

# Indoor Clutter Object Removal Method for an As-Built Building Information Model Using a Two-Dimensional Projection Approach

Sung-Jae Bae and Jung-Yeol Kim \*

Department of Architectural Engineering, Inha University, Incheon 22212, Republic of Korea; bsj4695@naver.com

\* Correspondence: jungkim@inha.ac.kr

**Abstract:** Point cloud data are used to create an as-built building information model (as-built BIM) that reflects the actual status of any building, whether being constructed or already completed. However, indoor clutter objects in the point cloud data, such as people, tools, and materials, should be effectively eliminated to create the as-built BIM. In this study, the authors proposed a novel method to automatically remove indoor clutter objects based on the Manhattan World assumption and object characteristics. Our method adopts a two-dimensional (2D) projection of a 3D point cloud approach and utilizes different properties of indoor clutter objects and structural elements in the point cloud. Voxel-grid downsampling, density-based spatial clustering (DBSCAN), the statistical outlier removal (SOR) filter, and the unsupervised radius-based nearest neighbor search algorithm were applied to our method. Based on the evaluation of our proposed method using six actual scan datasets, we found that our method achieved a higher mean accuracy (0.94), precision (0.97), recall (0.90), and F1 core (0.93) than the commercial point cloud processing software. Our method shows better results than commercial point cloud processing software in classifying and removing indoor clutter objects in complex indoor environments acquired from construction sites. As a result, assumptions about the different properties of indoor clutter objects and structural elements are being used to identify indoor clutter objects. Additionally, it is confirmed that the parameters used in the proposed method could be determined by the voxel size once it is decided during the downsampling process.

**Keywords:** point cloud; indoor clutter objects; as-built BIM; two-dimensional projection; DBSCAN

**Citation:** Bae, S.-J.; Kim, J.-Y. Indoor Clutter Object Removal Method for an As-Built Building Information Model Using a Two-Dimensional Projection Approach. *Appl. Sci.* **2023**, *13*, 9636. <https://doi.org/10.3390/app13179636>

Academic Editor: Jongwon Seo

Received: 18 July 2023

Revised: 21 August 2023

Accepted: 22 August 2023

Published: 25 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

An as-built building information model (BIM) is an innovative tool for construction management such as quality control, progress monitoring, inspection, safety monitoring, and so forth [1–5]. Additionally, as-built BIM is also useful for the control and monitoring of automated construction equipment and robots [6,7]. In order to create an as-built BIM, it is necessary to collect the actual status of construction sites or an existing building. The most common methods to gather status data are laser scanning or photogrammetry [8], and their output is point cloud data. However, the point cloud data typically includes unnecessary objects, such as people, tools, and materials, which are considered clutter in the point cloud data [9,10]. These clutter objects can negatively impact the accuracy and speed of automated as-built BIM creation, particularly in terms of point-cloud semantic segmentation of building elements [11,12]. Therefore, it is critical to remove these indoor clutter objects from the point cloud. Removing outdoor clutter objects from point cloud data can be easily accomplished manually. However, it is relatively difficult to remove indoor clutter objects due to the large number of items and their interconnection with building elements such as floors, walls, and ceilings [13]. Therefore, manual elimination

of indoor clutter objects is an inefficient process, which is why automation is necessary to develop efficient as-built BIM creation methods [14].

Removing indoor clutter objects intersects with several research areas, including the automatic creation of as-built BIM, the automation of two-dimensional (2D) floor plan generation, and point cloud semantic segmentation. The authors believe that removing indoor clutter objects could enhance the outcomes of critical processes in Scan-to-BIM, such as point cloud semantic segmentation, line fitting, and plane fitting. The proposed method aimed to obtain an x–y plane from which indoor clutter objects were removed. The obtained x–y plane could be used to remove indoor clutter objects from the point cloud.

The most typical method for removing indoor clutter objects is the line-fitting-based method. The line-fitting-based approach identifies outliers using obtained lines or planes that represent the additional elements that need to be preserved [9,10,14–16]. However, the line-fitting-based method often ignores certain types of elements, such as indoor columns or walls, depending on their parameter values defined a priori, and it may not accurately reflect the thickness of inner walls [14,16]. Furthermore, while an appropriate horizontal x–y plane is essential, related studies have either manually selected this plane or determined it based on a z-axis value. Another category of removal methods are feature-based approaches. However, it is challenging to set an appropriate parameter value in clustering [17,18], and there is a limitation near the contact area of different objects [19]. In order to overcome these limitations, active research is being conducted on methods that use geometric features such as the normal vectors [18,20–22], as well as those that use features extracted from deep learning models, such as pointNet [23]. Recently, deep learning models have been utilized to conduct semantic segmentation into learned classes. Subsequently, these models label any unlearned point cloud as clutter. While these methods classify structural elements based on distinct geometric features, distinguishing elements with similar features, such as walls and columns, remains challenging. Furthermore, removing indoor clutter objects that have geometrical features similar to structural elements poses a significant challenge.

This study proposes an indoor clutter object removal method that works even when the geometric features of indoor clutter objects are similar to those of structural elements. Additionally, the proposed method can extract a representative line of structural elements and identify indoor clutter objects and structural elements based on the Manhattan World (MW) assumption. The study is based on the following two assumptions:

- Structural elements, such as columns and walls, are in contact with the floor and ceiling.
- Indoor clutter objects mainly exist on the floor and do not extend to the ceiling.

The proposed method was developed based on these two assumptions and the 2D projection approach. The method uses voxel-grid downsampling, DBSCAN, a statistical outlier removal (SOR) filter, and an unsupervised radius-based nearest neighbor search algorithm.

## 2. Literature Review

The proposed method was developed based on a 2D projection approach of the point cloud. In this section, the authors provide an overview of the literature on line-fitting-based and feature-based methods.

### 2.1. Line-Fitting-Based Method

The following studies utilized the line-fitting-based method to extract structural element lines. The RANSAC-based method is a model-fitting method that can perform well in the presence of outliers [24]. It is particularly suitable for fitting 2D lines or planes of structural elements from a point cloud. Babacan et al. [10] created various horizontal slices based on the floor and ceiling to create as-built BIMs. The horizontal slice with the fewest indoor clutter objects was selected for RANSAC application. The 2D line derived from the

structural elements was used in the as-built BIM. Pouraghdam et al. [15] selected the horizontal slice intended for RANSAC applications 0.3 m below the ceiling. Gankhuyag and Han [14] determined the z-coordinate of the floor and the ceiling to determine indoor height. The horizontal slice for RANSAC application was then extracted from the z-coordinate, which was estimated based on the multiplication of the threshold and the indoor height.

The RANSAC-based method is directly applicable to both horizontal slices and point clouds. When applied to point clouds, the RANSAC-based method includes fitting the plane of walls or columns as general structural elements. Previtali et al. [9] applied RANSAC to a point cloud to fit the plane of structural elements, and the location of the plane after fitting was used in as-built BIM modeling. Wang et al. [16] used RANSAC to detect wall candidates from a point cloud. The detected wall candidate was used in the line segment, and a 2D floor plan was created.

Several studies applied other line-fitting methods to obtain more accurate lines of structural elements. Kim and Lee [25] applied a voxelization-based method to obtain structural element lines. They utilized the horizontal slice between the floor and ceiling, which is the most clutter-free slice. Martens and Blankenbach [26] adopted morphological operations on the x–y plane of a point cloud to remove indoor noise. Wu et al. [27] developed a Modified Ring-Stepping Clustering (M-RSC) method to extract structural element lines in complex indoor environments. However, their method involved a manual task to remove indoor clutter object data. Macher et al. [28] applied the Maximum Likelihood Estimation SAMple Consensus (MLE-SAC) method to extract structural element lines. After that, the indoor clutter objects were removed from the structural element point that was obtained from the structural element line.

The line-fitting-based method can generate a point cloud of structural elements and 2D floor plans, considering indoor clutter objects in the point cloud. Notably, the RANSAC-based approach has been robustly employed to extract lines or planes of structural elements. However, it struggles to accurately represent the thickness of inner walls or columns due to the challenges in optimizing the right parameters. Furthermore, line-fitting-based methods are required to select an appropriate horizontal slice. However, it is difficult to define the appropriate horizontal slice that is least affected by the indoor clutter object data.

## 2.2. Feature-Based Method

The feature-based method was primarily developed for semantic segmentation, and several studies have applied this approach to segment indoor point clouds. These methods can be categorized into clustering-based methods and deep learning-based methods.

The clustering-based method segments targets, such as the structural elements and the indoor object, and accounts for the indoor clutter objects using an approach that determines the point cloud of the clutter aside from the target. For example, Yang and Wu [22] used pointNet features to perform clustering-based segmentation of point clouds by applying DBSCAN to two selected features. Yao et al. [29] applied supervoxel and DBSCAN to remove the point cloud of the floor, while Chen et al. [19] proposed a new density-based clustering method to segment indoor objects. Czerniawski et al. [18] determined the normal vector from the point cloud and applied DBSCAN to the generated sparse normal space to preserve indoor objects and remove planar elements. Stojanovic et al. [12] segmented a point cloud based on the z-axis into three segments for the construction of an as-built BIM and used the point cloud of the middle segment to create the floor plan. They applied k-means clustering to the x–y plane to extract the structural elements. Romero-Jarén and Arranz [30] proposed an automatic segmentation and classification method based on geometric feature clustering. The indoor clutter was categorized into virtual other objects, virtual objects on the floor, and virtual objects on the ceiling. The main approach of this study was based on the fact that indoor clutter objects have non-planar characteristics.

