IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 May 2024, accepted 16 June 2024, date of publication 24 June 2024, date of current version 3 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3417904

RESEARCH ARTICLE

Indoor Point Cloud Segmentation Based on Hierarchical Fusion of Structural Features

YUDONG ZHOU, FENGBAO YANG, AND MIN GAO
School of Information and Communication Engineering, North University of China, Taiyuan 030051, China

Corresponding author: Fengbao Yang (yfengb@163.com)

This work was supported in part by the Opening Project of International Joint Research Center of Robotics and Intelligence System of Sichuan Province under Grant JQZN2023-002.

ABSTRACT Aiming at the problem that the existing algorithms are difficult to segment efficiently in indoor scenes due to the high similarity of stacked and closely adjacent objects, this paper proposes an indoor point cloud segmentation algorithm based on hierarchical feature fusion. Firstly, according to the results of stratification, the principal component analysis method is adopted to reduce the dimension of each layer point clouds, and then the decision rules on the two-dimensional plane are established based on the structural characteristics of each layer objects. Combined with the analysis of hierarchical connectivity, fusing the features of each layer, so as to obtain the global positioning of the object. Last, for further division, the density segmentation method based on inter cluster constraints are taken into consideration. The experimental results show that the location and segmentation effect of stacked objects is fine, it is worth mentioning that this method has high accuracy, and the average mIoU can reach 0.811. In addition, another advantage of this method is that it takes less time to process point clouds, the average consumption time is about 6.59 seconds. These characteristics represent that this method can better complete the segmentation of indoor scenes.

INDEX TERMS Point clouds segmentation, indoor scenes, structural features, principal component analysis, hierarchical integration, inter cluster constraint.

I. INTRODUCTION

With the development of emerging technologies such as 3D scanning technology [1] and Lidar technology [2], 3D point cloud-based applications have penetrated into various fields. As an important research topic in point cloud processing, segmentation has made significant contributions to many applications such as indoor reconstruction [3], [4], object recognition [5], [6], and reconstruction of building frame information [7], [8].

However, when facing the complex indoor scene with tight distribution of objects, the similarity between objects is too high due to the stacking, tightness and occlusion of the placement, which greatly increases the difficulty of cloud segmentation of indoor field spots. Therefore, despite extensive research on indoor point cloud segmentation, it remains a challenging task to segment tightly stacked objects into different categories [9].

The point cloud segmentation algorithm for closely adjacent objects is an extension of the traditional point cloud segmentation, which finds applications in various fields such as autonomous driving technology, three-dimensional modeling and virtual reality, intelligent robot navigation, and medical image analysis. It excels at segmenting complex real-life scenarios encountered in these domains. For instance, in autonomous cars, multiple objects are often found in close proximity to the vehicle. The point cloud segmentation algorithm designed specifically for nearby object segmentation enables accurate identification of different entities on the road including vehicles, pedestrians, and road signs. This facilitates more precise decision-making during driving operations. Similarly, this algorithm can be applied to intelligent robots where accurate segmentation of closely adjacent objects is crucial for tasks like path planning, obstacle avoidance, and manipulation. In warehouse robotics systems

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

specifically, such segmentation algorithms aid robots in distinguishing between goods and obstacles effectively enabling efficient cargo handling and stacking operations. In conclusion, the research and application of this algorithm holds great significance in promoting advancements across related fields while enhancing efficiency and quality of life and work.

In the current research, a variety of segmentation ideas have been provided: region growth segmentation [10], [11], [12] model-based fitting [13], [14] and unsupervised clustering segmentation [15], [16]. The segmentation based on region growth mainly divides the points with similar characteristics near the seed points into a region. This method has high accuracy, stability and robustness [17]. So, on this basis, many scholars have improved this algorithm or combined it with other methods to make the segmentation effect more ideal. Wang et al. [12] projected the point cloud data onto the image plane, selected the initial seed points according to their geometric information, and determined the visibility of each point according to the defined growth criteria, which ultimately greatly improved the efficiency of segmentation of visible points. Wang et al. [18] through setting σth as the $95th$ percentile of curvature and strengthening the region growing criteria through connective and coplanar analysis. The non-connecting points and non-coplanar points are detected and ignored, which improves the segmentation efficiency, which improves the segmentation efficiency. Jin et al. [19] combined deep learning with the regional growth algorithm to detect stalks through Faster R-CNN, and used the regional growth algorithm to finely segment a single corn according to the detected stem and seed points, reducing the algorithm's dependence on seed points. However, this kind of algorithm still has the problems of difficult parameter selection and high computational complexity, which can not be adapted.

Compared with the region growth algorithm, the method based on model fitting is mainly used to segment specific objects by fitting geometric primitives [20], such as RANSAC [21], [22], Hough transformations [23], [24], etc. However, these methods are difficult to segment complex shapes or achieve complete automation, because details can not always be modeled by easily identifiable geometric shapes. Therefore, it is usually used in combination with other methods [22], [25] to produce accurate results. Li et al. [21] represented the 3D unorganized point cloud with a group of NDT units, and selected a planar nondestructive testing unit as the minimum sample in each iteration to ensure the correctness of sampling on the same planar surface and improve the performance of the algorithm. Yang et al. [26] obtained a series of parallel planes from angle clustering according to the improved RANSAC detection, extracted the adjacent overlapping planes in the space, and further improved the accuracy of point cloud segmentation while reducing the processing speed. Song et al. [27] and others transformed 3D points into Hough space through HT algorithm, and then trained CNN model to segment objects. To some extent, they overcame the problems of unstructured spatial distribution, disordered arrangement and sparse distribution of point clouds.

For the unsupervised clustering segmentation algorithm, the point clouds are segmented according to the spatial position, color or other characteristics between the points, such as k-means [28], DBSCAN [29], [30], etc. The clustering algorithm can classify the points with continuous and uniform density, and realize the accurate segmentation of the plane. Chen et al. [31] obtained the local maximum of CHM through the local maximum method to determine the location of the initial cluster center and the cluster K value of the K-means algorithm, which made the segmentation accuracy higher. But the process of this method is cumbersome, it is easy to cause data loss in the process of converting lidar data into two-dimensional data, and can not avoid the defect that k-means takes too long. Czerniawski et al. [32] proposed to combine the semantic information stored in the plane using DBSCAN in 6D space, and train the decision tree classifier through dimensionality reduction and unsupervised learning, which can achieve an accuracy of more than 90% in large-scale building plane segmentation. Chen et al. [33] proposed additional DBSCAN clustering conditions and adaptive threshold based on candidate sample selection and plane validity detection in three-dimensional space to obtain an effective fitting plane. Although DBSCAN has good segmentation performance, like the region growing algorithm, it is extremely difficult to determine the parameters of the algorithm, and different parameters directly lead to significantly different results.

In addition to the above methods, point cloud segmentation methods based on deep learning are also widely used. These methods use deep neural networks to directly learn and predict the original point cloud data end-to-end, without manually designing feature extractors, and can better capture the local features and global structural information of point cloud data. It has good performance when dealing with large-scale and complex point cloud data. However, the methods based on deep learning have the disadvantages of high demand for training data, large time complexity, and weak generalization ability of the model, which are difficult to be used for small sample point cloud segmentation.

To sum up, the current point cloud segmentation algorithm has made good progress. Researchers have improved the traditional algorithm through different methods, making the segmentation accuracy continuously improved. However, due to the disorder and high density of point clouds, the traditional segmentation algorithm is still very time-consuming, and it is easy to produce the results of under segmentation or over segmentation, especially in the face of closely distributed objects, the similarity of different objects caused by the contact and overlap between point clouds is high, and it is difficult to segment efficiently. Therefore, how to segment indoor point cloud quickly and accurately is still a difficult problem in 3D point cloud processing.

