $B'_j=\{x_{j\_min}, x_{j\_max}, y_{j\_min}, y_{j\_max}\}$
1. Prepare the target box B' by adding B_1 to B'_1 and deleting B_1 from B.
2. Begin a loop for B. Set j=0 at the beginning. Set the terminal condition for the length of B to 0.
2.1 Select the last element of B' as a comparative box b. Create a set B_{temp} and put b into B_{temp} as the 1st element.
2.2 Begin a loop for B. Set i=0 at the beginning. The terminal condition is that the loop is at the end of B.
2.2.1 Determine whether B_i intersects with b. When an intersection is present, then add B_i to B_{temp} and delete B_i from B.
2.3 Determine whether the length of B_{temp} has increased. If so, calculate the maximum outer bounding box (merging the newly added boxes into one box) and add it to B.
2.4 When the length has not increased, add B_1 into B'_1 and remove B_1 from B.

size decreases, and the time complexity of the outer loop is $O(n\log n)$.

$$\begin{aligned} dx_1 &\le dx_2 \quad and\ dy_1 \le dy_2 \\ dx_1 &= \left|x_{min} + x_{max} - x'_{min} - x'_{max}\right| \\ dx_2 &= (x_{max} - x_{min}) + \left(x'_{max} - x'_{min}\right) \\ dy_1 &= \left|y_{min} + y_{max} - y'_{min} - y'_{max}\right| \\ dy_2 &= (y_{max} - y_{min}) + \left(y'_{max} - y'_{min}\right). \end{aligned} \tag{5}$$

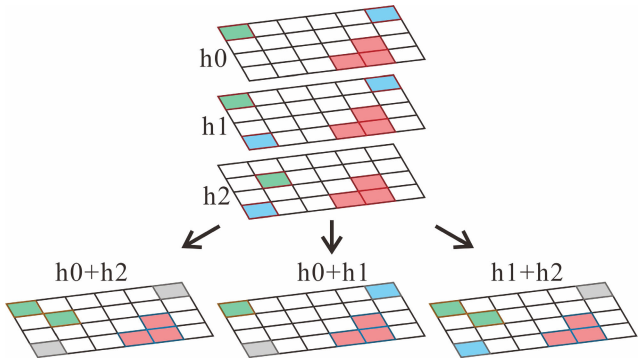

FIGURE 7. Different vertical hierarchical clustering results from slices at different feature elevations.

3) VERTICAL HIERARCHICAL CLUSTERING

We can project the slices of point cloud data of different heights onto the same plane for hierarchical clustering using a vertical constraint strategy. We divide the point cloud data into three height ranges: h0, h1 and h2. In each height range, the point cloud data are clustered by the horizontal clustering algorithm for the voxel grid. If the corresponding grids belong to clustered grids in different height intervals at the same horizontal grid positions, then the clusters can be recombined into one group. In this case, it can be seen that point clouds belonging to the same cluster have better continuity in the different height intervals (red grids, Figure 7).

However, noise data should not have the same vertical continuity. Noise signals may have continuity within a certain height range but poor continuity over a larger height range, as illustrated by the blue grids in Figure 7. The horizontal hierarchical clustering results of the discontinuous height intervals are used for reclustering, which has a stronger ability to suppress noise. Therefore, we can choose the clustering results in discontinuous slices for vertical hierarchical clustering (Figure 7).

If the results of the horizontal hierarchical clustering at different height intervals show intersection, adjacency, or inclusion relationships, we consider them as the same cluster and use the algorithm shown in Table 1 to recluster the boxes. This case is depicted as the clustering process for the green grids in Figure 7. This process repeatedly calls the recombination algorithm (Table 1), and the number of calls is constant and determined by the number of horizontal slices. The time complexity of this process is $O(n\log n)$.

To avoid undersegmentation caused by the point cloud of connecting wires at the top of each connected equipment (shown in the violet frames in Figure 3), we cluster the point clouds below the wire height. Considering the vertical continuity of each piece of equipment, when the equipment point clouds under the wire are clustered accurately, the horizontal range of each cluster will represent the horizontal range of the equipment. The height of the wire connecting each piece of equipment is determined by the height of the device itself and is usually no more than 5 m [35].

We conservatively select the point clouds whose height does not exceed 4 m as the objects to be clustered. At the same time, to further prevent any influence from individual ground point clouds that have not been removed, the lower limit of the height of the clustered point clouds is set to 1 m. Therefore, the three projection slices in Figure 7 are equally divided from point clouds with heights of 1–4 m. The three slices represent point cloud data at three different height intervals: 1–2 m, 2–3 m, and 3–4 m. Here, we use the previously estimated ground height as the starting 0 m height.

C. FINE RECLUSTERING

After implementing the abovementioned hierarchical clustering strategy, a small number of noise signals remain in the clustering results, but their noise density is relatively small and easy to identify. To further remove this remaining noise, we use DBSCAN to recluster the point cloud, which still possesses a small amount of noise. The DBSCAN algorithm is advantageous because there is no need to prespecify the number of clusters, which will reflect the unpredictable number of devices in the substation. Another feature of the DBSCAN algorithm is that it can judge abnormal values during clustering, which meets the requirement for removing noise from the data.

Although there are some state-of-the-art methods that are better than DBSCAN [36]–[38], when the results of fast hierarchical clustering are sufficiently accurate, traditional DBSCAN can already achieve better clustering results. If the fine re-clustering based on DBSCAN can obtain good results, it cannot only show the effectiveness of our framework, but also show that our framework can provide preprocessing solutions without these state-of-the-art methods.

The parameters required by the DBSCAN algorithm are the minimum intercluster distance ε and the number of minimum points in each cluster *min_pts*. The minimum distance between clusters refers to the smallest horizontal distance between substation equipment. The horizontal distance between equipment is often greater than 5 m, and the minimum distance is typically between 1.5 and 2 m. Assuming that the horizontal distance between the devices in each cluster is greater than 1.0 m, this fully complies with the safety distance specifications for placing these devices [35]. Therefore, we set the minimum intercluster distance (ε) in DBSCAN to 0.7 m.