The deep learning-based method is widely used for semantic segmentation of point clouds. In general, these methods identify and learn the key structural elements or primary objects of interest. They can also classify any unlearned point cloud as clutter. Park et al. [2] applied pointNet [23] for semantic segmentation, and Kim and Kim [31] applied DGCNN [32]. Perez-Perez et al. [33] developed their own model to apply scan-to-BIM. Besides, other deep learning models such as RandLa-Net [34] and pointNet++ [35] have recently been actively developed and applied to Scan-to-BIM research [36,37], segmenting both structural elements and indoor clutter objects. However, the classification performance between objects with similar geometric features, such as walls and columns, is still unsatisfactory. Therefore, if certain indoor clutter objects have similar geometric features with structural elements, it would be difficult to clearly identify those indoor clutter objects as clutter.

As the literature above mentions, the feature-based method is effective in utilizing the geometrical features of the object to be preserved. However, they have limitations in accurately determining indoor clutter objects that have similar geometrical features to the structural elements. Also, they tend to create errors when the clutter objects are close to the structural elements. Therefore, the authors believe that if the indoor clutter objects are removed before the semantic segmentation task, it may lead to more accurate as-built BIM modeling.

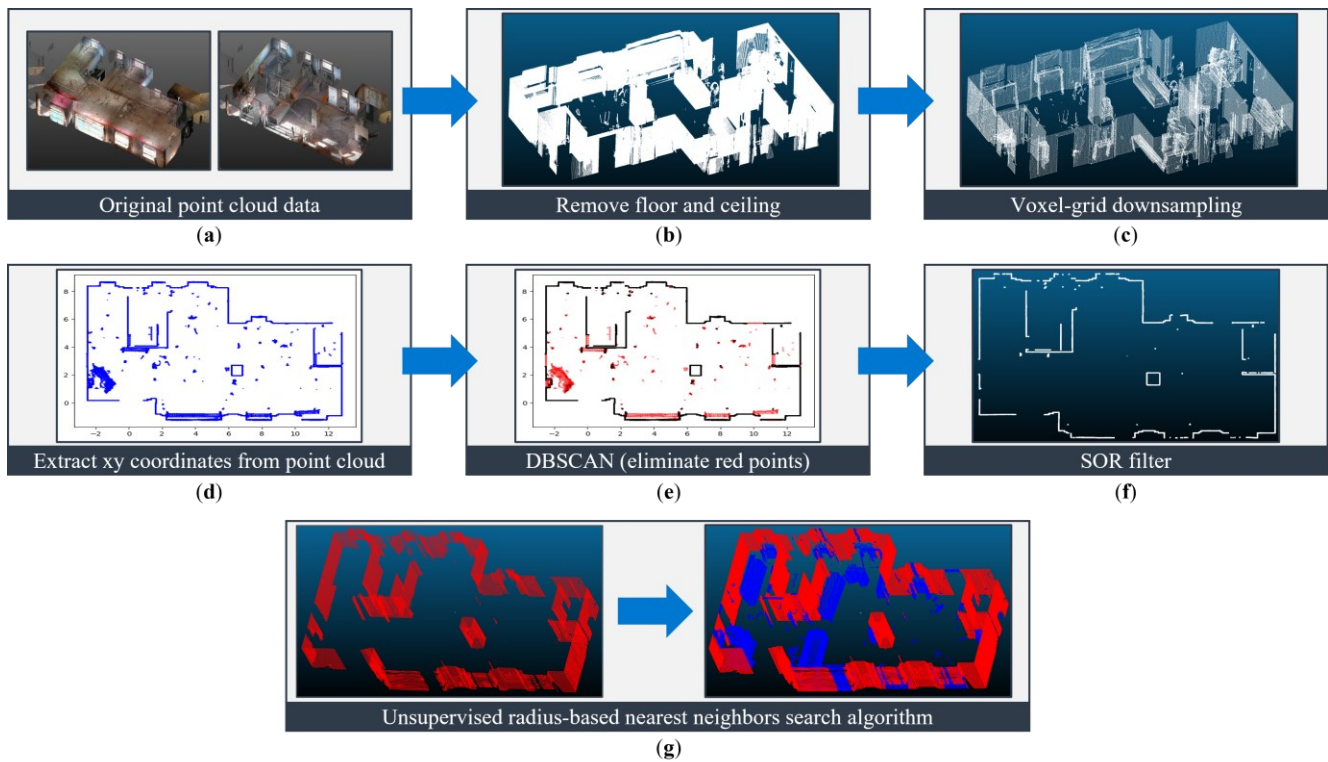
### 3. Methods

#### 3.1. Method Overview and Assumptions

The proposed method was designed to efficiently eliminate indoor clutter objects from the point cloud data obtained from a construction site. It is based on the assumptions that the structural elements are connected from the floor to the ceiling and that indoor clutter objects exist mostly on the floor and are not connected from the floor to the ceiling. The proposed method used voxel-grid downsampling, DBSCAN clustering, and the SOR filter to accurately identify and remove indoor clutter while preserving the structural elements in the point cloud.

Figure 1 illustrates the framework of the proposed method, which consists of seven steps (a to g). First, the proposed method receives an original point cloud as input data, as shown in Figure 1a. Subsequently, the point cloud near the floor and the ceiling is eliminated, as shown in Figure 1b. This is done to ensure that the method focuses on the indoor clutter on the floor, as the structural elements are assumed to be connected from the floor to the ceiling. The voxel-grid downsampling is then applied to generate a uniform point-cloud density, as shown in Figure 1c. This process reduces the computational burden and allows for faster processing. The x- and y-coordinates are then extracted from the point cloud that was downsampled with the voxel grid, as shown in Figure 1d.

Next, the extracted x- and y-coordinates of the structural element candidates are clustered through DBSCAN, as shown in Figure 1e. DBSCAN is used to group the points with similar spatial coordinates into clusters, which helps identify the structural elements. The SOR filter is then applied to obtain more accurate structural element candidates, and the indoor clutter objects are removed, as shown in Figure 1f. The SOR filter is used to smooth the surface of the structural element candidates and remove the remaining noise data. Finally, the obtained structural element candidates are used to search the structural elements in the point cloud using an unsupervised radius-based nearest neighbor search algorithm, as shown in Figure 1g. Once the voxel size is determined, the proposed method operates automatically, except for the SOR filtering step. The parameters required for each step are automatically determined based on the voxel size. The details of these steps are explained in Section 3.2.



**Figure 1.** Proposed method framework.

### 3.2. Removal of the Floor and Ceiling

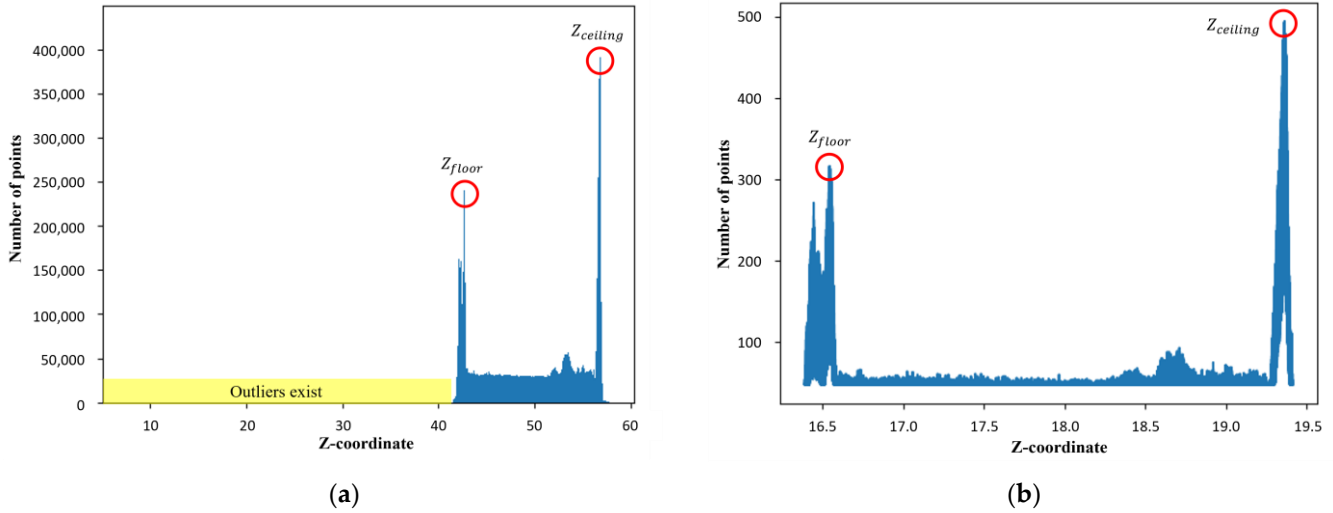
The proposed method was developed based on the Manhattan World assumption. According to the abovementioned assumptions, the histogram of the number of points according to the z-coordinate of the point cloud shows a sharp rise near the floor and ceiling, as seen in Figure 2a. However, there may be outdoor outliers of the target object based on the z-axis. To consider these outliers, the proposed method uses the average of the number of points based on the z-axis. This method automatically obtains the z-coordinates of the floor and ceiling by following these steps: (1) determining the average value of the number of points according to the z-coordinate; (2) preserving only the z-coordinate that has a greater number of points than the average number of points, as shown in Figure 2b; (3) distinguishing the low-ranking 30% and high-ranking 30% data based on the z-coordinate from the preserved z-coordinate and number of points data; and (4) determining the z-coordinates at which the number of points is maximized from the low- and high-ranking 30% data as the floor and ceiling z-coordinates, respectively.