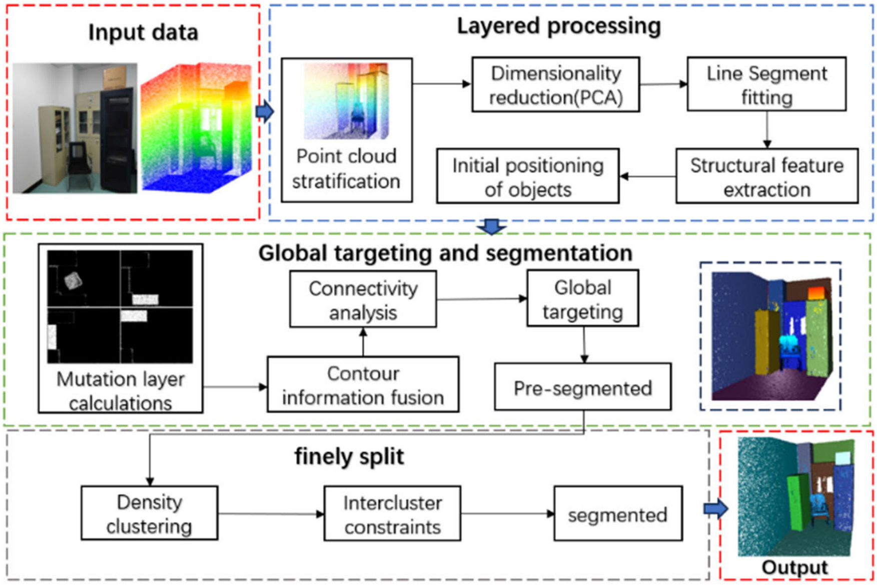

Input data
Layered processing
Point cloud stratification
Dimensionality reduction(PCA)
Line Segment fitting
Structural feature extraction
Initial positioning of objects
Global targeting and segmentation
Mutation layer calculations
Contour information fusion
Connectivity analysis
Global targeting
Pre-segmented
finely split
Density clustering
Intercluster constraints
segmented
Output

FIGURE 1. Overall flow chart.

In this paper, the point cloud is processed hierarchically to improve the computational efficiency, and the clustering method is used to make up the shortage of segmentation accuracy. The hierarchical processing of point cloud can effectively extract and represent the structure and characteristics of large-scale point cloud data, and realize the efficient storage, rapid retrieval and accurate analysis of point cloud data. Through layering, the processed object is composed of multi-layer small-scale point clouds, which makes it easier to extract the local features of the object, and avoids the result of low computational efficiency and difficulty in feature extraction caused by too dense data. After that, density clustering is used for secondary segmentation. Although the segmentation accuracy of this method is very considerable, it is inefficient in the face of large-scale point clouds, and the parameter selection is extremely difficult, so it needs to constantly adjust the parameters to achieve the expected results. Therefore, this paper proposes an algorithm combining the advantages of stratification and density clustering to achieve fast and accurate indoor point cloud segmentation. The overall flowchart of the proposed method is shown in Figure 1.

The rest of the paper is organized as follows: Section II explains in detail the rationale of the proposed method. Section III introduces the experimental design and experimental results. The discussion and conclusion of this paper are respectively in Section IV and Section V.

II. METHODOLOGY

This section will introduce the 3D point cloud segmentation algorithm from three stages: (1) point cloud hierarchical processing; (2) global object localization and segmentation; (3) Fine segmentation based on density clustering.

A. POINT CLOUD LAYERED PROCESSING

In this subsection, the original point cloud is stratified, and the point cloud of each layer is reduced to a two-dimensional plane by Principal Component Analysis method. After that, the dimension reduction points were fitted into line segments by setting thresholds for the length of the line segment and the Euclidean distance between the points on the line segment. Then, based on the normal vector Angle and intersection of the line segments in each layer, the closure analysis of the line segments in each layer is carried out, so as to obtain the contour features of different objects in each layer, and preliminarily realize the local location of objects in each layer. As shown in Figure 2, it is the flow chart of point cloud hierarchical processing.

Point cloud layering is a method to divide 3D point cloud data into different levels according to the height or depth standards. Through point cloud layering, the complexity of data can be greatly reduced, which is convenient for data processing and analysis. For the collected point cloud, it is not difficult to get the maximum value of the data on the Z axis. The number of layers to be divided is determined according to the size and height of the point cloud, and the height of each layer is obtained. The calculation formula of the center height of each layer is as follows:

$$\mathrm{h}_0 = \mathrm{z}_{\min} + \mathrm{d} \tag{1}$$

$$\mathrm{h}_\mathrm{n} = \mathrm{h}_0 + 2 * \mathrm{n} * \mathrm{d} \tag{2}$$

wherein, h_n represents the height of the nth layer. d is the distance from the center height of each layer to both ends. The value of d directly determines the number of layers to be divided. Figure 3 shows the layering results when the value of d is different. Each layer is given different colors and set

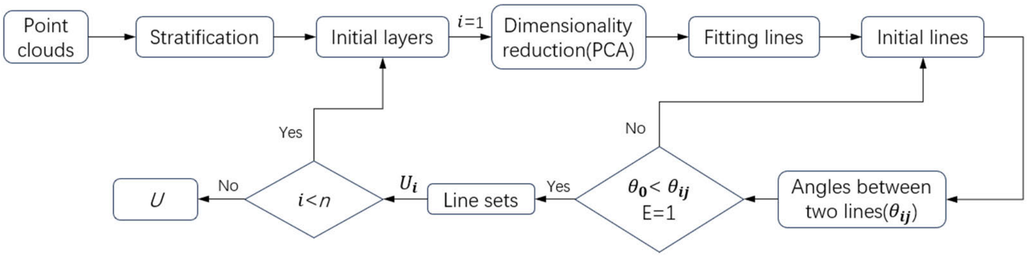
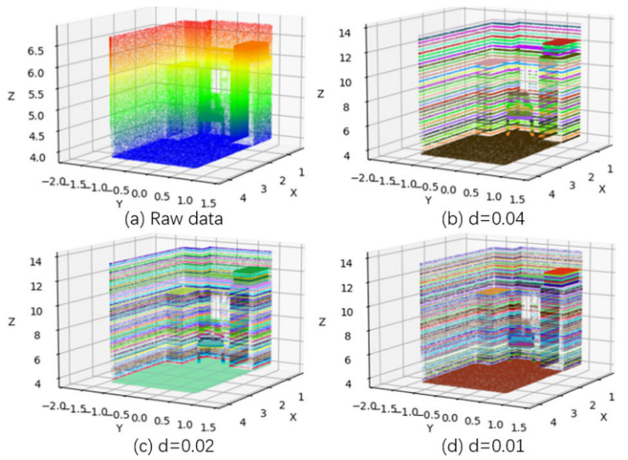


FIGURE 2. Hierarchical processing flowchart.



(a) Raw data

(b) d=0.04

(c) d=0.02

(d) d=0.01

FIGURE 3. Dimension reduction at each layer.

a certain spacing. Generally speaking, the lower the value of d, the more layers to be divided, making the structural characteristics of the object more obvious. Therefore, this algorithm selects d as 0.01 for layering.

In order to reduce the dimensionality of data, remove redundant information contained in the point cloud, and extract the most representative and distinguishing features in the point cloud, principal component analysis (PCA) [34] was used to reduce the dimensionality of each layer of point cloud.