The size of equipment is typically on the order of meters, while the distance between the points in the point cloud representing the equipment is on the order of centimeters. This means that in a three-dimensional space, a point cloud cluster representing equipment contains hundreds or thousands of points. Thus, we conservatively set the minimum number of points in each cluster (*min_pts*) in DBSCAN to 10, which is sufficient to meet the clustering requirements.

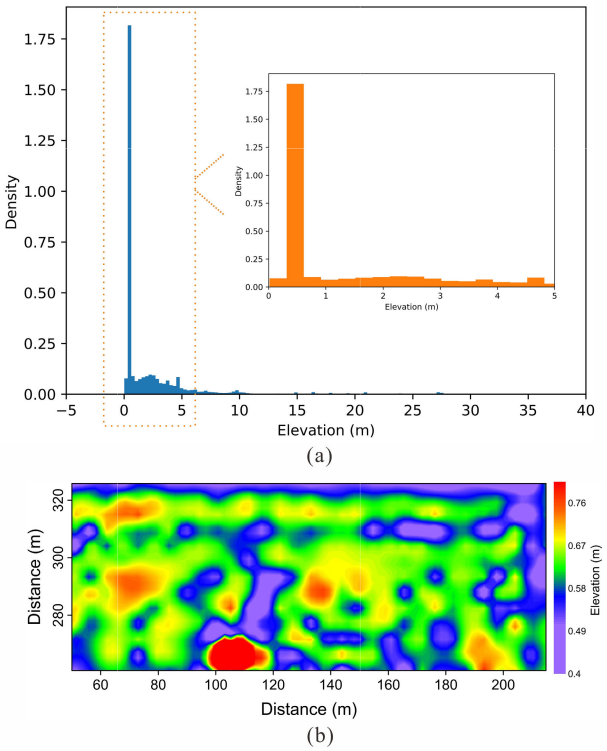

FIGURE 8. Ground surface elevation: (a) Histogram analysis of the ground elevation; (b) contour map of the ground elevation. The contour map formed by the ground height values of each grid after kriging interpolation is shown in Figure 8b.

V. RESULTS

A. REMOVING GROUND DATA

The number of substation scene point clouds (Equation 2) that appear at different height intervals were counted as shown in Figure 8a. From the histogram, the number of points counted in bins from 0.3–0.6 m is much larger than the number of points in the other bins. Consequently, this interval can be considered as the height interval where the ground is located. The ground height estimation for the entire substation scene reflects the ground height at a global scale.

Considering the height variations of the ground, to obtain a more precise estimation of the ground height, we count the number of substation scene point clouds (Equation 4) that fall into different height intervals and assume that the ground height corresponding to each grid is the height interval corresponding to the peak value in each histogram. The ground height value can be obtained by taking the middle value of the height interval. The contour map formed by the ground height values of each grid after kriging interpolation is shown in Figure 8b.

The results after removing the ground points (Figure 9) show that overall statistics of the ground height of the substation without grid division would be unable to obtain accurate ground height information and that using the overall statistics would not completely remove the below-ground point cloud data (Figure 9a). Thus, ground height

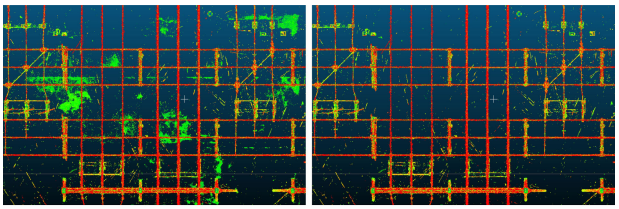

FIGURE 9. Deleting underground point cloud data: (a) the ground height is obtained from the statistical analysis of the overall point cloud data, and the residual ground data are marked by the white frames; (b) the ground height is obtained from the statistical analysis of the gridded point cloud data.

statistics based on the grid division provide more accurate ground height information. The result after removing the ground point clouds by this method are shown in Figure 8b. The removal is complete and clean. At this point, all ground information including the cement roads surface is removed.

The change in the point cloud dataset size after removing ground point cloud data is shown in Appendix Table 4. Removing the ground data reduces the number of points substantially (by approximately 40%), which greatly improves the computational efficiency of subsequent processing. The vertical starting heights of the point clouds do not all start at 0 m. In the substation scenario, the vertical size of each device is the relative height information. To obtain the point cloud at the characteristic height, it is necessary to obtain the absolute height value starting from 0 m. Therefore, another function of evaluating the ground height is to obtain the starting height of the slicing operation in hierarchical clustering.

B. FAST CLUSTERING WITH HIERACHICAL STRATEGY

We extracted the original point cloud data with an elevation of less than 4 m into three slices at one-meter intervals. The H0 slice data correspond to point cloud data with elevations of 1–2 m; the H1 slice data correspond to point cloud data with elevations of 2–3 m; and the H2 slice data correspond to point cloud data with elevations of 3–4 m. The original point cloud data for slices H0 and H2 are shown in Figure 10a and b. Each slice is processed by the voxel grid clustering algorithm and the horizontal hierarchical clustering algorithm to obtain the outer bounding boxes of each cluster.

Then, the boxes from the different height slices can be reclustered by the vertical hierarchical clustering algorithm, yielding a new clustering result on the horizontal ranges of equipment. The boxes in H0 and the boxes in H2 are integrated by the vertical hierarchical clustering algorithm to obtain the clustering result H02 (Figure 10c), which still contains multiple outer bounding boxes. Then, these new boxes are reintegrated with the boxes in H1 through the vertical hierarchical clustering algorithm (Figure 10d).