The obtained floor and ceiling z-coordinates were used to remove the floor and ceiling. For this purpose, the value that added 0.2 m to the floor's z-coordinate ( $Z_{floor}$ ) was defined as  $Z_{min}$  (for the data intended to be preserved), as shown in Equation (1). In addition, the value that subtracted 0.2 m from the ceiling's z-coordinate ( $Z_{ceiling}$ ) was defined as  $Z_{max}$  (for the data intended to be preserved), as shown in Equation (2).  $Z_{mid}$  was determined using Equation (3), according to the determined  $Z_{min}$  and  $Z_{max}$ . The determined  $Z_{min}$ ,  $Z_{mid}$ , and  $Z_{max}$  were used to determine the parameters of the DBSCAN, SOR filter, and unsupervised radius-based nearest neighbor search algorithm in conjunction with the voxel size (used in the voxel-grid downsampling that will be performed subsequently). Figure 3 shows the data from which the floor and the ceiling were removed based on the z-coordinate of the determined  $Z_{min}$  and  $Z_{max}$ .

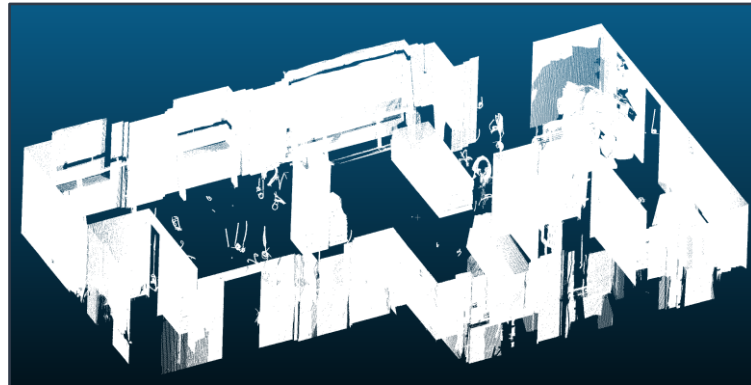
$$Z_{min} = Z_{floor} + 0.2 \quad (1)$$

$$Z_{max} = Z_{ceiling} - 0.2 \quad (2)$$

$$Z_{mid} = \frac{|Z_{max} - Z_{min}|}{2} \quad (3)$$



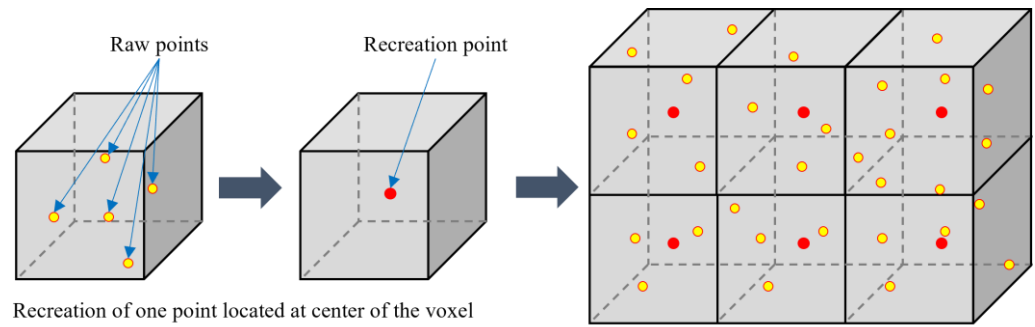
**Figure 2.** Histograms of the number of points as a function of the z-coordinate of the point cloud. (a) Histogram of the number of points as a function of the z-coordinate of the raw point cloud; (b) histogram of the number of points equal to or greater than the average number of points as a function of the z-coordinate.



**Figure 3.** Point cloud data with the floor and ceiling removed using  $Z_{min}$  and  $Z_{max}$ .

### 3.3. Voxel-Grid Downsampling

The point density of a point cloud obtained by a three-dimensional (3D) scanner varies depending on the distance between the scanner and the target of scanning. Objects that are far from the 3D scanner have a lower point density, whereas objects that are close to each other have a relatively higher point density. Meanwhile, the proposed method applies DBSCAN based on the x- and y-coordinates of the point cloud. From the perspective of DBSCAN operating on the x- and y-coordinates, the desired results cannot be obtained if the point density of the point cloud is not uniform. Therefore, to apply the proposed method, it is necessary for the point cloud to have a uniform point density. To achieve this, the method uses voxel-grid downsampling. Voxel-grid downsampling creates a voxel with a length equal to the previously defined voxel size, as shown in Figure 4. The voxel-grid downsampling recreates the representative point located at the center of the voxel instead of the points located inside the voxel. The voxel size was set to 0.05 m, which was appropriate for reducing the weight of the point cloud while preserving the shapes of the inner columns or the thickness of the inner wall. The authors utilized the voxel-grid downsampling algorithm in Open3d (ver. 0.17.0).

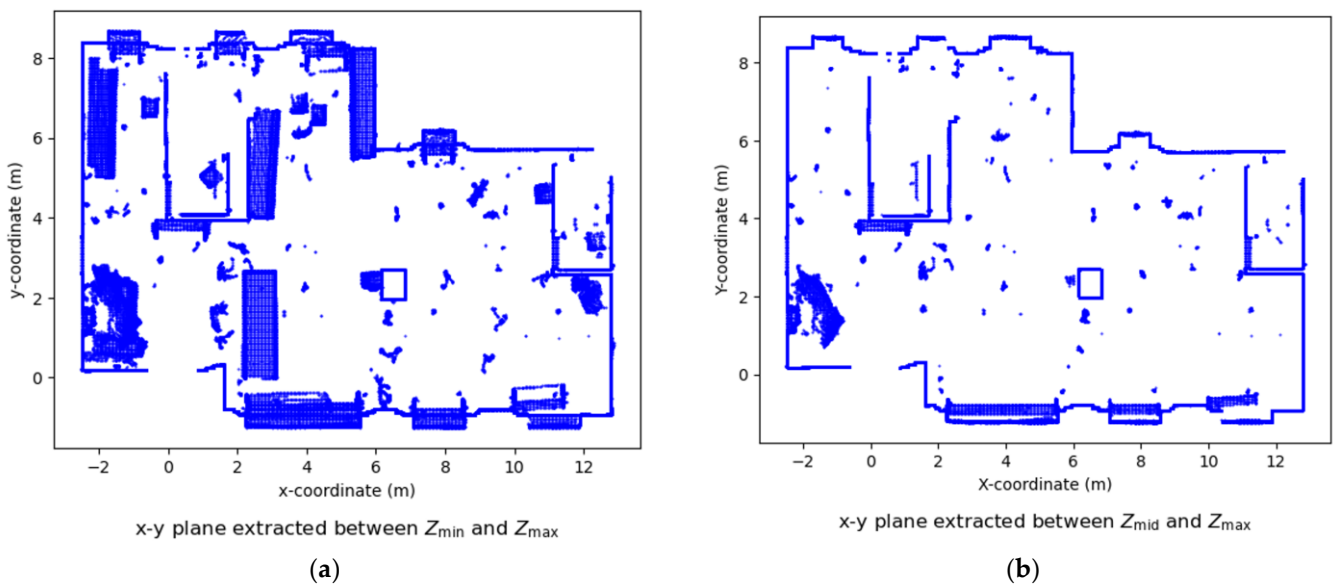


**Figure 4.** Concept of voxel-grid downsampling in the Open3D library.

### 3.4. Extraction of XY Coordinates from the Point Cloud between $Z_{mid}$ and $Z_{max}$

The x- and y-coordinates of the points to be used for DBSCAN are extracted between  $Z_{mid}$  and  $Z_{max}$  based on the z-axis. This is based on the second assumption in this study, that indoor clutter objects mainly exist on the floor and do not extend to the ceiling. The purpose of DBSCAN is to extract the x- and y-coordinates of the structural elements. Therefore, the proposed method uses the x- and y-coordinates between  $Z_{mid}$  and  $Z_{max}$ —which have a small impact on the indoor clutter objects—to increase the efficiency of DBSCAN.