The point cloud data is first converted into a matrix form, where each row represents a point and each column represents a dimension. Then zero averaging is performed on each column of the data.

$$X' = (X - \mu)/\sigma \tag{3}$$

After the data is zero-averaged, its covariance matrix C is calculated, which describes the relationship between the various variables in the data. At the same time, the eigenvalues and eigenvectors of the covariance matrix are obtained using the eigenvalue decomposition method. The eigenvalues are sorted from largest to smallest, the first two eigenvectors are selected, and the point cloud data is projected onto the low-dimensional space described by these eigenvectors to obtain two-dimensional point cloud data Y. The specific formula is as follows:

$$C = \frac{1}{n} * X' * X'^{T} \tag{4}$$

$$Y = PX \tag{5}$$

The dimensionality reduction plan of each layer is then obtained. As shown in Figure 4 below, the dimensionality reduction results of most layers are composed of multiple line segments. Therefore, in order to obtain the distribution characteristics of objects in each layer, it is necessary to analyze the existence and position coordinates of objects after locating the line segments of each layer.

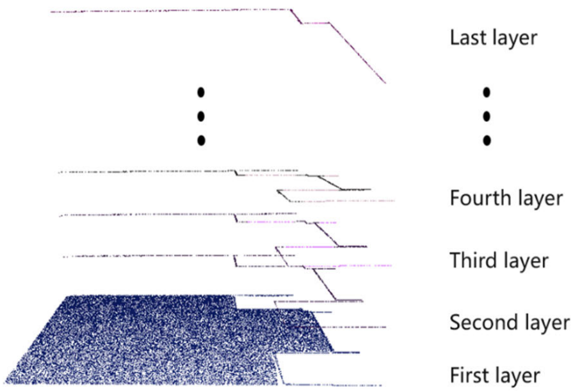


FIGURE 4. Laying with different width.

On the basis of the least square method [35], a threshold limit is set for the length of each line segment and the distance between the nearest neighbor points. However, this method may not be stable for the presence of noise and outliers in the data, so on the basis of the least square method, a threshold limit is set for the distance between the length of each line segment and the nearest neighbor point. When equations (6) and (7) are satisfied, that is, the length of the line segment is greater than the threshold λ and the Euclidean distance between the nearest neighbor points is less than the threshold ε, the line segment can be formed, where M is the set composed of all point indexes in this line segment. If the condition is not satisfied, the line segment is split and the second length determination is performed. Finally, the fitting and positioning of each line segment are realized, and the accuracy limitation caused by outliers and overfitting is

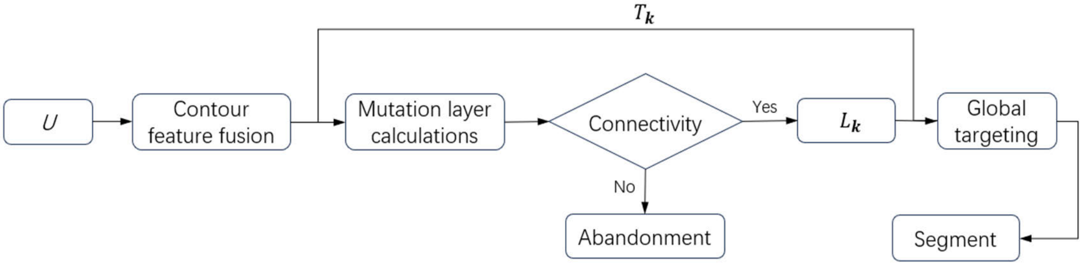


FIGURE 5. Flowchart of global localization and segmentation.

avoided when the least square method is used to fit multiple line segments.

$$\sqrt{(x_{max}-x_{min})^2+(y_{max}-y_{min})^2} \geq \lambda \tag{6}$$

$$\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2} \leq \varepsilon \tag{7}$$

After all the line segments of each layer are obtained, the closure analysis of each layer is carried out respectively, so as to obtain the preliminary judgment of the position of objects in each layer. The treatment of each layer is as follows:

(1) Randomly select a line segment and calculate the normal vector Angle and distance between other line segments.

(2) Determine whether the angle θ_{ij} between line segments is greater than the Angle threshold, such as equation (8), where is the direction vector of the line segments.

$$\theta_0 < \arccos\left(\frac{v_1 \cdot v_2}{|v_1| \cdot |v_2|}\right) \tag{8}$$

(3) The correlation between the line segment satisfying the condition and the original line segment is analyzed, that is, whether it approximately intersects. If relevant, the original line segment and the line segments satisfying the condition form a list and store it in the set U_i. The formed list consists of multiple line segments, which represent the combination of line segments that may be closed in the ith layer.

$$U_i = \{(a_0, a_1, a_2), (b_0, b_1), \ldots (k_0, k_1, k_2)\} \tag{9}$$

(4) Return to (1) and loop through until all line segments of this layer have been determined.

(5) The preliminary judgment result of the position of the object in each layer is obtained after removing the repeated combination of the layer set.

The combination of possible closed line segments of each layer is stored in the set U, as in equation (10), where n is the total number of layers. This result represents the edge contour information of each height of the object.

$$U = \{U_1, U_2 \ldots\ldots U_n\} \tag{10}$$

B. GLOBAL POSITIONING AND SEGMENTATION

In this section, the object contour features belonging to the same two-dimensional plane region in each layer are fused, and the global position information of the object is obtained by combining the analysis of the hierarchical connectivity, so as to realize the global localization, and the object-level segmentation is carried out according to this result. The algorithm flow is shown in Figure 5.

In order to accurately locate the global position of the object, it first needs to traverse the layers from bottom to top, and divide the U_i of each layer in the contour information set U, so that the sets representing the same position area in each layer are grouped into one object T_k. T_k represents the two-dimensional contour information of different objects in different layers in the scene, and it reflects the approximate position of each object to a certain extent. The set T composed of T_k is the set of contour information of all objects in the scene.

$$T = \{T_1, T_2, \ldots, T_k\} \tag{11}$$

Since the ray emitted by lidar cannot penetrate the solid object, generally only the point cloud data of the object surface can be obtained, that is, the top surface and side surface without occlusion. Therefore, in the vast majority of layers after stratification, the point cloud representing the object is composed of multiple line segments, as shown in Figure 6. However, in the process of traversing each layer from bottom to top, as the number of layers becomes higher, the top layer of each object will be continuously reached, and the number of point clouds in this area of the layer will change abruptly, which represents the top of the object to some extent, as shown in Figure 7.

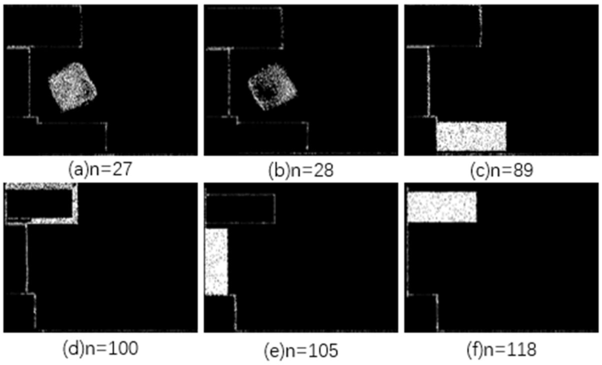


FIGURE 6. Normal layer plane.

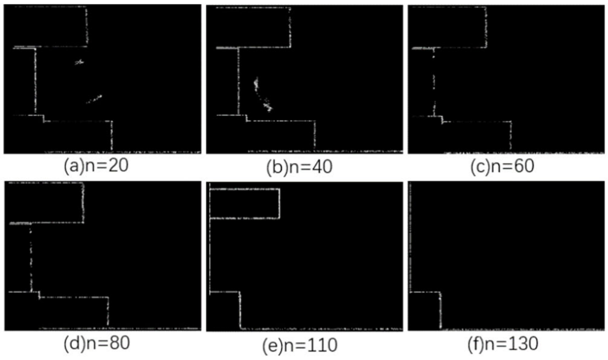


FIGURE 7. Abrupt change layer plane.