The number of noise point clouds in H02 is smaller than the number of noise point clouds in H0 and H2. This shows

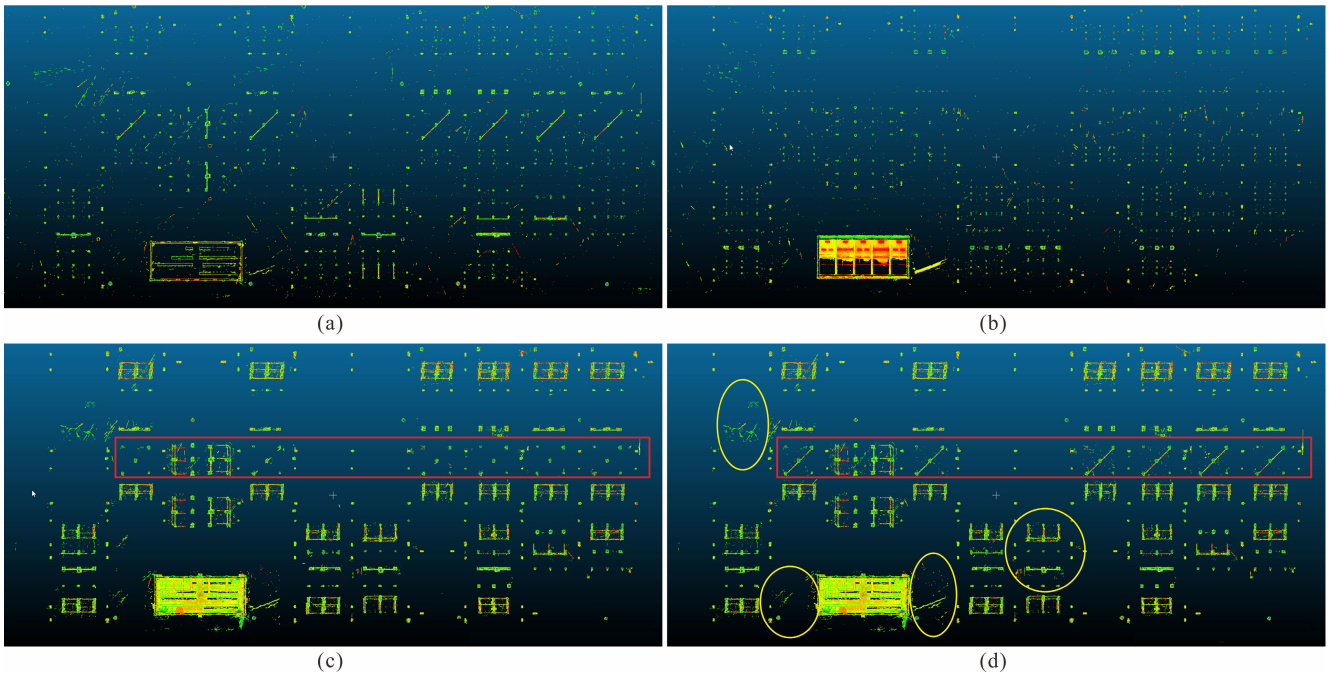

FIGURE 10. Extracting point clouds of each slice of equipment in the substation: (a) Horizontal slice H0 of point cloud data with an elevation range of 1-2 m, (b) Horizontal slice H2 of point cloud data with an elevation range of 3-4 m, (c) The H0 and H2 slices are clustered separately according to the voxel grid clustering algorithm, and the clustering results of H0 and H2 are reclustered according to the hierarchical clustering algorithm. The point clouds extracted after the hierarchical clustering is H02. (d) Horizontal slice H1 of point cloud data with an elevation range of 2-3 m. The H1 slice is clustered according to the voxel grid clustering algorithm, and the clustering results of H1 and H02 are reclustered according to the hierarchical clustering algorithm. The point cloud extracted after hierarchical clustering is H012. The red boxes show the advantages of adding H1 slice data to the cluster, and the yellow boxes show the areas with more noise.

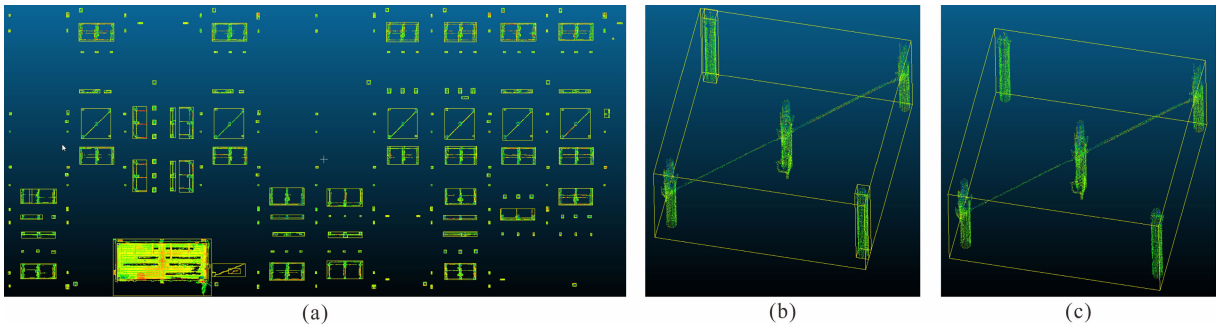

FIGURE 11. DBSCAN clustering results: (a) Point cloud data clustered by the DBSCAN algorithm; (b) clustering results by the refine recluster for abnormally distributed devices, which contain 1 three-phase device and 2 single-phase devices; (c) clustering result by the fast clustering with hierarchical strategy for abnormally distributed devices, which shows the undersegmentation of three devices into a cluster.

that using the vertical hierarchical clustering constrains the point clouds belonging to different height intervals, which effectively uses the vertical discontinuity of the noise point cloud and successfully removes most of the noise in the point cloud of the substation scene. However, because the H0 and H2 data do not include the point cloud of the connecting rod of the three-phase equipment (as shown in the red frames in Figure 10), the point cloud in H02 represents the three-phase equipment as three single-phase equipment.