Figure 5a shows the x–y plane where the x- and y-coordinates are plotted from the total point cloud between  $Z_{min}$  and  $Z_{max}$ . Figure 5b shows the x–y plane where the x- and y-coordinates are plotted from the point cloud between  $Z_{mid}$  and  $Z_{max}$ . It can be identified that the indoor clutter objects are less prevalent in Figure 5b than in Figure 5a. Therefore, the extraction of the x- and y-coordinates from the point cloud between  $Z_{mid}$  and  $Z_{max}$  is more efficient for DBSCAN.



**Figure 5.** Plots of the x–y planes with the x- and y-coordinates extracted from the point cloud obtained with voxel-grid downsampling. (a) Plot of the x–y plane with x- and y-coordinates extracted between  $Z_{min}$  and  $Z_{max}$ ; (b) plot of the x–y plane with x- and y-coordinates extracted between  $Z_{mid}$  and  $Z_{max}$ .

### 3.5. DBSCAN

DBSCAN requires two parameters: min points and epsilon. In the proposed method, the min points and epsilon are determined from the voxel size,  $Z_{mid}$ , and  $Z_{max}$ . Figure 6 shows an example of the point arrangement of a wall downsampled to a voxel size of 0.05

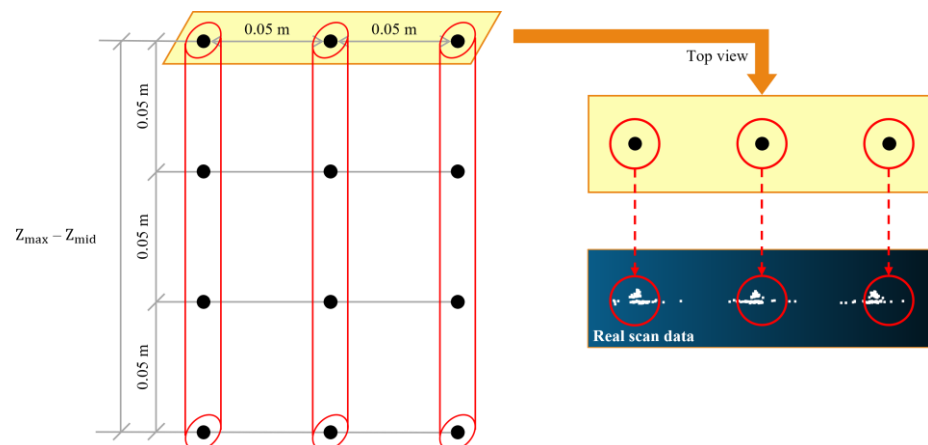


m as an ideal case. The height of the input point cloud data was determined as the difference between  $Z_{mid}$  and  $Z_{max}$ , and the points were arranged at 0.05 m intervals. When viewed from the top (x-y plane), a cluster is formed in which an integer number of points divided by 0.05 m from the height of the input point cloud data gather at one point on the x-y plane.

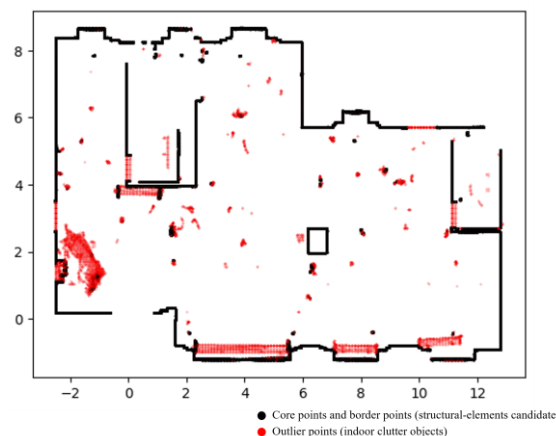
Therefore, the ideal epsilon value of DBSCAN operating in the 2D projection x-y plane was set to 0.05 m, which is the same as the voxel size in the proposed method, as shown in Equation (4). The ideal min point value was an integer value that eliminated the decimal point of the value after the division of the height of the input data (arranged in 0.05 intervals). In this study, the authors used an integer value obtained by subtracting two from the ideal min point value to account for possible omitted points, as shown in Equation (5). This approach allows for a more robust clustering result, even when some points may be missing from the data. Figure 7 shows the results following the use of the min points and the epsilon determined based on the height of the input data and the voxel size. The red points in Figure 7 indicate the data determined to be outlier points, and the black points include the core points and border points. The proposed method defines the structural element candidates as the core points and border points, which are the result of DBSCAN. The authors utilized DBSCAN algorithm in scikit-learn library (ver. 1.2.0).

$$\epsilon = \text{voxel size} \quad (4)$$

$$\text{min points} = \left\lfloor \frac{|Z_{max} - Z_{mid}|}{\text{voxel size}} \right\rfloor - 2 \quad (5)$$



**Figure 6.** Example of point arrangement of two-dimensional (2D) density-based spatial clustering of applications with noise (DBSCAN) input data and top views of real scan data.



**Figure 7.** DBSCAN results of the proposed method.

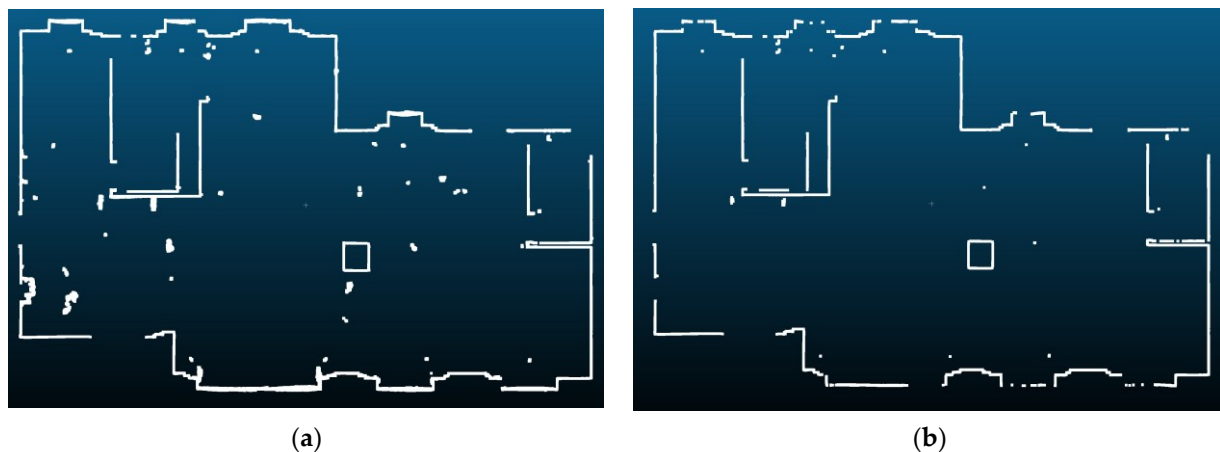


### 3.6. SOR Filter

The structural element candidates obtained from DBSCAN include the locations of structural elements, such as walls or columns, on the x–y plane. However, the points of indoor clutter objects may remain. If these coordinates are used to search the point cloud of structural elements, which is used subsequently, it could result in errors associated with the determination of the structural elements as indoor clutter objects. Therefore, it is necessary to remove the x- and y-coordinates of indoor clutter objects. To achieve this, the proposed method used the SOR filter in Cloudcompare (v2.1.2). The SOR filter eliminates noise data based on the maximum distance, which is determined according to the standard deviation multiplier threshold ( $sT$ ) to estimate the outlier and the number of points ( $k$ ), which is used to calculate the average distance ( $\delta$ ) and standard deviation ( $\sigma$ ), as shown in Equation (6).

$$\max distance = \delta + sT \times \sigma \quad (6)$$

In the proposed method, the value of  $k$  for the SOR filter is set to the number of min points obtained from DBSCAN. Additionally,  $sT$  is set to 0.1 considering a voxel size of 0.05 m. Figure 8a shows the structural element candidates before applying the SOR filter, and Figure 8b shows the results after applying the SOR filter.



**Figure 8.** Statistical outlier removal (SOR) filter application results: (a) before the use of the SOR filter; (b) after the use of the SOR filter.

### 3.7. Unsupervised Radius-Based Nearest Neighbor Search Algorithm

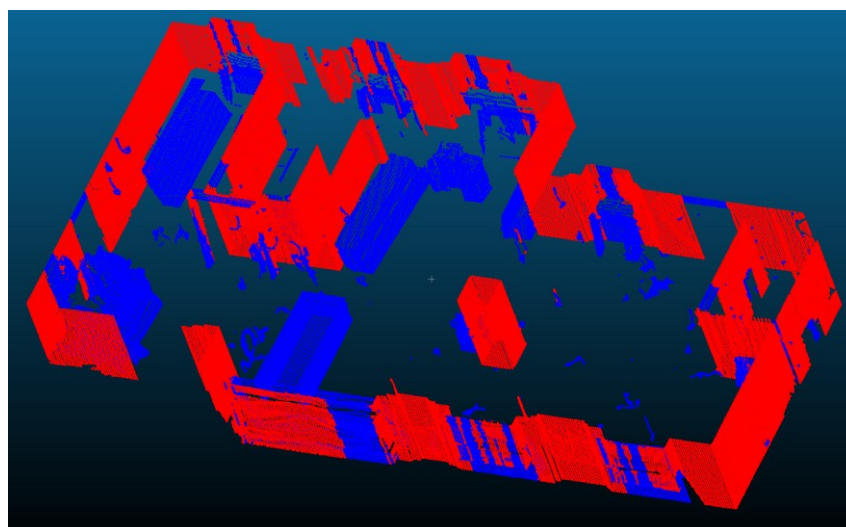
The refined structural element candidates obtained through SOR filtering were used to identify structural elements within the original point cloud. However, these structural element candidates do not directly correspond to the original point cloud due to the application of voxel-grid downsampling. To address this issue, nearest neighbor search algorithms such as unsupervised k-nearest neighbor search and unsupervised radius-based nearest neighbor search can be employed. In this study, the authors applied the unsupervised radius-based nearest neighbor search algorithm, which could determine the radius value based on the voxel size, as the setting for the  $k$  value in k-nearest neighbor was unclear. The radius value was set at 0.1 m, taking into consideration the voxel size of 0.05 m.