In order to avoid that the top point cloud is too thick due to the light transmission of the object, which makes the object difficult to accurately locate, it is necessary to analyze its connectivity, and determine the exact number of layers L_k on the top of each object through the line segment distribution information of the layers above and below the mutation layer. As shown in Figure 6 (a) and (b), the mutation layer of this object is 28 because there is no mutation in this region within the range of layers above the 28 layers, and the distribution characteristics of the lower layers are similar to that of this layer.

After obtaining the number of abrupt change layers of each object, the position coordinates of each object are determined by locating the point cloud dense area of the abrupt change layer, and combined with the contour information T_k of the corresponding position below the layer, the global positioning of the object is realized one by one. According to its position, the point cloud in the nearby area is classified in the three-dimensional space to complete the first segmentation of the densely packed indoor objects. The segmentation results are shown in Figure 9 (a), where the set of points defined as the same color is the class of objects or planes segmented.

C. PRECISE SEGMENTATION BASED ON DENSITY CLUSTERING

In the previous section, we accurately segmented indoor objects. However, due to the sparsity, inhomogeneity and noise of point clouds, the results of line segment fitting and top object positioning would be affected, making it difficult to segment some objects with sparse bottoms. Therefore, on this basis, the remaining unsegmented points in the point cloud need to be segmented twice, so as to improve the accuracy of the algorithm to a greater extent. Since unsegmented point clouds are always a small part, we adopt density-based segmentation method to achieve further segmentation for a small part of point clouds, which avoids excessive computational complexity when facing large-scale points and ensures segmentation accuracy.

The principle of density segmentation is to connect the points in the point cloud together to form clusters according to the density, and classify the points without sufficient density as noise points or outliers. However, this method is easy to lead to under-segmentation and over-segmentation in the segmentation of regions with different densities. Therefore, this paper adds the restriction of cluster spacing on this basis.

The specific steps are as follows: First choose any point, and then find all other points within the radius of this point as the center of the circle and eps, the number of which is marked as m, if $j = 0$, then the origin is marked as noise. If $j = 1$, it is labeled as the core sample and assigned a new cluster label, as in equation (12), where MPT is the minimum number of points within the range that can be attributed to the core sample. All neighbors of that point (within distance eps) are then accessed. If they haven't already been assigned a cluster, assign them the new cluster label you just created. If they are core samples, visit their neighbors in turn, and so on. As shown in Figure 8 (a), the red dot is the core sample.

$$j = \begin{cases} 0 & m < MPT \\ 1 & M \geq MPT \end{cases} \tag{12}$$

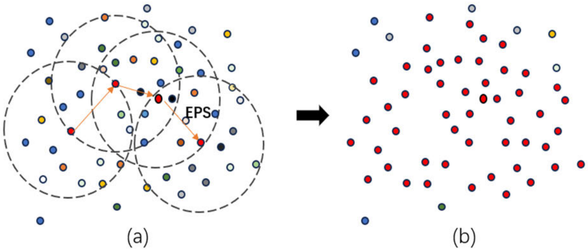


FIGURE 8. Density segmentation.

The cluster gradually grows until there are no more core samples in the eps range of the cluster. Pick another point that has not yet been visited and repeat the same process, stopping when all points are marked, as shown in Figure 8 (b).

After the segmentation result is obtained, the shortest distance between each cluster can be calculated, and the two clusters with too low distance can be classified to avoid excessive segmentation. At the same time, each cluster is assigned a different color according to the final result, which is the segmentation result, and the stability of the algorithm is further improved.

Figure 9 shows the specific segmentation process. (a) is the result of the first segmentation. Obviously, it can be seen that the chair and the box above the cabinet are not

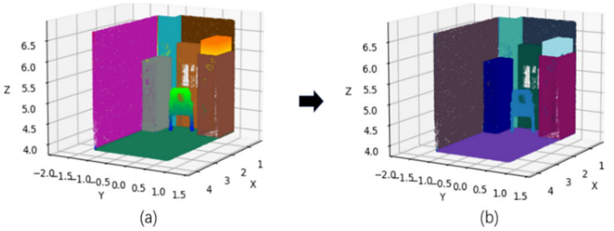


FIGURE 9. Segmentation results.

well segmented, and the density-based fine segmentation just solves this problem, as shown in Figure 9 (b), which further improves the accuracy of the algorithm.

III. EXPERIMENTS

In this section, we first briefly introduce the 3D LiDAR point cloud acquisition equipment. Secondly, the evaluation indexes of experimental results are introduced. Then, in order to illustrate the applicability of the proposed algorithm in various scenarios, this section conducts data acquisition and segmentation of indoor space in three different scenarios according to the above proposed methods. Finally, several common point cloud segmentation algorithms are selected for comparison to demonstrate the feasibility and advantages of the proposed method.

A. DATA COLLECTION

The hardware equipment used in the experiment is shown in Figure 10, which is a ground-borne LiDAR scanner, mainly composed of a laser sensor, a camera sensor and a fuselage, in which the fuselage contains a rotating head, inertial measurement unit (IMU) and other equipment. When collecting data, station scanning is required at multiple locations of the test site to ensure that all corners are laser fired to ensure adequate data.

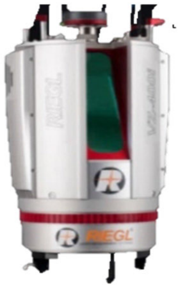


FIGURE 10. LiDAR scanner.

The principle of data acquisition in the experiment is as follows: First, start the device, the radar emits a laser beam through the built-in laser transmitter, the laser beam shines on the object in the surrounding environment, and the laser receiver receives the returned reflected laser. By measuring the time difference and angle information of the reflected laser light, the distance and spatial position between the object and the radar can be calculated. The radar then acquires its position and direction information via GPS or inertial navigation systems. Through multiple scans, a large amount of distance and location data is obtained, and these data are converted into point cloud data, that is, a collection of discrete points in three-dimensional space.

B. EXPERIMENTAL RESULTS

In this paper, the above equipment is used to collect data for three indoor scenes, and the point clouds data corresponding to these scenes were collected. The applicability of the proposed algorithm is demonstrated through the segmentation of indoor frame (scenario 1) and object accumulation (scenario 2 and 3).

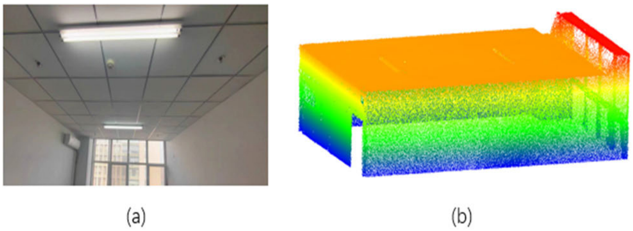


FIGURE 11. The real scene of scene 1; (b) the original point cloud.

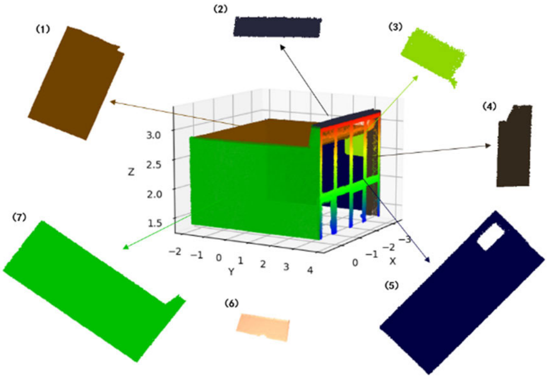


FIGURE 12. Final segmentation results for scene 1.