With the addition of the H1 data, the result (H012) includes the connecting rod point cloud, and the point cloud that needs to be clustered becomes complete. However, the small amount of noise point clouds in H02 did not decrease with the addition of H1. The main noise areas are marked by the yellow frames in Figure 10d.

C. FINE SEGMENTATION

We further performed DBSCAN clustering after hierarchical clustering. The results showed that the DBSCAN clustering algorithm is able to remove most of the residual noise (Figure 11). The remaining main noise in H012 (indicated by the yellow frames in Figure 10d) is effectively removed in the DBSCAN clustering results. The better the denoising effect is, the smaller the clustering interference of the noise point cloud is to the equipment point cloud, and the more accurate the clustering result is.

The DBSCAN results show that it not only effectively removes noise but also accurately extracts abnormally distributed equipment (Figure 11b, c). The range of clusters is intelligently determined by our hierarchical clustering method based on box intersections is rectangular. When the main axis of the three-phase equipment does not strictly lie along the X or Y directions, the distribution of this device is considered abnormal. The bounding boxes of the clustering results will enclose the point clouds of other equipment (Figure 11b), resulting in undersegmentation. DBSCAN solves this undersegmentation problem and extracts the single-phase equipment point cloud mistakenly enclosed in the three-phase equipment point cloud cluster.

VI. DISCUSSION

A. NOISE REMOVAL

The most important feature of our method is to cluster the point clouds of equipment in the substation point cloud scene by using the rapid clustering characteristics of distance information. However, the effect of the method based on distance information is seriously affected by noise. Therefore, it is necessary to evaluate the effect of removing noise point clouds from the substation scene point cloud. The statistical outlier removal (SOR) is a commonly used point cloud denoising method, and we adopt this method to evaluate the denoising effect.

The results of SOR are affected by the k and n values. When n changes, the effects of processing the data directly with SOR filtering are shown in Figure 12, where it can be found that changes to the n value have little effect on the denoising results, but when k is too large, the connecting rods will be noncontinuous. We want the denoising effect to be affected by k as little as possible and retrain more effective point clouds.

After using the hierarchical clustering method to process the point cloud of the entire substation, the equipment point clouds can be extracted with a lower noise level. We use SOR to further filter these point cloud data to observe whether changes occur in the equipment point cloud. The SOR result is affected by the values of k and n. The effects of using SOR filtering to process the data directly are shown in Figure 12 as k and n change. The results show that changes to these parameter values in the SOR have little effect on the final device point clouds produced by our method, which indicates that our method has strong denoising ability.

The k value is a parameter that determines the size of the estimated neighborhood, and it has a greater impact on the point clouds of connected parts of the equipment. We selected a three-phase equipment for testing purposes that connected part of the point cloud and whose clustering result was discontinuous due to the presence of occlusions. The test results are shown in Figure 13. The k value is less sensitive to the processing results of the proposed method (Figure 13a, b), but when processing point cloud data with traditional methods,

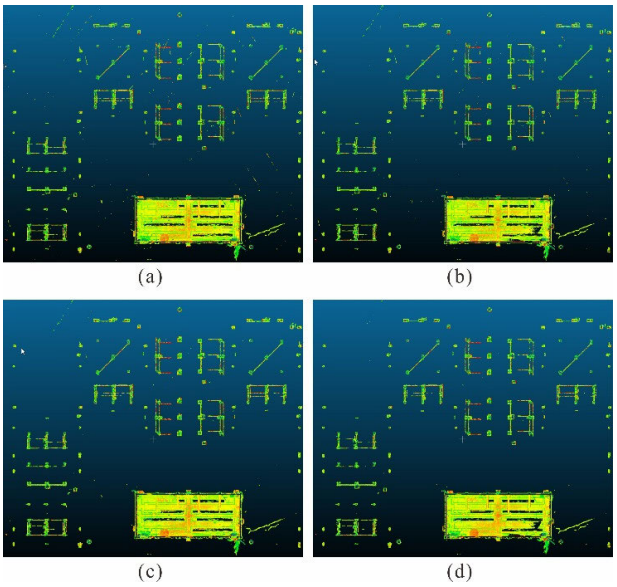

FIGURE 12. SOR clustering results: (a) $k = 10, n = 1$; (b) $k = 10, n = 0.5$; (c) $k = 50, n = 1$; (d) $k = 50, n = 0.5$.

even small variations may cause the connecting rod in the middle of the device to appear disconnected (Figure 13c). This disconnection can cause a three-phase device to be mistakenly clustered into two devices. This test verifies that the data we extracted have a low noise level.

B. REMOVING OTHER INTERERENCES

Substation scenes include various power switch boxes on the ground, and the point clouds may also contain data of construction workers. The vertical hierarchical clustering algorithm with different elevations can prevent the final clustering results from including such nonequipment information.

The heights of construction workers and power switch boxes are generally no greater than approximately 2 m. Slices are selected in the 1–2 m elevation interval for horizontal hierarchical clustering, and a set of clustering results will includes these interferences. In the proposed algorithm, we also select slices with a height of 3-4 m for horizontal hierarchical clustering. Then, we use the vertical hierarchical clustering method to integrate the horizontal clustering results of these two different height intervals. The integration process will directly identify point cloud data representing construction workers and power switch boxes as noise (Figure 14).

C. SEGMENTATION ACCURACY AND EFFICIENCY

In the proposed method, the results of hierarchical clustering are then reclustered by DBSCAN. We compare the results of the proposed method for processing the point cloud of the substation scene with the result of using DBSCAN directly.