To ensure the effective operation of the unsupervised radius-based nearest neighbor search algorithm, a data structuring method must be selected. Popular options for data structuring include KD-tree and ball-tree, which can significantly reduce computational costs for nearest neighbor search. In this study, the results obtained from applying KD-tree and ball-tree to the original point cloud (52,150,674 points) and the downsampled point cloud (106,391 points) were compared. The comparison results are summarized in Table 1. When applied to the original point cloud, the ball-tree method took 1 min and 48

s, while the KD-tree method took 2 min and 11 s. Furthermore, when applied to the downsampled point cloud, the ball-tree method took 0.2 s, while the KD-tree method took 0.4 s. Consequently, this study employed the unsupervised radius-based nearest neighbor search algorithm using the ball-tree method for the exploration of structural elements within the original point cloud. Figure 9 shows the results of classifying indoor clutter objects and structural elements based on the proposed method. The authors utilized scikit-learn library (ver. 1.2.0).

**Table 1.** Comparison of KD-tree and ball-tree methods for the unsupervised radius-based nearest neighbor search algorithm (radius = 0.1 m).

Data Structuring Method	Original Point Cloud (52,150,674 Points)	Downsampled Point Cloud (106,391 Points)
KD-tree	2 m 11 s	0.4 s
Ball-tree	1 m 48 s	0.2 s

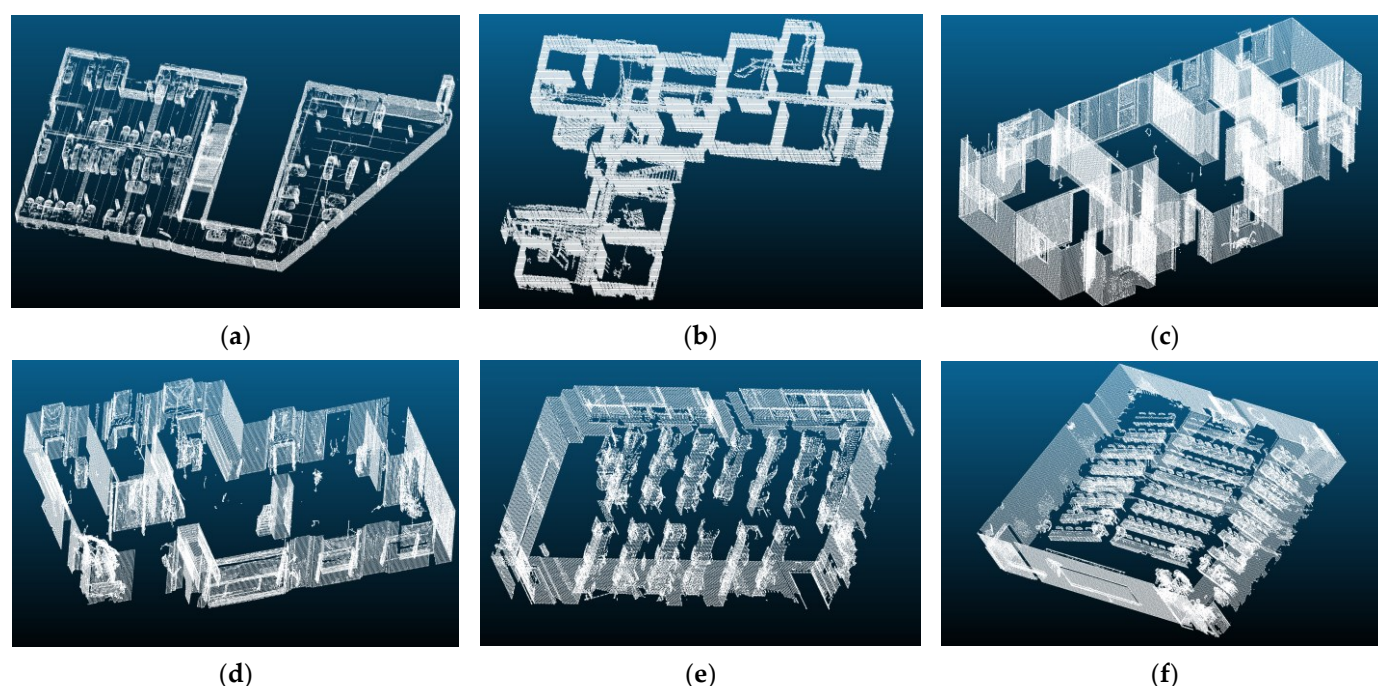


**Figure 9.** Structural elements and indoor clutter objects classification example based on the use of the unsupervised radius-based nearest neighbor search algorithm (red points: structural elements; blue points: indoor clutter objects).

## 4. Experiments

### 4.1. Experimental Data

To perform a comprehensive performance evaluation of the proposed method, six actual scan datasets were used in this study. Four of these datasets were point cloud data obtained from the parking lot, basement, and apartments 1 and 2 of an apartment complex construction site. The remaining two datasets were point cloud data obtained from lecture rooms 1 and 2 at Inha University. Figure 10 shows the six experimental datasets after voxel-grid downsampling and removal of floors and ceilings. Detailed descriptions of these datasets can be found in Table 2.



**Figure 10.** Experimental data downsampled to a voxel size of 0.05 m from where the floors and ceilings were removed: (a) parking lot, (b) basement, (c) apartment 1, (d) apartment 2, (e) lecture room 1, and (f) lecture room 2.

**Table 2.** Details of datasets used in this study.

Dataset	Initial Number of Points (Size)	Downsampled Points (Size)	Volume Size of Point Cloud (m)	Indoor Complexity
Parking lot	158,575,772 (6.3 GB)	877,184 (10.0 MB)	$100.5 \times 52.6 \times 2.2$	high
Basement	16,518,293 (0.7 GB)	333,587 (3.8 MB)	$26.3 \times 21.6 \times 2.5$	low
Apartment 1	121,652,038 (3.9 GB)	149,896 (1.7 MB)	$10.2 \times 16.7 \times 2.3$	low
Apartment 2	52,150,675 (1.4 GB)	106,391 (1.2 MB)	$15.4 \times 9.9 \times 2.3$	low
Lecture room 1	22,082,415 (0.7 GB)	65,027 (0.8 MB)	$12.6 \times 7.7 \times 2.2$	high
Lecture room 2	43,856,412 (1.4 GB)	116,533 (1.9 MB)	$12.6 \times 15.5 \times 3.3$	high

#### 4.2. Performance Evaluation and Metrics

To evaluate the proposed method's performance appropriately, this study applied voxel-grid downsampling and removed the floor and ceiling from the point cloud data. The ground truth data was manually labelled and classified into structural elements and indoor clutter objects. The red points indicate the structural elements, while the blue points indicate indoor clutter objects. This study compared the ground truth labels with the results of classifying the six experimental datasets into structural elements and indoor clutter objects using the proposed method.

Moreover, the proposed method's performance was compared with the Auto-Classify Indoor function of commercial point cloud processing software. Auto-Classify Indoor automatically classifies the point cloud into indoor elements, including walls, floors, ceilings, and other remaining parts, using feature-based methods. The performance of the proposed method was evaluated using metrics such as accuracy, precision, recall, and F1 score. These metrics are calculated based on the values from the confusion matrix. True Positive (TP) refers to instances that are actually structural elements and are also predicted as structural elements. True Negative (TN) refers to instances that are actually not structural elements, clutter objects in this study, and are not predicted as structural elements. False Positive (FP) refers to instances that are actually not structural elements but are

predicted as structural elements. False Negative (FN) refers to instances that are actually structural elements but are not predicted as structural elements. Each metric is calculated according to Equations (7)–(10).