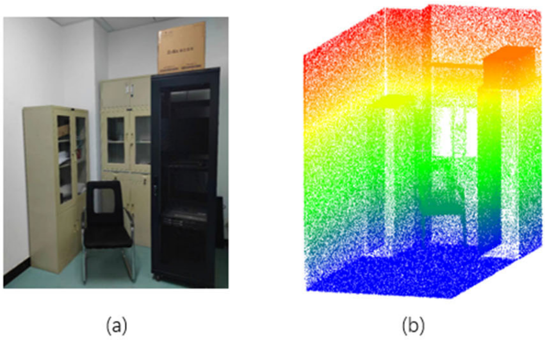


FIGURE 13. The real scene of scene 2; (b) the original point cloud.

Firstly, the first scene is the upper space of the office, as shown in Figure 11 (a), which mainly shows the segmentation effect of the proposed algorithm for the irregular indoor space structure. The corresponding point cloud data is shown in Figure 11 (b). The segmentation results are shown in Figure 12. The segmented planes of the same color are marked as (1)-(7) respectively, corresponding to different planes in scene 1, where (6) is the occluded surface at the wall, and the unsegmented area is represented by the origin cloud color. And the Table 1 reflects the relationship between the segmentation results of various types of objects and the true number of points in this scene.

As shown in Figure 13 (a), Scene 2 is an indoor space equipped with scattered objects, and the corresponding point

TABLE 1. **The *IoU* and indicators of various objects in Scenario 1.**

Metrics	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8	Class9	Class10
Real points	8890	66383	25928	15685	7152	64086	6350	2503	197939	2406
Intersection	8890	66233	24561	9492	6989	64084	5591	2502	194894	2405
Union	10833	68762	35076	17015	7431	65766	6526	203495	206540	203496
IoU	0.8206	0.9632	0.7002	0.5578	0.9405	0.9744	0.8567	0.0123	0.9436	0.0118

TABLE 2. **The *IoU* and indicators of various objects in Scenario 2.**

Metrics	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8	Class9
Real points	45446	12501	28113	42250	36329	23589	24715	15476	40153
Intersection	45259	11625	26682	42193	36267	23251	24687	13136	40153
Union	45446	12501	28150	43607	48208	23818	24779	15475	42097
IoU	0.9959	0.9299	0.9478	0.9676	0.7523	0.9762	0.9963	0.8488	0.9538

cloud data is shown in Figure 13 (b), which contains about 600,000 points, showing the segmentation effect of the proposed algorithm in the indoor space. The segmentation results are shown in Figure 14 below. Different colors represent different categories of objects divided, among which (1), (2), (7) and (9) are the spatial framework of the room, and (3), (4), (5), (6) and (8) are cabinets, boxes and chairs in the room. And the Table 2 reflects the relationship between the segmentation results of various types of objects and the true number of points in this scene.

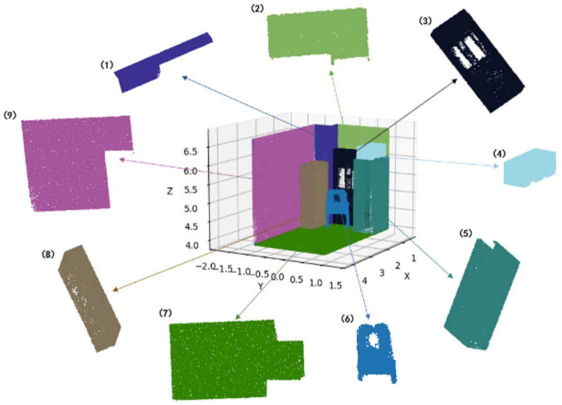


FIGURE 14. **Final segmentation results for scene 2.**

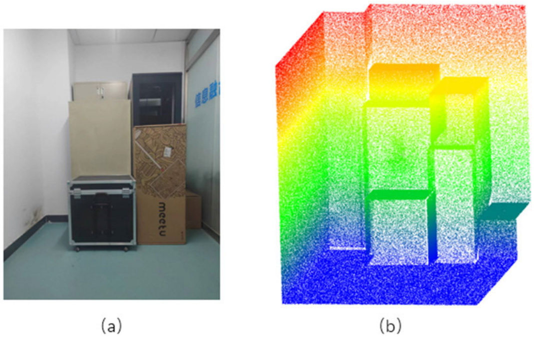


FIGURE 15. **The real scene of scene 3; (b) the original point cloud.**

Scene 3, like scene 2, is a storage room with objects of different sizes, as shown in Figure 15 (a). The number of objects in this scene increases, and the way of stacking is closer, which mainly shows the plane segmentation results of the proposed algorithm when the objects in the storage room are arranged more closely. Figure 15 (b) is the initial point cloud of scenario 3, while Figure 16 shows the final segmentation results and the individual segmented objects. The segmented planes of the same color are labeled (1-10), corresponding to different planes in scene 3. And the Table 3 reflects the relationship between the segmentation results of various types of objects and the true number of points in this scene.

C. COMPARISON WITH OTHER METHODS

To illustrate the advantages of the proposed algorithm, this section is segmented by other methods to demonstrate the advantages of the proposed algorithm in an intuitive form. In order to minimize the influence of processing devices on the final segmentation results, all algorithms are run on the same device, and we also adjust the parameters in several comparison methods to obtain relatively better segmentation results. The segmentation results of five comparison segmentation methods obtained in three scenarios are shown in

TABLE 3. The *IoU* and indicators of various objects in Scenario 3.

Metrics	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8	Class9	Class10
Real points	38976	88707	71943	39876	65697	33695	32097	38345	134158	26027
Intersection	38594	88525	71943	39843	64147	27598	32065	38199	118888	25313
Union	39050	88707	73818	42228	69618	74746	47113	38590	134316	52595
IoU	0.9883	0.9979	0.9746	0.9435	0.9214	0.3692	0.6806	0.9899	0.8851	0.4813

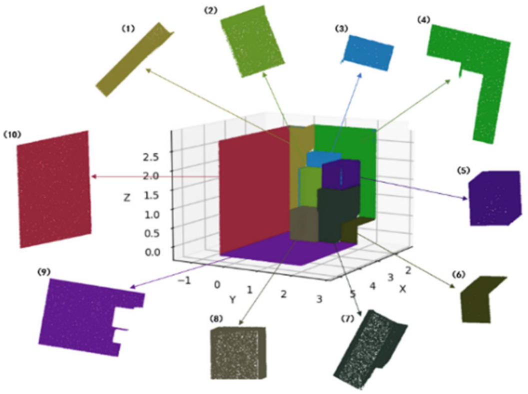


FIGURE 16. Final segmentation results for scene 3.

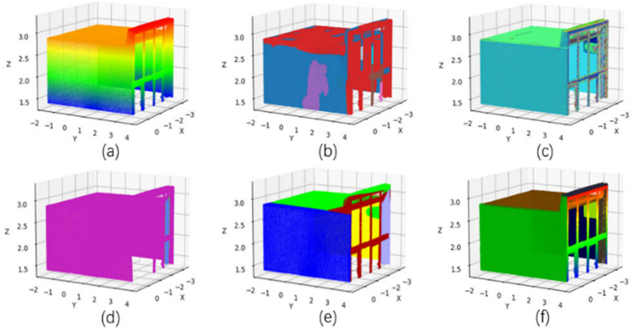


FIGURE 17. Comparison of scene 1.

Figure 17, Figure18 and Figure19. In each figure, (a) represents the original point cloud; (b), (c), (d), (e) and (f) stand for DBSCAN, RANSAC, RG, and PointNet respectively. The segmentation results of the algorithm are also reflected in this section. And the segmented point clouds are displayed in different colors.