The results after using DBSCAN to directly process substation site point clouds show that the clustering results are shown in Figure 15. In the processing result of the original point cloud (Figure 15a), the clustering result

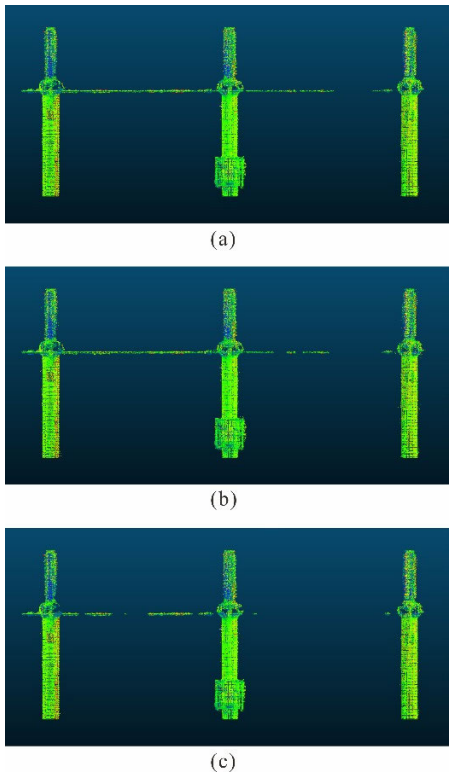

FIGURE 13. Extracted point clouds of discontinuous connectors by different methods: (a) Our method $k = 10$; (b) our method $k = 50$; (c) traditional method: the result of SOR ($k = 30$) filtering on the original point cloud data.

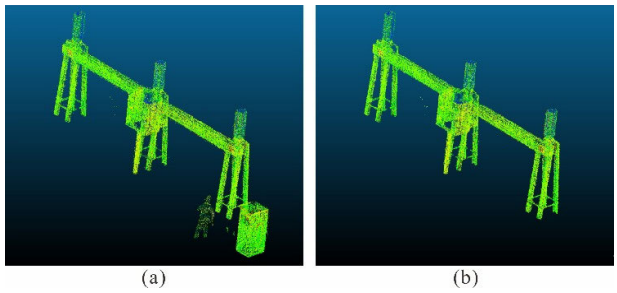

FIGURE 14. A comparison between the extraction results of our method and those of the conventional method: (a) point clouds after filtering the original point clouds with SOR; (b) point cloud data extracted by our algorithm.

is chaotic and mixed with a large number of noise point clouds that have not been effectively eliminated by DBSCAN.

Using SOR ($k = 20, n = 1.0$) to process the original point cloud, which can effectively remove most of the noise point cloud in the original point cloud. The results of clustering the denoised point clouds with DBSCAN show that DBSCAN's ability to suppress residual noise is still limited (Figure 15b, c and d). The residual noise will cause a large number of oversegmentation and undersegmentation in the clustering results of DBSCAN.

The number of point cloud clusters segmented by the two methods is shown in Table 2. Different DBSCAN parameters

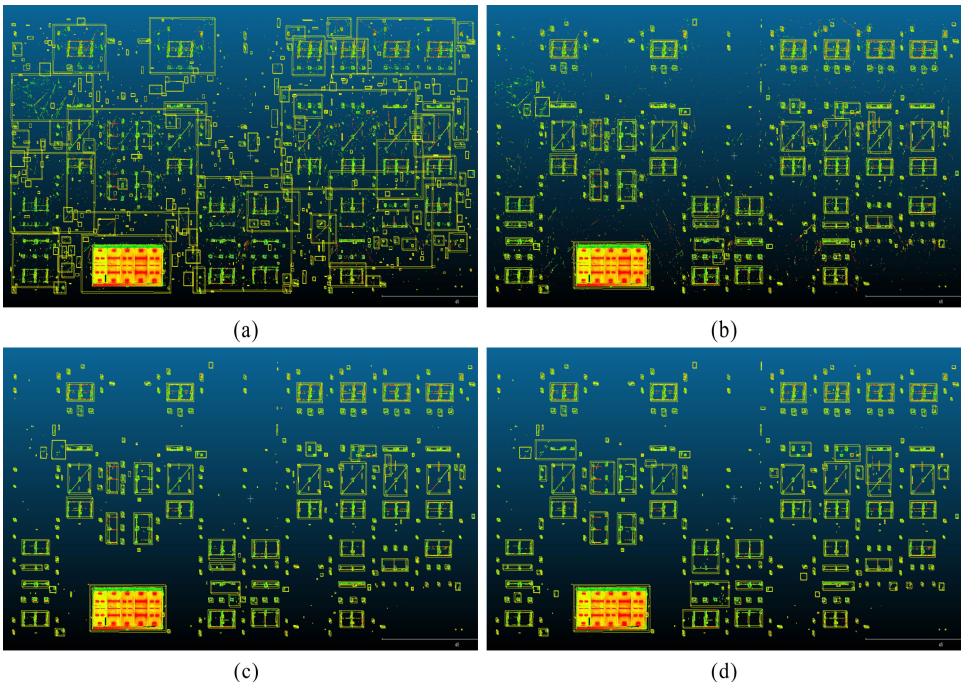

FIGURE 15. Clustering results of point cloud processing using DBSCAN directly. (a) Clustering result of processing the original point cloud directly using DBSCAN, (b-d) clustering results of processing the denoised point cloud directly using DBSCAN. The denoising of point cloud is performed by SOR ($k = 20$, $n = 1.0$). (b) $\varepsilon = 0.7$ and $min\_pts = 50$, (b) $\varepsilon = 1.0$ and $min\_pts = 50$, (c) $\varepsilon = 1.3$ and min_pts = 50.

TABLE 2. Statistical table of clustering results of different methods.