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (7)$$

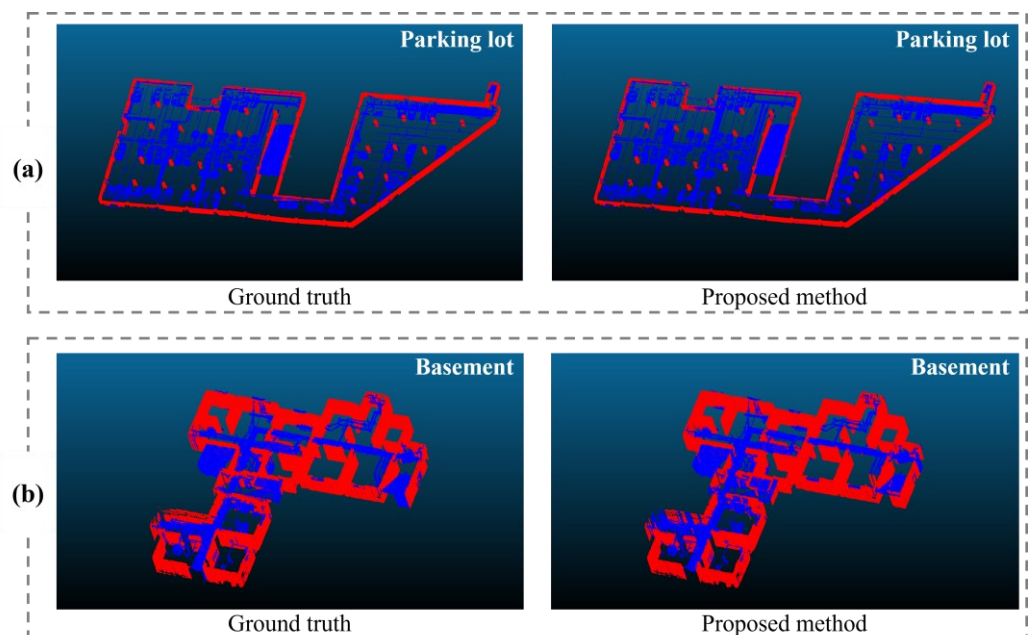
$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

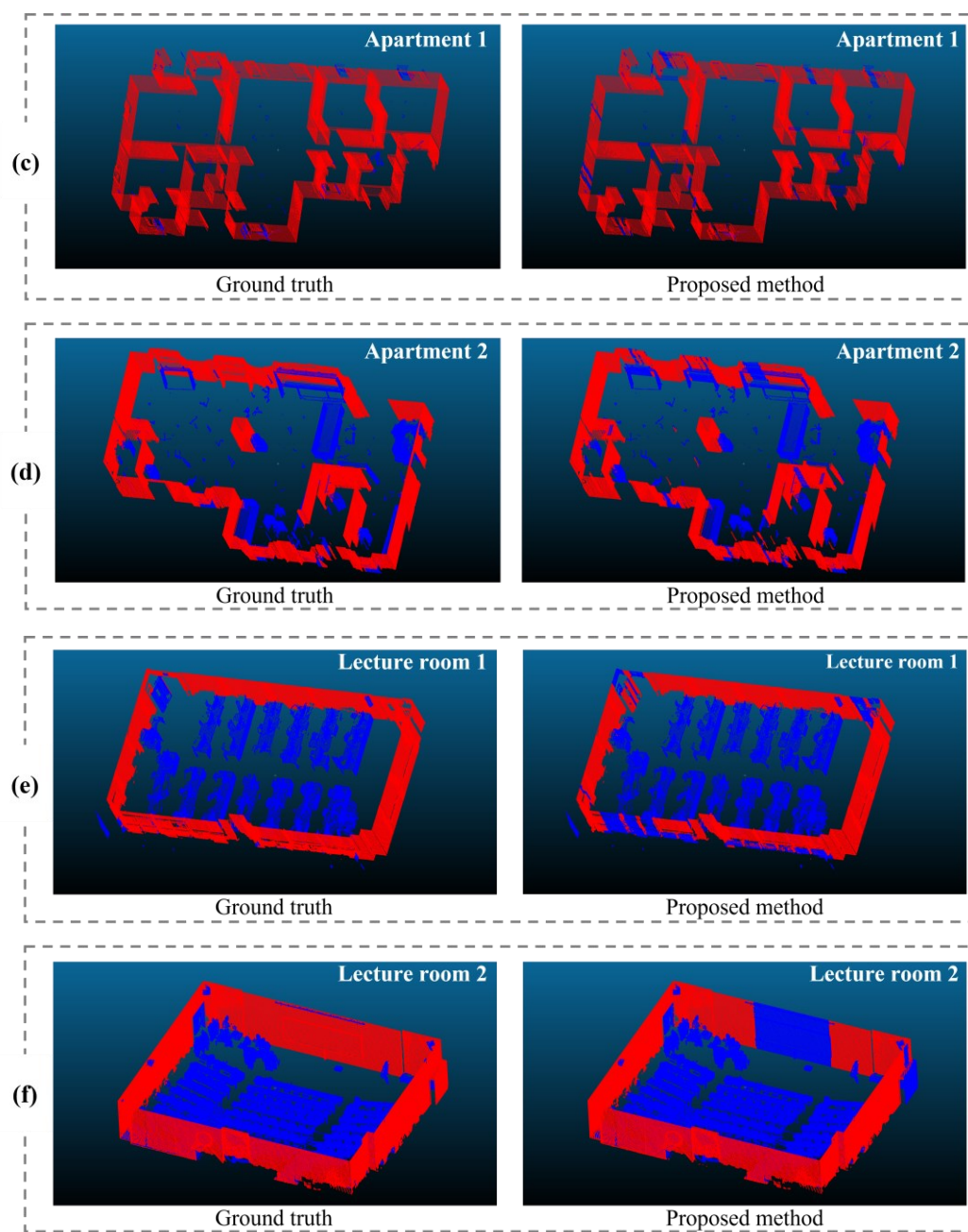
$$F1 \text{ score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

#### 4.3. Experimental Results

All experiments were performed on a PC running Windows 10, equipped with an AMD Ryzen 9 5900X 12-Core Processor running at 3.70 GHz and 64 GB of RAM. This study aimed to classify the six datasets described in Figure 10 and Table 2 into structural elements and indoor clutter objects for removing indoor clutter objects in the point cloud. Figure 11 displays the ground truth and classification results of the structural elements and indoor clutter objects of the six datasets using the proposed method. The results of the ground truth-based accuracy, precision, recall, and F1 score are summarized in Table 3. The proposed method based on the six datasets achieved an average accuracy of 0.94, an average precision of 0.97, an average recall of 0.90, and an average F1 score of 0.94. Table 4 presents the classification results of the experimental data into structural elements and indoor clutter objects using the Auto-Classify Indoor function of commercial software. Table 5 presents the processing times of the proposed method for each dataset. The times were calculated except for the SOR filter step because it is manually performed. As observed in our experiments, the SOR filter step required a time of 1 to 2 s for its operation using commercialized software. Therefore, the times listed in Table 5 satisfactorily represent the time needed for the operation of the proposed method. Figure 12 presents a graph that compares the average performance of the proposed method and the Auto-Classify Indoor function of commercial software.







**Figure 11.** Experimental results. (a) Parking lot, (b) basement, (c) apartment 1, (d) apartment 2, (e) lecture room 1, and (f) lecture room 2.

**Table 3.** Performance outcomes of the proposed method.

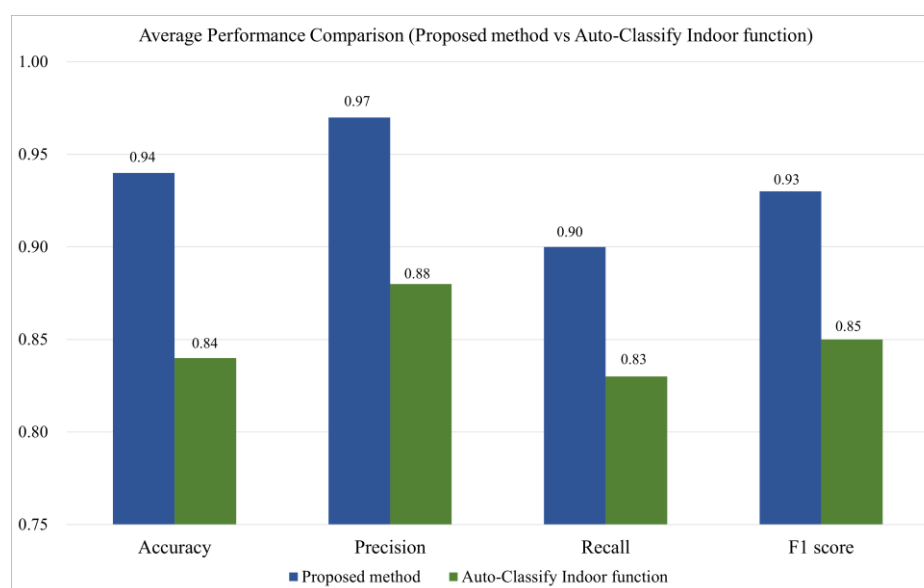
Dataset	Accuracy	Precision	Recall	F1 Score
Parking lot	0.98	0.98	0.97	0.98
Basement	0.93	0.88	0.89	0.88
Apartment 1	0.96	0.99	0.97	0.98
Apartment 2	0.94	0.99	0.86	0.92
Lecture room 1	0.90	0.97	0.83	0.89
Lecture room 2	0.94	0.99	0.86	0.92
Average performance	0.94	0.97	0.90	0.93