D. EVALUATION INDEX OF EXPERIMENTAL RESULTS

This section introduces several parameters to evaluate the quality of each experimental result, namely *F1-score* and operation time T. *F1-score* is the harmonic average of accuracy (S_p) and recall rate (S_r), and its size reflects the accuracy and completeness of the classification model, that is, the accuracy of the model in identifying positive cases and the completeness of all positive cases. The number of samples is used here for calculation. The specific calculation formula is as follows:

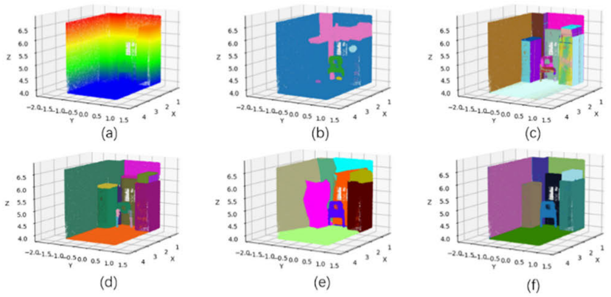


FIGURE 18. Comparison of scene 2.

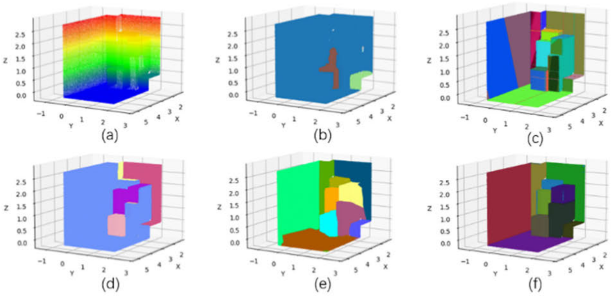


FIGURE 19. Comparison of scene 3.

$$F1 - score = 2 * (S_p * S_r)/(S_p + S_r) \tag{13}$$

Accuracy measures how much of the sample in which the model is predicted to be positive is actually positive, i.e. the proportion of the sample in which the model is predicted to be positive is correctly classified. The recall rate measures how many true positive cases the correctly identifies, i.e. the proportion of true positive cases that the model correctly identifies. The calculation formula of accuracy and recall rate is as follows:

$$S_p = TP/(TP + FP) \tag{14}$$

$$S_r = TP/(TP + FN) \tag{15}$$

TP indicates True Positive, that is, the number of samples that are correctly classified as positive. *FP* indicates False

TABLE 4. Evaluation parameter.

Scene	Method	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8	Class9	Class10
1	DBSCAN	0.0225	0.0546	0.0679	0.0411	0.0188	0.1670	0.1670	0.0666	0.5155	0.0063
	RANSAC	0.6311	**0.9791**	0.3487	0.5232	0.9162	**0.9767**	0.3976	**0.2557**	**0.9544**	**0.1831**
	RG	0.0240	0.1889	0.4145	0.6037	0.0021	0.1830	0.7521	0.0056	0.5562	0.0025
	PointNet	0.0448	0.8482	0.3673	**0.6782**	0.1238	0.8615	0.0112	0.0136	0.7363	0.0131
	Ours	**0.8206**	0.9632	**0.7002**	0.5578	**0.9405**	0.9744	**0.8567**	0.0123	0.9436	0.0118
2	DBSCAN	0.2978	0.0465	0.2680	0.0938	0.2409	0.1637	0.1056	0.8726	0.2655	—
	RANSAC	0.9506	0.7575	0.6653	0.8197	0.6748	0.6295	0.8467	0.3142	0.9420	—
	RG	0.7998	0.6198	**0.9580**	0.5283	0.6873	0.5371	0.8536	0.2623	0.9472	—
	PointNet	0.7336	0.5089	0.8553	0.8491	0.7450	0.5549	0.6154	**0.9381**	0.9019	—
	Ours	**0.9959**	**0.9299**	0.9478	**0.9676**	**0.7523**	**0.9762**	**0.9963**	0.8488	**0.9538**	—
3	DBSCAN	0.1057	0.3019	0.2435	0.2571	0.3524	0.1281	0.4596	0.9475	0.5027	0.0082
	RANSAC	0.5056	0.9974	0.4240	0.8609	0.5451	**0.8113**	0.6242	0.7151	0.1522	0.4909
	RG	0.5660	0.3013	0.2430	0.1280	0.3931	0.2197	0.6653	0.2600	0.6362	0.1065
	PointNet	0.7201	0.9143	0.8601	0.8522	0.8741	0.6427	**0.8272**	0.8774	0.7772	**0.5156**
	Ours	**0.9883**	**0.9979**	**0.9746**	**0.9435**	**0.9214**	0.3692	0.6806	**0.9899**	**0.8851**	0.4813

Positive, that is, the number of samples that are incorrectly classified as positive. FN indicates False Negative, that is, the number of samples that are incorrectly classified as negative.

Intersection over Union (IoU) is a common metric to evaluate the degree of overlap between the predicted results and the true labels in object detection or segmentation tasks. It is obtained by computing the intersection area between the predicted region and the true region divided by their union area. Mean Intersection over Union ($mIoU$) is the average of IoU, which is commonly used in the evaluation of multi-class segmentation tasks, and it is calculated by summing the IoU of each class and taking the average.

The evaluation indicators in the comparison experiment are shown in Table 4 below, which visually shows the parameters of S_p, S_r, *F1-score*,$mIoU$ and T of each method in the three scenes. And the IoU of various objects in different scenarios is reflected in Table 5.

IV. DISCUSSION

In terms of experimental results, for indoor spatial structure, the algorithm proposed in this paper can easily divide the wall into various planes in the final segmentation results. As can be seen from Figure 12 and Figure 17, although other algorithms can also extract part of the planes, the proposed planes are often incomplete or inaccurate. The phenomena of under-segmentation, unidirectional segmentation and over-segmentation are obvious. In contrast, the algorithm in this paper performs well in dividing the architectural structure of the interior space. However, it can be clearly seen from Figure 17 that all methods have unsatisfactory segmentation effect on the window part. This is because the glass on the window has a high transparency, most of the laser beam will be absorbed or transmitted when passing through the glass, and only a small amount of laser can be reflected back, which will cause the window glass points to be missing in the point cloud data. All kinds of algorithms will have obvious under-segmentation phenomenon when they segment the window frame with too low similarity, which is also the future direction of the algorithm.

When the scene is a storage room adjacent to an object, it can be seen from Figure 18 and 19 that the similarity between objects is high due to blurred or overlapping boundaries and excessive density when objects are densely placed, which makes it difficult for common segmentation algorithms such as DBSCAN, RANSAC and Region Growth Segmentation to correctly segment the object. Among them, DBSCAN is the most affected. Wrongly categorize the objects stacked together into one class, which greatly reduces the accuracy; Although RANSAC has a good effect, it is easy to fall into the local optimal solution, and it is obvious that the phenomenon of over-segmentation is serious. The region growth algorithm relies too much on neighborhood information, resulting in excessive segmentation, and the time complexity is too high. As can be seen from Table 5, the algorithm obviously takes much longer time than other methods. What is more, the PointNet algorithm has a prominent segmentation effect on each scene by using the trained model. However, due to the insufficient number of samples, the learning ability of the model on local features is weak, and finally the edge position of each object is difficult to segment accurately. Overall, the algorithm proposed in this paper performs preliminary screening of objects at each layer after point cloud layering and processing, and adjusts the results through hierarchical connectivity analysis, which finally solves the problem of difficult segmentation when objects are tightly packed. Compared with other algorithms, the proposed algorithm is more accurate in the segmentation of tightly packed goods, and is superior to the four algorithms in completeness and vision.

TABLE 5. The *IoU* of various objects in different scenarios.