Method	ε	*min_pts*	Time (s)	Number of clusters	Accuracy
DBSCAN	0.7	10	286.27	459	39.15%
		50	333.57		
		100	340.27		
	1.0	10	891.79	394	55.32%
		50	1123.09		
		100	1086.09		
	1.3	10	1714.35	330	71.49%
		50	1700.50		
		100	1756.46		
Our method	0.7	10	43.16+31.01	241	97.45%
		50	43.16+32.40	249	94.67%
		100	43.16+32.15	252	93.62%
	1.0	10	43.16+43.00	234	98.72%
		50	43.16+43.60	231	97.87%
		100	43.16+44.40	237	99.15%
	1.3	10	43.16+57.75	223	96.60%
		50	43.16+58.88	223	96.60%
		100	43.16+58.82	223	96.60%

(ε and *min_pts*) have been considered. The ε value fluctuates based on the 0.7 m we set. Considering that the point cloud density is strongly affected by the sparse ratio parameters, we set a large range of 10–100 for *min_pts*. The number of clusters in Table 2 shows that the *min_pts* parameter values have little effect on the clustering results.

A comparison of the number of acquired point cloud clusters shows that the segmentation accuracy of using DBSCAN directly is extremely unstable. The segmentation accuracy falls below 40% when ε is 0.7 m because most equipment point clouds of equipment are then judged as noise by DBSCAN. The results of our method are shown in Figure 16. When different DBSCAN parameters are applied in our method, the clustering results remain basically the same. This comparison shows that our method has better parameter stability.

In Figure 16a, c and f, the point clouds of the three-phase equipment are oversegmented into two or three closely connected clusters, which is an important reason underlying the different numbers of point cloud clusters in Table 2. This oversegmentation occurs when the point clouds of three-phase equipment with connecting rods are sparser than those of other equipment. Fortunately, these oversegmented clusters are adjacent and can be merged directly by using the neighbor relationship. The total number of merged point cloud clusters is shown in Table 2. The final accuracy rate shows that different DBSCAN parameter values have little effect on the segmentation accuracy.

In Table 2, the calculation time of our method is divided into two parts: the running time of fast hierarchical clustering and the running time of DBSCAN in fine reclustering. The time complexity of the terrain correction step is $O(n)$, which is suitable for processing a large number of three-dimensional point clouds of the entire substation scene. In the fast hierarchical clustering step, the voxelized clustering belongs to an agglomerative cluster with a time complexity of

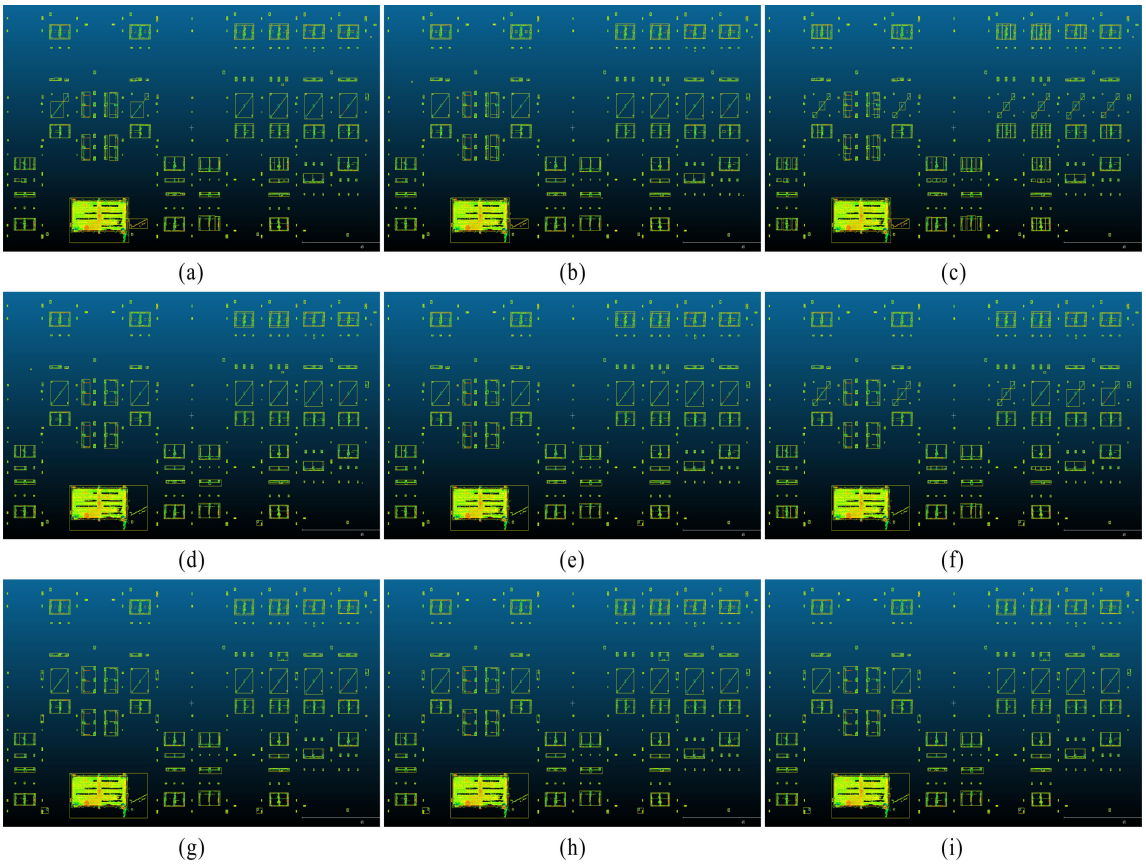

FIGURE 16. Comparison between the clustering results of different parameters of DBSCAN in the refine clustering. (a) $\varepsilon = 0.7$ and $min\_pts = 10$, (b) $\varepsilon = 0.7$ and $min\_pts = 50$, (c) $\varepsilon = 0.7$ and min_pts = 100, (d) $\varepsilon = 1.0$ and $min\_pts = 10$, (e) $\varepsilon = 1.0$ and $min\_pts = 50$, (f) $\varepsilon = 1.0$ and min_pts = 100, (g) $\varepsilon = 1.3$ and $min\_pts = 10$, (h) $\varepsilon = 1.3$ and $min\_pts = 50$, (i) $\varepsilon = 1.3$ and min_pts = 100.