**Table 4.** Performance outcomes of the Auto-Classify Indoor function of commercial software.

Dataset	Accuracy	Precision	Recall	F1 Score
Parking lot	0.86	0.93	0.73	0.81
Basement	0.78	0.62	0.80	0.70
Apartment 1	0.74	0.98	0.75	0.85
Apartment 2	0.79	0.82	0.87	0.84
Lecture room 1	0.90	0.98	0.84	0.91
Lecture room 2	0.97	0.95	0.97	0.96
Average performance	0.84	0.88	0.83	0.85

**Table 5.** Time taken to run the proposed methodology for each dataset.

Dataset	Time (s)
Parking lot	300.3
Basement	38.2
Apartment 1	171.6
Apartment 2	65.3
Lecture room 1	28.2
Lecture room 2	62.4

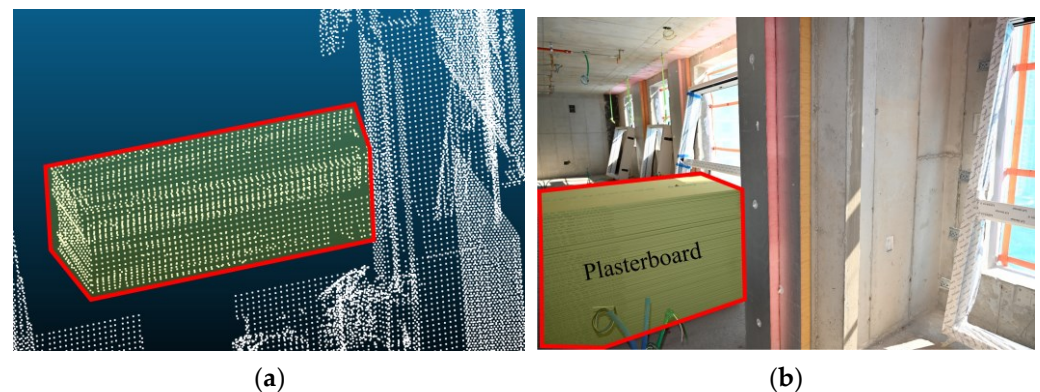
**Figure 12.** Comparison between the proposed method and the commercial software Auto-Classify Indoor function.

#### 4.4. Discussion

The experimental results classified the structural elements and the indoor clutter of the six datasets with an average accuracy of 0.94, an average precision of 0.97, an average recall of 0.90, and an average F1 score of 0.93. In addition, the proposed method yielded improved performances for all evaluation metrics in comparison to the Auto-Classify Indoor function of the commercial software, as shown in Figure 12. In particular, all the metrics from the parking lot dataset and the apartment 1 dataset were 0.96 or higher. The improved performance of the apartment 1 dataset was attributed to the set's relatively low indoor complexity. Meanwhile, improved performance was observed in the case of the parking lot's dataset (among all datasets), despite its high indoor complexity.

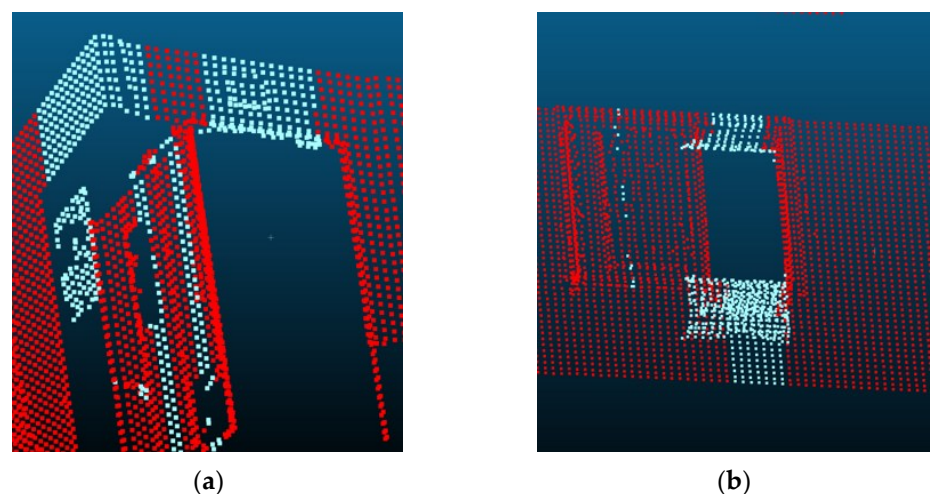
The proposed method yielded higher performance than the feature-based Auto-Classify Indoor function; this can be explained using Figure 13. Figure 13a shows the point cloud that is classified into the indoor Auto-Classify function and the structural elements,

and Figure 13b is the actual target object. The highlighted area of Figure 13b is where the plasterboard was loaded on the floor. Therefore, it is appropriate to remove the plasterboard as indoor clutter objects. However, the point cloud of the plasterboard in the Auto-Classify Indoor function has vertical geometrical features, such as the wall, and it has therefore been classified as structural elements. Thus, the proposed method can operate powerfully even for indoor clutter objects with geometrical features similar to those of the structural elements.



**Figure 13.** Comparison of images of the apartment 2 dataset and the actual target object. (a) Point cloud data that Auto-Classify Indoor function classified as structural elements; (b) photograph of an actual target object.

The advantages of the proposed method were summarized above; however, it did not perform outstandingly in all experimental datasets. In particular, the results from datasets from lecture rooms 1 and 2 exhibited high performances in the case of the Auto-Classify Indoor function, but by a narrow margin. This was because the proposed method performed weakly near the window and the door, as shown in Figure 14. The proposed method was developed based on a 2D projection approach. The point cloud obtained from the window is unstable, and the points of the structural elements that exist above the door can be classified as indoor clutter in the DBSCAN of the proposed method. The authors reviewed the experimental results of the proposed method and found that most errors occurred at the upper part of the window and the door. Therefore, it is expected that an improvement in these limitations in the future could remove the indoor clutter objects more accurately.



**Figure 14.** Examples of erroneous classifications of the point cloud at the window and door. (a) Point cloud classification results near the door of lecture room 2; (b) point cloud classification results near the windows of apartment 2 (red points: structural elements; cyan points: indoor clutter objects).



## 5. Conclusions

In this study, the authors proposed a novel method to determine the indoor clutter objects based on the assumptions that (a) the structural elements stretched from the floor to the ceiling and (b) the indoor clutter objects existed on the floor and did not stretch to the ceiling. The proposed method includes the removal of the floor and ceiling, voxel-grid downsampling, DBSCAN, SOR filter application, and an unsupervised radius-based nearest neighbor search algorithm.

The experiment with the six scan datasets from actual sites showed higher accuracy, precision, recall, and F1 scores than conventional methods in identifying indoor clutter objects. Specifically, the proposed method achieved an accuracy of 0.94, a precision of 0.97, a recall of 0.90, and an F1 score of 0.94. When compared to the Auto-Classify Indoor function of commercial point-cloud processing software, the proposed method showed higher performances by 0.10, 0.09, 0.07, and 0.09 in terms of accuracy, precision, recall, and F1 score, respectively.

The contributions of this study are as follows:

- The proposed method can accurately determine and remove indoor clutter objects with higher performance than commercial software;
- The proposed method can extract an appropriate x-y plane that represents structural elements, including inner walls and columns;
- The proposed method can identify indoor clutter objects among objects with similar geometrical features to structural elements;
- The parameters of DBSCAN, the SOR filter, and the unsupervised radius-based nearest neighbor search algorithm used in the proposed method are automatically determined by the voxel size.

However, the proposed method has some limitations in accurately determining the structural elements near windows and doors. In the future, the authors plan to improve this method to more accurately determine the structural elements at all locations, including regions near windows and doors. Furthermore, we will also consider pipes that are installed horizontally. The authors will adopt the proposed method in the Scan-to-BIM process to improve point cloud semantic segmentation results. The proposed method has the potential to be applied in various fields, such as architecture, civil engineering, and interior design.

**Author Contributions:** Conceptualization, S.-J.B. and J.-Y.K.; Methodology, S.-J.B. and J.-Y.K.; Software, S.-J.B.; Validation, S.-J.B.; Formal analysis, S.-J.B.; Investigation, S.-J.B.; Resources, S.-J.B.; Writing—original draft, S.-J.B.; Visualization, S.-J.B.; Writing—review and editing, J.-Y.K.; Supervision, J.-Y.K.; Project administration, J.-Y.K.; Funding acquisition, J.-Y.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by grant No. 1615012983 from the Digital-based Building Construction and Safety Supervision Technology Research Program funded by the Ministry of Land, Infrastructure and Transport of the Korean Government.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Volk, R.; Stengel, J.; Schultmann, F. Building Information Modeling (BIM) for Existing Buildings—Literature Review and Future Needs. *Autom. Constr.* **2014**, *38*, 109–127. <https://doi.org/10.1016/j.autcon.2013.10.023>.
2. Park, J.; Kim, J.; Lee, D.; Jeong, K.; Lee, J.; Kim, H.; Hong, T. Deep Learning-Based Automation of Scan-to-BIM with Modeling Objects from Occluded Point Clouds. *J. Manag. Eng.* **2022**, *38*, 04022025. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0001055](https://doi.org/10.1061/(ASCE)ME.1943-5479.0001055).
3. Wang, T.; Chen, H.-M. Integration of Building Information Modeling and Project Management in Construction Project Life Cycle. *Autom. Constr.* **2023**, *150*, 104832. <https://doi.org/10.1016/j.autcon.2023.104832>.