Scene	Method	S_p(%)	S_r(%)	F1-score(%)	mIoU	T(s)
Scene1	DBSCAN	55.6	44.4	49.4	0.127	2.490
	RANSAC	75.0	66.7	70.6	0.617	8.185
	RG	50.0	10.0	16.7	0.276	175.280
	PointNet	80.0	40.0	53.3	0.370	54217.243
	Ours	100.0	70.0	82.3	0.678	3.167
Scene2	DBSCAN	66.7	44.4	53.3	0.265	3.729
	RANSAC	72.7	88.9	79.9	0.733	13.372
	RG	80.0	88.9	84.2	0.688	117.77
	PointNet	88.9	100.0	94.1	0.745	54217.243
	Ours	90.0	100.0	94.7	0.933	3.418
Scene3	DBSCAN	33.3	10.0	15.4	0.331	16.62
	RANSAC	57.1	80.0	66.6	0.613	22.74
	RG	40.0	20.0	26.7	0.352	142.3
	PointNet	80.0	100.0	88.9	0.734	54217.243
	Ours	90.9	100.0	95.2	0.823	13.19

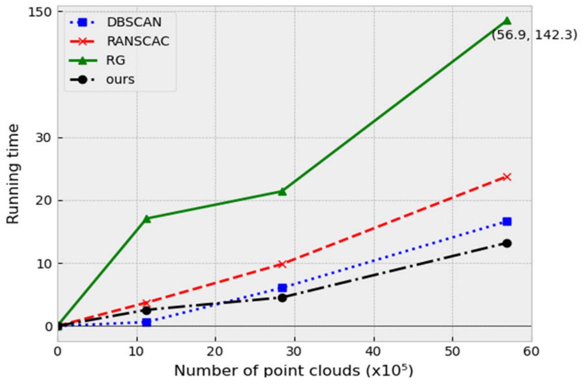


FIGURE 20. Efficiency of the four segmentation methods in Scene 3.

In terms of evaluation parameters, the efficiency and accuracy of our algorithm is better than that of the comparison algorithm. In Table 5, the segmentation time of scene 3 is 16.62s (DBSCAN), 22.74s (RANSAC), 142.3s (RG) and 13.19s (Ours) respectively, for PointNet, it takes about 15 hours to train the model. Obviously, our algorithm runs faster and performs well in terms of efficiency. In addition, the plane segmentation accuracy of the proposed algorithm is also slightly higher than that of the other four methods. We can intuitively see in Figures 17, 18 and 19 that the segmentation effect of this algorithm is more prominent, and from the relevant parameters in Figure 21 and Table 5, we can also observe that the *F1-score* of the four algorithms varies greatly in each scene, but the performance of the proposed algorithm is still stable in the complex environment scene 3, with the *F1-score* reaching 95.2% and the *mIoU* reaching 0.823, which is still better than the other four methods in general.

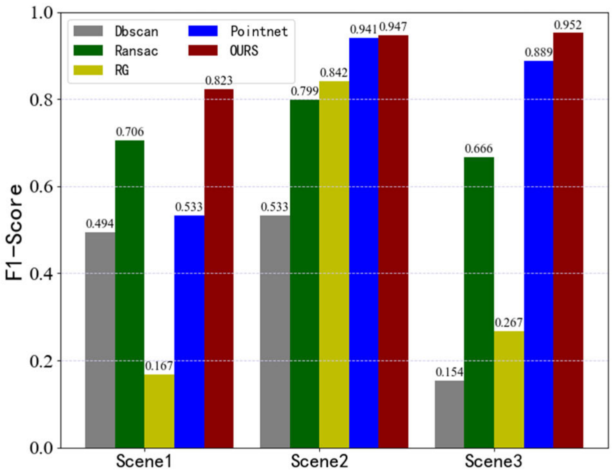


FIGURE 21. *F1-score* values of four segmentation methods in three scenes.

In general, the proposed algorithm has certain advantages over other algorithms, with an average *F1-score* of 90.73 and an average *mIoU* of 0.811, which is superior to other algorithms in terms of both visual and evaluation indicators. However it also has some shortcomings: First, when dealing with objects consisting of a small number of point clouds, such as window frames, the algorithm is difficult to segment effectively due to the inconspicuous features. What is more, the current segmentation of indoor close-packed objects is mainly for the scenarios of densely stacked goods such as storage rooms, which may make the algorithm unable to achieve the expected results in the very messy indoor scenarios where various items are randomly stacked.

V. CONCLUSION

In this paper, an indoor point cloud segmentation method based on hierarchical structure feature fusion is proposed, combining the advantages of hierarchical structure feature fusion and density clustering. This method is tested in the scene of irregular indoor frames and objects closely stacked, and good results are obtained. The main conclusions are as follows:

(1) From the direct comparison of experimental results, the proposed method accurately segments indoor point clouds, effectively solves the phenomenon of under-segmentation and over-segmentation caused by too high similarity, and the segmented area has a high consistency with the real indoor environment. In the quantitative analysis, the average *F1-score* value of the proposed method is 87.8%, the average *mIoU* and runtime can reach 0.811 and 6.59s, which is superior to the other three test methods, which verifies the superiority of the proposed method.

(2) Comprehensively considering the distribution information and hierarchical structure characteristics of point clouds, it realizes the global positioning and segmentation of indoor objects, especially in the point cloud segmentation when objects are closely adjacent and stacked. The precision and efficiency of segmentation are effectively improved.

In the future, we will continue to optimize the point cloud segmentation algorithm and try to solve the problem that the algorithm is difficult to efficiently segment under the scene of too high randomness of items. The iterative optimization method is considered to adapt the model parameters according to the segmentation results, and gradually improve the segmentation results, and combine with the current research to improve the performance of high-point cloud segmentation.

CONFLICTS OF INTERESTS

The authors declare no conflict of interest regarding the publication of this article.

ACKNOWLEDGMENT

The authors are grateful to the editor and reviewers for their professional comments, which significantly improved this work.

REFERENCES

[1] F. Di Stefano, S. Chiappini, A. Gorreja, M. Balestra, and R. Pierdicca, "Mobile 3D scan LiDAR: A literature review," *Geomatics, Natural Hazards Risk*, vol. 12, no. 1, pp. 2387–2429, Jan. 2021, doi: 10.1080/19475705.2021.1964617.

[2] R. Bogue, "The growing importance of LiDAR technology," *Ind. Robot: Int. J. Robot. Res. Appl.*, vol. 49, no. 6, pp. 1025–1031, Sep. 2022, doi: 10.1108/ir-05-2022-0138.

[3] B. Zhao, X. Hua, K. Yu, W. Xuan, X. Chen, and W. Tao, "Indoor point cloud segmentation using iterative Gaussian mapping and improved model fitting," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 7890–7907, Nov. 2020, doi: 10.1109/TGRS.2020.2984943.

[4] H. Macher, T. Landes, and P. Grussenmeyer, "From point clouds to building information models: 3D semi-automatic reconstruction of indoors of existing buildings," *Appl. Sci.*, vol. 7, no. 10, p. 1030, Oct. 2017, doi: 10.3390/app7101030.

[5] X. Chen, H. Wu, D. Lichti, X. Han, Y. Ban, P. Li, and H. Deng, "Extraction of indoor objects based on the exponential function density clustering model," *Inf. Sci.*, vol. 607, pp. 1111–1135, Aug. 2022, doi: 10.1016/j.ins.2022.06.032.

[6] Z. Xu, Y. Liang, Y. Xu, Z. Fang, and U. Stilla, "Geometric modeling and surface-quality inspection of prefabricated concrete components using sliced point clouds," *J. Construction Eng. Manage.*, vol. 148, no. 9, Sep. 2022, Art. no. 04022087, doi: 10.1061/(asce)co.1943-7862.0002345.

[7] A. Arnaud, M. Gouiffès, and M. Ammi, "On the fly plane detection and time consistency for indoor building wall recognition using a tablet equipped with a depth sensor," *IEEE Access*, vol. 6, pp. 17643–17652, 2018, doi: 10.1109/ACCESS.2018.2817838.