$O(n^2 \log n)$, which has a relatively high complexity. However, the voxelization reduces the amount of data, which makes the calculation time of the fast hierarchical clustering process not significantly increase. The time complexity of horizontal hierarchical clustering, vertical hierarchical clustering and DBSCAN are all $O(n \log n)$, which is consistent with the common clustering algorithm using accelerated index structure with $O(\log n)$. The subsequent neighbor agglomerative process is based on the existing clustering results, and the size of the cluster set participating in the clustering is small. Therefore, even if the time complexity of the final aggregation process is $O(n^2 \log n)$, the actual execution speed is still relatively fast.

VII. CONCLUSION

In this paper, we proposed a data processing framework to quickly extract power equipment point clouds from point cloud substation scenes. The framework contains three main steps: terrain correction, hierarchical clustering and fine reclustering. We applied the proposed method to process a measured point cloud for a substation. The results show that terrain correction can remove nearly 40% of the redundant point cloud data. The horizontal hierarchical clustering based on bounding box cross-validation avoids the need to calculate the mutual distances between each point in each cluster,

TABLE 3. Terrain correction algorithm.

Input: point clouds $P_i=\{x_i, y_i, z_i, c_i\}$ (i=1,2,3,…,N)
Output: point clouds $P'_j=\{x_j, y_j, z_j, c_j\}$ (j=1,2,3,…,N')
1. Voxelization
2. Subsampling
3. Analyze the ground elevation. The number of points in different height ranges are counted.
4. Address outliers
4.1 Use 5 times the average value as the threshold value to find points with abnormal values
4.2 Connect adjacent abnormal points by region growing
4.3 Find the points around the abnormal points, calculate their average value, and assign the average value to the grid corresponding to the abnormal points
5. Remove all point clouds below the ground elevation
6. **Return**

while vertical hierarchical clustering integrates the reduced-dimensional clustering results on the horizontal plane. This hierarchical clustering strategy not only improves the clustering efficiency but also effectively removes most of the noise point clouds from the substation scene. Based on the hierarchical clustering results, a fine reclustering using DBSCAN is able to successfully extract point clouds of equipment in a substation. Compared with using DBSCAN directly, our method both reduces the calculation time (to approximately

TABLE 4. Size comparison of point clouds after removing or not removing ground data and not removing ground data.

	Subsampled data	Nonsubsampled data
Not removed	15.7 Gb size	146 Gb size
Removed	7.32 Gb size	55.5 Gb size
Ratio data size	46.3%	38.1%

5-30% of the original time), and improves the segmentation accuracy to approximately 95%. Moreover, our method is less affected by preset parameter values and is extremely robust to noise.

APPENDIX

The specific terrain correction algorithm is described in Table 3. The original point cloud $\mathbf{P}$ contains a point cloud representing the ground, and the number of points in point cloud $\mathbf{P}$ is N. After removing the point cloud representing the ground from $\mathbf{P}$ using the terrain correction algorithm, the remaining point cloud is $\mathbf{P}'$, and the number of points in $\mathbf{P}'$ is N'.

The change in the point cloud dataset size after removing ground point cloud data is shown in Appendix Table 4.

REFERENCES

[1] M. Arastounia and D. D. Lichti, "Automatic extraction of insulators from 3D LiDAR data of an electrical substation," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 2, pp. 19–24, Oct. 2013.

[2] M. Arastounia and D. Lichti, "Automatic object extraction from electrical substation point clouds," *Remote Sens.*, vol. 7, no. 11, pp. 15605–15629, Nov. 2015, doi: 10.3390/rs71115605.

[3] Q. Wu, H. Yang, M. Wei, O. Remil, B. Wang, and J. Wang, "Automatic 3D reconstruction of electrical substation scene from LiDAR point cloud," *Isprs J. Photogramm. Remote Sens.*, vol. 143, pp. 57–71, Sep. 2018.

[4] C. Munteanu, M. Purcar, D. Bursasiu, E. Merdan, and V. Farcas, "CAD/CAE modeling of the human exposure to electric field inside a high voltage substation," in *Proc. Int. Conf. Expo. Elect. Power Eng.*, Oct. 2014, pp. 476–479.

[5] Y. Zhang, X. Yuan, Y. Fang, and S. Chen, "UAV low altitude photogrammetry for power line inspection," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 1, Jan. 2017, Art. no. 1, doi: 10.3390/ijgi6010014.

[6] M. Arastounia and D. D. Lichti, "Segmentation of planar surfaces in LiDAR point clouds of an electrical substation by exploring the structure of points neighbourhood," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 40, no. 5, pp. 55–62, 2014.

[7] G. Sohn, Y. Jwa, and H. B. Kim, "Automatic powerline scene classification and reconstruction using airborne lidar data," *Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 13, Aug. 2012, Art. no. 167172.

[8] D. Popovic, V. Pajic, D. Jovanovic, F. Sabo, and J. Radovic, "Semi-automatic classification of power lines by using airborne lidar," *FIG Work. Week*, vol. 12, pp. 1–12, May 2017.

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

[10] Q. Li, Z. Chen, and Q. Hu, "A model-driven approach for 3D modeling of pylon from airborne lidar data," *Remote Sens.*, vol. 7, no. 9, Sep. 2015, Art. no. 9, doi: 10.3390/rs70911501.

[11] M. Ester, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowledg Discovery Data Mining*, 1996, pp. 226–231.

[12] Peng, Wang, Hongyan, Li, Hongfang, and Zhou, "Research on adaptive parameters determination in DBSCAN algorithm," *J. Inf. Comput. Sci.*, vol. 28, no. 3, pp. 289–292, 2012.

[13] P. Dorninger and C. Nothegger, "3D segmentation of unstructured point clouds for building modelling," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 35, pp. 191–196, Jul. 2007.