4. Tang, P.; Huber, D.; Akinci, B.; Lipman, R.; Lytle, A. Automatic Reconstruction of As-Built Building Information Models from Laser-Scanned Point Clouds: A Review of Related Techniques. *Autom. Constr.* **2010**, *19*, 829–843. <https://doi.org/10.1016/j.autcon.2010.06.007>.
5. Azhar, S.; Khalfan, M.; Maqsood, T. Building Information Modeling (BIM): Now and Beyond. *Australas. J. Constr. Econ. Build.* **2012**, *12*, 15–28. <https://doi.org/10.3316/informat.013120167780649>.
6. Asadi, K.; Ramshankar, H.; Noghabaei, M.; Han, K. Real-Time Image Localization and Registration with BIM Using Perspective Alignment for Indoor Monitoring of Construction. *J. Comput. Civ. Eng.* **2019**, *33*, 04019031. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000847](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000847).
7. Kim, I.; Lee, C. Development of Video Shooting System and Technique Enabling Detection of Micro Cracks in the Tunnel Liningwhile Driving. *J. Korean Soc. Hazard Mitig* **2018**, *18*, 217–229. <https://doi.org/10.9798/KOSHAM.2018.18.5.217>.
8. Wang, Q.; Kim, M.-K. Applications of 3D Point Cloud Data in the Construction Industry: A Fifteen-Year Review from 2004 to 2018. *Adv. Eng. Inform.* **2019**, *39*, 306–319. <https://doi.org/10.1016/j.aei.2019.02.007>.
9. Previtali, M.; Díaz-Vilariño, L.; Scaioni, M. Indoor Building Reconstruction from Occluded Point Clouds Using Graph-Cut and Ray-Tracing. *Appl. Sci.* **2018**, *8*, 1529. <https://doi.org/10.3390/app8091529>.
10. Babacan, K.; Jung, J.; Wichmann, A.; Jahromi, B.A.; Shahbazi, M.; Sohn, G.; Kada, M. Towards Object Driven Floor Plan Extraction from Laser Point Cloud. *ISPRS-Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *49*, 3–10. <https://doi.org/10.5194/isprs-archives-XLI-B3-3-2016>.
11. Nikoohemat, S.; Peter, M.; Oude Elberink, S.; Vosselman, G. Semantic Interpretation of Mobile Laser Scanner Point Clouds in Indoor Scenes Using Trajectories. *Remote Sens.* **2018**, *10*, 1754. <https://doi.org/10.3390/rs10111754>.
12. Stojanovic, V.; Trapp, M.; Richter, R.; Döllner, J. Generation of Approximate 2D and 3D Floor Plans from 3D Point Clouds. In Proceedings of the 14th International Conference on Computer Graphics Theory and Applications, Prague, Czech Republic, 9 December 2018; pp. 177–184.
13. Shukor, S.A.A.; Rushforth, E.J. Adapting Histogram for Automatic Noise Data Removal in Building Interior Point Cloud Data. *AIP Conf. Proc.* **2015**, *1660*, 070074. <https://doi.org/10.1063/1.4915792>.
14. Gankhuyag, U.; Han, J.-H. Automatic 2D Floorplan CAD Generation from 3D Point Clouds. *Appl. Sci.* **2020**, *10*, 2817. <https://doi.org/10.3390/app10082817>.
15. Pouraghdam, M.H.; Saadatseresht, M.; Rastiveis, H.; Abzal, A.; Hasanlou, M. Building Floor Plan Reconstruction from Slam-Based Point Cloud Using Ransac Algorithm. *ISPRS-Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *4218*, 483–488. <https://doi.org/10.5194/isprs-archives-XLII-4-W18-483-2019>.
16. Wang, Q.; Zhu, Z.; Chen, R.; Xia, W.; Yan, C. Building Floorplan Reconstruction Based on Integer Linear Programming. *Remote Sens.* **2022**, *14*, 4675. <https://doi.org/10.3390/rs14184675>.
17. Wang, C.; Ji, M.; Wang, J.; Wen, W.; Li, T.; Sun, Y. An Improved DBSCAN Method for LiDAR Data Segmentation with Automatic Eps Estimation. *Sensors* **2019**, *19*, 172. <https://doi.org/10.3390/s19010172>.
18. Czerniawski, T.; Nahangi, M.; Walbridge, S.; Haas, C. Automated Removal of Planar Clutter from 3D Point Clouds for Improving Industrial Object Recognition: 33rd International Symposium on Automation and Robotics in Construction, ISARC 2016. In Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC 2016), Auburn, AL, USA, 18–21 July 2016; pp. 357–365.
19. Chen, X.; Wu, H.; Lichti, D.; Han, X.; Ban, Y.; Li, P.; Deng, H. Extraction of Indoor Objects Based on the Exponential Function Density Clustering Model. *Inf. Sci.* **2022**, *607*, 1111–1135. <https://doi.org/10.1016/j.ins.2022.06.032>.
20. Wang, C.; Hou, S.; Wen, C.; Gong, Z.; Li, Q.; Sun, X.; Li, J. Semantic Line Framework-Based Indoor Building Modeling Using Backpack Laser Scanning Point Cloud. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 150–166. <https://doi.org/10.1016/j.isprsjprs.2018.03.025>.
21. Mura, C.; Mattausch, O.; Villanueva, A.J.; Gobbetti, E.; Pajarola, R. Robust Reconstruction of Interior Building Structures with Multiple Rooms under Clutter and Occlusions. In Proceedings of the 2013 International Conference on Computer-Aided Design and Computer Graphics, Guangzhou, China, 16–18 November 2013; pp. 52–59.
22. Yang, H.; Wu, H. Intelligent Classification of Point Clouds for Indoor Components Based on Dimensionality Reduction. In Proceedings of the 2020 5th International Conference on Computational Intelligence and Applications (ICCIA), Beijing, China, 19–21 June 2020; pp. 89–93.
23. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Xplore, 21–26 July 2017; pp. 652–660.
24. Martin, A. Fischler; Rober C. Bolles Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395. <https://doi.org/10.1145/358669.358692>.
25. Kim, M.; Lee, D. Automated Two-Dimensional Geometric Model Reconstruction from Point Cloud Data for Construction Quality Inspection and Maintenance. *Autom. Constr.* **2023**, *154*, 105024. <https://doi.org/10.1016/j.autcon.2023.105024>.
26. Martens, J.; Blankenbach, J. VOX2BIM+—A Fast and Robust Approach for Automated Indoor Point Cloud Segmentation and Building Model Generation. *PFG* **2023**, *91*, 273–294. <https://doi.org/10.1007/s41064-023-00243-1>.
27. Wu, H.; Yue, H.; Xu, Z.; Yang, H.; Liu, C.; Chen, L. Automatic Structural Mapping and Semantic Optimization from Indoor Point Clouds. *Autom. Constr.* **2021**, *124*, 103460. <https://doi.org/10.1016/j.autcon.2020.103460>.

28. Macher, H.; Landes, T.; Grussenmeyer, P. From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings. *Appl. Sci.* **2017**, *7*, 1030. <https://doi.org/10.3390/app7101030>.
29. Yao, T.; Yang, Q.; Zhang, R. Fast 3D Object Segmentation Using DBSCAN Clustering Based on Supervoxel. In Proceedings of the 2nd International Conference on Signal Image Processing and Communication (ICSIPC 2022), SPIE, Qingdao, China, 9 October 2022; Volume 12246, pp. 311–316.
30. Romero-Jarén, R.; Arranz, J.J. Automatic Segmentation and Classification of BIM Elements from Point Clouds. *Autom. Constr.* **2021**, *124*, 103576. <https://doi.org/10.1016/j.autcon.2021.103576>.
31. Kim, H.; Kim, C. 3D As-Built Modeling from Incomplete Point Clouds Using Connectivity Relations. *Autom. Constr.* **2021**, *130*, 103855. <https://doi.org/10.1016/j.autcon.2021.103855>.
32. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* **2019**, *38*, 1–12. <https://doi.org/10.1145/3326362>.
33. Perez-Perez, Y.; Golparvar-Fard, M.; El-Rayes, K. Scan2BIM-NET: Deep Learning Method for Segmentation of Point Clouds for Scan-to-BIM. *J. Constr. Eng. Manag.* **2021**, *147*, 04021107. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0002132](https://doi.org/10.1061/(ASCE)CO.1943-7862.0002132).
34. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the 2020 Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11108–11117.
35. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
36. Yin, C.; Wang, B.; Gan, V.J.L.; Wang, M.; Cheng, J.C.P. Automated Semantic Segmentation of Industrial Point Clouds Using ResPointNet++. *Autom. Constr.* **2021**, *130*, 103874. <https://doi.org/10.1016/j.autcon.2021.103874>.
37. Kim, J.; Chung, D.; Kim, Y.; Kim, H. Deep Learning-Based 3D Reconstruction of Scaffolds Using a Robot Dog. *Autom. Constr.* **2022**, *134*, 104092. <https://doi.org/10.1016/j.autcon.2021.104092>.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.