[8] T. Wang, Q. Wang, H. Ai, and L. Zhang, "Semantics-and-primitives-guided indoor 3D reconstruction from point clouds," *Remote Sens.*, vol. 14, no. 19, p. 4820, Sep. 2022, doi: 10.3390/rs14194820.

[9] J. Xiao, J. Zhang, B. Adler, H. Zhang, and J. Zhang, "Three-dimensional point cloud plane segmentation in both structured and unstructured environments," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1641–1652, Dec. 2013, doi: 10.1016/j.robot.2013.07.001.

[10] W. Zhang, F. Zhou, L. Wang, and P. Sun, "Region growing based on 2-D–3-D mutual projections for visible point cloud segmentation," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021, doi: 10.1109/TIM.2021.3080385.

[11] T. Rabbani, F. V. D. Heuvel, and G. Vosselman, "Segmentation of point clouds using smoothness constraint," in *Proc. ISPRS Commission V Symp. 'Image Eng. Vis. Metrol.*, 2006, pp. 1–6.

[12] X. Wang and J. Xiao, "Research of plane extraction methods based on region growing," in *Proc. Int. Conf. Virtual Reality Vis. (ICVRV)*, 2016, pp. 298–303.

[13] D. Maturana, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2015, pp. 922–928.

[14] W. Zhang, Z. Li, and J. Shan, "Optimal model fitting for building reconstruction from point clouds," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 9636–9650, 2021, doi: 10.1109/JSTARS.2021.3110429.

[15] E. Grilli and F. Menna, "A review of point clouds segmentation and classification algorithms," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W3, pp. 339–344, 2017.

[16] S. Zhang, S. Cui, and Z. Ding, "Hypergraph spectral clustering for point cloud segmentation," *IEEE Signal Process. Lett.*, vol. 27, pp. 1655–1659, 2020, doi: 10.1109/LSP.2020.3023587.

[17] P. Li, R. Wang, Y. Wang, and G. Gao, "Automated method of extracting urban roads based on region growing from mobile laser scanning data," *Sensors*, vol. 19, no. 23, p. 5262, Nov. 2019, doi: 10.3390/s19235262.

[18] W. Wang, Y. Zhang, G. Ge, Q. Jiang, Y. Wang, and L. Hu, "Indoor point cloud segmentation using a modified region growing algorithm and accurate normal estimation," *IEEE Access*, vol. 11, pp. 42510–42520, 2023, doi: 10.1109/ACCESS.2023.3270709.

[19] S. Jin, Y. Su, S. Gao, F. Wu, T. Hu, J. Liu, W. Li, D. Wang, S. Chen, Y. Jiang, S. Pang, and Q. Guo, "Deep learning: Individual maize segmentation from terrestrial LiDAR data using faster R-CNN and regional growth algorithms," *Frontiers Plant Sci.*, vol. 9, Jun. 2018, Art. no. 00866, doi: 10.3389/fpls.2018.00866.

[20] A. Raffo, C. Romanengo, B. Falcidieno, and S. Biasotti, "Fitting and recognition of geometric primitives in segmented 3D point clouds using a localized voting procedure," *Comput. Aided Geometric Design*, vol. 97, May 2022, Art. no. 102123, doi: 10.1016/j.cagd.2022.102123.

[21] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, and L. Tang, "An improved RANSAC for 3D point cloud plane segmentation based on normal distribution transformation cells," *Remote Sens.*, vol. 9, no. 5, p. 433, May 2017, doi: 10.3390/rs9050433.

[22] B. Xu, W. Jiang, J. Shan, J. Zhang, and L. Li, "Investigation on the weighted RANSAC approaches for building roof plane segmentation from LiDAR point clouds," *Remote Sens.*, vol. 8, no. 1, p. 5, Dec. 2015, doi: 10.3390/rs8010005.

[23] E. Widyaningrum, B. Gorte, and R. Lindenbergh, "Automatic building outline extraction from ALS point clouds by ordered points aided Hough transform," *Remote Sens.*, vol. 11, no. 14, p. 1727, Jul. 2019, doi: 10.3390/rs11141727.

[24] H. Wang, C. Wang, H. Luo, P. Li, Y. Chen, and J. Li, "3-D point cloud object detection based on supervoxel neighborhood with Hough forest framework," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 4, pp. 1570–1581, Apr. 2015, doi: 10.1109/JSTARS.2015.2394803.

[25] X. Zhang, W. Wan, and L. Xiao, "Mean shift clustering segmentation and RANSAC simplification of color point cloud," in *Proc. Int. Conf. Audio, Lang. Image Process.*, 2014, pp. 837–841.

[26] L. Yang, Y. Li, X. Li, Z. Meng, and H. Luo, "Efficient plane extraction using normal estimation and RANSAC from 3D point cloud," *Comput. Standards Interfaces*, vol. 82, 2022, Art no. 103608, doi: 10.1016/j.csi.2021.103608.

[27] W. Song, L. Zhang, Y. Tian, S. Fong, J. Liu, and A. Gozho, "CNN-based 3D object classification using Hough space of LiDAR point clouds," *Human-Centric Comput. Inf. Sci.*, vol. 10, no. 1, p. 8, Dec. 2020, doi: 10.1186/s13673-020-00228-8.

[28] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020, doi: 10.1109/ACCESS.2020.2988796.

[29] H. Chen, T. Xie, M. Liang, W. Liu, and P. X. Liu, "A local tangent plane distance-based approach to 3D point cloud segmentation via clustering," *Pattern Recognit.*, vol. 137, Aug. 2023, Art. no. 109307, doi: 10.1016/j.patcog.2023.109307.

[30] H. Fu, H. Li, Y. Dong, F. Xu, and F. Chen, "Segmenting individual tree from TLS point clouds using improved DBSCAN," *Forests*, vol. 13, no. 4, p. 566, Apr. 2022, doi: 10.3390/f13040566.

[31] X. Chen., "Study on single-tree segmentation of Chinese fir plantations using coupled local maximum and height-weighted improved K-means algorithm," *Forests*, vol. 14, no. 11, p. 2130, 2023.

[32] T. Czerniawski, B. Sankaran, M. Nahangi, C. Haas, and F. Leite, "6D DBSCAN-based segmentation of building point clouds for planar object classification," *Autom. Construct.*, vol. 88, pp. 44–58, Apr. 2018, doi: 10.1016/j.autcon.2017.12.029.

[33] H. Chen, M. Liang, W. Liu, W. Wang, and P. X. Liu, "An approach to boundary detection for 3D point clouds based on DBSCAN clustering," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108431.

[34] A. L. M. Levada, "PCA-KL: A parametric dimensionality reduction approach for unsupervised metric learning," *Adv. Data Anal. Classification*, vol. 15, no. 4, pp. 829–868, 2021, doi: 10.1007/s11634-020-00434-3.

[35] A. Keshvari, "Segmented concave least squares: A nonparametric piecewise linear regression," *Eur. J. Oper. Res.*, vol. 266, no. 2, pp. 585–594, Apr. 2018, doi: 10.1016/j.ejor.2017.10.006.

YUDONG ZHOU received the B.S. degree in communication engineering from the North University of China, Taiyuan, China, in 2022, where he is currently pursuing the master's degree in information fusion and identification. His research interests include LiDAR data processing, target recognition, and space planning.

FENGBAO YANG received the Ph.D. degree in measuring and testing technologies and instruments from the North University of China, Taiyuan, in 2003. From 2004 to 2007, he was a Postdoctoral Research Fellow with Beijing Institute of Technology. He is currently a Full Professor with the North University of China. His research interests include quantitative remote sensing and information fusion theory of uncertainty.

MIN GAO received the B.S. degree in communication engineering from the Central South University of Forestry and Technology, Changsha, China. She is currently pursuing the Ph.D. degree with the North University of China, Taiyuan, China. Her research interests include UAV remote sensing and LiDAR data processing.