[14] J. Chen and B. Chen, "Architectural modeling from sparsely scanned range data," *Int. J. Comput. Vis.*, vol. 78, nos. 2–3, pp. 223–236, Jul. 2008, doi: 10.1007/s11263-007-0105-5.

[15] X. Zhang, G. Li, Y. Xiong, and F. He, "3D mesh segmentation using mean-shifted curvature," in *Proc. Int. Conf. Geometric Modeling Process.*, 2008, pp. 465–474.

[16] E. Grilli, F. Menna, and F. Remondino, "A review of point clouds segmentation and classification algorithms," *ISPRS Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 42, pp. 339–344, Feb. 2017.

[17] A. Nguyen and B. Le, "3D point cloud segmentation: A survey," in *Proc. IEEE Conf. Robot., Automat. Mechatronics*, Nov. 2013, pp. 225–230.

[18] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 36, no. 5, pp. 248–253, 2006.

[19] E. Castillo, J. Liang, and H. Zhao, *Point Cloud Segmentation and Denoising Via Constrained Nonlinear Least Squares Normal Estimates*. Berlin, Germany: Springer, 2013.

[20] Y. Liu and Y. Xiong, "Automatic segmentation of unorganized noisy point clouds based on the Gaussian map," *Comput.-Aided Des.*, vol. 40, no. 5, pp. 576–594, 2008.

[21] G. Vosselman, B. G. H. Gorte, G. Sithole, and T. Rabbani, "Recognising structure in laser scanner point clouds," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 46, pp. 33–38, Oct. 2004.

[22] A. Jagannathan and E. L. Miller, "Three-dimensional surface mesh segmentation using curvedness-based region growing approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2195–2204, Dec. 2007.

[23] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3D point clouds: A paradigm on lidar data for autonomous vehicle applications," in *Proc. Int. Conf. Robot. Automat.*, May 2017, pp. 5067–5073.

[24] A.-V. Vo, L. Truong-Hong, D. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 104, pp. 88–100, Jun. 2015, doi: 10.1016/j.isprsjprs.2015.01.011.

[25] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[27] Y. He, H. Tan, W. Luo, H. Mao, D. Ma, S. Feng, and J. Fan, "MR-DBSCAN: An efficient parallel density-based clustering algorithm using mapreduce," in *Proc. Int. Conf. Parallel Distrib. Syst.*, 2012, pp. 473–480.

[28] 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, p. 172, 2019.

[29] R. Pirrone, V. Cannella, S. Monteleone, and G. Giordano, "Linear density-based clustering with a discrete density model," Jul. 2018, *arXiv:1807.08158*. Accessed: Nov. 10, 2020. [Online]. Available: http://arxiv.org/abs/1807.08158

[30] R. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *Proc. 20th Int. Conf. Very Large Data Bases*. Santiago, Chile: Morgan Kaufmann, 1994.

[31] Y. Fan, Y. Huang, and J. Peng, "Point cloud compression based on hierarchical point clustering," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, Oct. 2013, pp. 1–7.

[32] X. Lu, J. Yao, J. Tu, K. Li, L. Li, and Y. Liu, "Pairwise linkage for point cloud segmentation," *Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3, no. 3, pp. 1–8, 2016.

[33] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Jun. 2014, pp. 6218–6225.

[34] D. Girardeau-Montaut. *Cloudcompare-Open Source Project; 2011*. [Online]. Available: http://www.danielgm.net/cc

[35] *IEEE Guide for the Design, Construction, and Operation of Electric Power Substations for Community Acceptance and Environmental Compatibility*, IEEE Standard 1127-2013 (Revision IEEE Std 1127-1998), Jan. 2014, pp. 1–50, doi: 10.1109/IEEESTD.2014.6720103.

[36] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4857–4868, Nov. 2020, doi: 10.1109/TNNLS.2019.2958324.

[37] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "COMIC: Multi-view clustering without parameter selection," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, Jun. 2019, pp. 5092–5101. [Online]. Available: http://proceedings.mlr.press/v97/peng19a.html
[38] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018, doi: 10.1109/TIP.2018.2848470.

JIANLONG GUO received the B.S. degree in electrical engineering and automation from the Taiyuan University of Technology, Taiyuan, China, and the M.S. degree in motors and appliances and the Ph.D. degree in power system and automation from the South China University of Technology, Guangzhou, China. He currently works as a Senior Economist at the Guangdong Power Grid Training and Evaluation Center. His research interest is in the area of power system and automation.

WEIXIA FENG received the B.S. degree in digital media technology and the M.S. degree in design from the Guangdong University of Technology, Guangzhou, China. He currently works as an Engineer with the Guangdong Power Grid Training and Evaluation Center. His research interest is in the area of power simulation training.

JIANG XUE received the B.S. degree in computer science and technology, and the M.S. and Ph.D. degrees in computer application technology from the South China University of Technology, Guangzhou, China. He currently works as a Senior Engineer with the Guangdong Power Grid Training and Evaluation Center. His research interest is in the area of computer application technology.

SHAN XIONG received the B.S. degree in electronic information science and technology from the Hunan University of Science and Technology, Xiangtan, China, and the M.S. degree in electrical engineering from Hunan University, Changsha, China. He currently works as a Senior Engineer with the Guangdong Power Grid Training and Evaluation Center. His research interest is in the area of power system and automation.

TENGFEI HAO received the B.S. degree in digital media from the Communication University of China, Nanjing, China. He currently works as an Assistant Engineer with the Guangdong Power Grid Training and Evaluation Center. His research interest is in the area of power simulation training.

RUIHENG LI received the Ph.D. degree in geophysics from the China University of Geosciences, Wuhan, China, in 2018. His research interests are in the area of power signal processing, substation modeling, and visualization.

HUBEN MAO received the B.S. degree in animation from the Hubei Institute of Fine Arts, Wuhan, China, and the M.S. degree in design art from Wuhan University, Wuhan, China. His research interest is in the area of multimedia design.