

Article

Effective Denoising Algorithms for Converting Indoor Blueprints Using a 3D Laser Scanner

Sehyeon Yoon [†], Sanghyun Choi [†] and Jhonghyun An ^{*†}

School of Computing, Gachon University, Seongnam 13120, Republic of Korea; wjwjwj2002@gachon.ac.kr (S.Y.); shchoi00@gachon.ac.kr (S.C.)

* Correspondence: jhonghyun@gachon.ac.kr

[†] These authors contributed equally to this work.

Abstract: This paper focuses on converting complex 3D maps created by LiDAR and SLAM technology into simple 2D maps to make them easier to understand. While 3D maps provide a lot of useful details for robots and computer programs, they can be difficult to read for humans who are used to flat maps. We developed a new system to clean up these 3D maps and convert them into intuitive and accurate 2D maps. The system uses three steps designed to correct different kinds of errors found in 3D LiDAR scan data: clustering-based denoising, height-based denoising, and Statistical Outlier Removal. In particular, height-based denoising is the method we propose in this paper, an algorithm that leaves only indoor structures such as walls. The paper proposes an algorithm that considers the entire range of the point cloud, rather than just the points near the ceiling, as is the case with existing methods, to make denoising more effective. This makes the final 2D map easy to understand and useful for building planning or emergency preparedness. Our main goal is to map the interior of buildings faster and more effectively, creating 2D drawings that reflect accurate and current information. We want to make it easier to use LiDAR and SLAM data in our daily work and increase productivity.

Keywords: denoising; LiDAR; 3D laser scanner; blueprint; digital twin; indoor; mapping



Citation: Yoon, S.; Choi, S.; An, J. Effective Denoising Algorithms for Converting Indoor Blueprints Using a 3D Laser Scanner. *Electronics* **2024**, *13*, 2275. <https://doi.org/10.3390/electronics13122275>

Academic Editors: Felipe Jimenez and Dah-Jye Lee

Received: 15 April 2024

Revised: 20 May 2024

Accepted: 4 June 2024

Published: 10 June 2024



Copyright: © 2024 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

In the evolving world of spatial analysis and indoor navigation, it is important to create accurate and easy-to-understand maps. In general, 3D maps contain more information than 2D maps. Robots use 3D information when perceiving space, and humans sometimes do too, because the third dimension can represent space in more detail.

However, when a building is remodeled or the structure of the building is changed, there are many differences between the existing drawings and the actual building. In addition, it is difficult to create a map that concentrates on a specific area, including the overall characteristics of the building. Furthermore, the drawings of existing buildings are not in a form that is convenient for users to use because they were created for the purpose of construction, so they are not suitable for the purpose of providing services using drawings. Traditionally, creating these maps has been performed by hand by human operators, which has the limitations of being error-prone and time-consuming.

As a solution to this problem, LiDAR simultaneous localization and mapping (SLAM) [1,2] technology has emerged, which enables the rapid collection of detailed spatial data to create 3D point cloud maps that provide a high level of accuracy and detail that cannot be matched by traditional methods.

However, these methods have several limitations. First, LiDAR sensors are inherently noisy. As the sensor navigates through an indoor space, it is affected by scattered reflections from highly reflective objects like windows and human trajectories. Second, it also includes moving objects that are not noise but are not typically included in the map. For example,

objects like chairs and potted plants are not noise, but they are easily moved, which is information that is not typically included in maps. The third is that LiDAR data are 3D. As mentioned earlier, the maps we are used to seeing are 2D. Three-dimensional maps sometimes provide more perceptual information, and 3D information is more useful for robots.

Many maps, such as emergency evacuation routes and smartphone maps, come in two-dimensional form, and people are used to two-dimensional maps. The intuitive and expressive nature of two-dimensional maps also makes them easier to modify and process than 3D.

To solve this problem, we want to process the relatively accurate three-dimensional maps obtained from LiDAR to create more intuitive and understandable 2D maps. By implementing this denoising system, we aim to significantly improve the indoor mapping process. This approach not only reduces the time and effort required to produce usable 2D maps, but also increases the accuracy and usability of the maps in a variety of applications. With this work, we are contributing to the broader field of digital mapping and indoor navigation, paving the way for new applications and efficiency in interaction methods by providing a practical solution that bridges the gap between advanced 3D mapping technology and the everyday needs of map users.

The rest of this paper is structured as follows: In Section 2, we discuss the related works in the field. Section 3 outlines the theoretical formulation of the proposed method. Moving on to Section 4, we describe the dataset configuration and the data logging system and present the experimental results. Finally, in Section 5, we provide a summary of the paper.

2. Related Work

In the domain of LiDAR data processing, numerous techniques have been developed to filter noise from 3D point clouds [3–6]. This section reviews the prominent methods and positions our approach within the existing landscape of research.

2.1. Statistical Outlier Removal (SOR)

One of the foundational methods in noise filtering for LiDAR data is Statistical Outlier Removal (SOR) [7]. This method relies on analyzing the local neighborhood of each point in the point cloud. By setting a threshold based on the mean and standard deviation of distances within a predefined number of neighboring points, SOR effectively identifies and removes points that deviate significantly from their surroundings, which are likely to be noise. While SOR is praised for its simplicity and effectiveness, it can sometimes lead to the removal of relevant features from the data, especially in environments with high point density variability.

2.2. Radius Outlier Removal (ROR)

Radius Outlier Removal (ROR) [8] is another widely used technique that considers the density of points within a specific radius. Points with fewer neighbors than a set threshold within this radius are considered outliers and are removed from the dataset. This method is particularly effective in environments where noise points are spatially isolated. However, the performance of ROR can degrade with increasing radius size, as it becomes computationally intensive and potentially overfilters the data, removing useful details along with the noise.

2.3. RANSAC (Random Sample Consensus)

RANSAC [9,10] is predominantly utilized for the robust estimation of model parameters in the presence of a large number of outliers. In the context of LiDAR data, RANSAC is often applied to identify and segregate ground points from non-ground points, facilitating a cleaner dataset for further processing. Although highly effective in many scenarios,

RANSAC's dependency on the assumption of a model can limit its applicability in complex environments where the ground surface may not conform to a simple geometric model.

2.4. Clustering

Clustering algorithms such as k-means [11], DBSCAN [12], and HDBSCAN [13] have also been applied to noise filtering in point clouds. These methods cluster points based on proximity and density, which can separate clusters of points that represent real objects from those that are likely to represent noise. Unlike k-means, DBSCAN and HDBSCAN do not require a predetermined number of clusters, making them more suitable for unpredictable 3D point cloud data characteristics. The main parameters of DBSCAN are the Epsilon (EPS) and the minimum number of points required to form a cluster (MinPts). DBSCAN suffers from the curse of dimensionality, making it less efficient on high dimensional data. To overcome this, the HDBSCAN algorithm has been introduced. HDBSCAN is an improved form of DBSCAN that performs clustering by automatically adjusting the EPS values. This provides more robust results for data with varying local densities. The hierarchical clustering approach automatically determines the optimal cluster size and effectively identifies clusters with different densities. It also has the advantage of having fewer parameters, making it easier to automate.

2.5. Combined Methods

Our research set out to explore the combination of these techniques to capitalize on their respective strengths while mitigating their weaknesses. We created an algorithm that combines the simple and efficient SOR described above with the HDBSCAN algorithm, which clusters by automatically adjusting EPS values, with optimal parameters. We also resolved points that were not removed by this combined algorithm with our own algorithm. HDBSCAN and SOR algorithms have the property of combining information that is close to each other. However, the points obtained by these algorithms contain information that is unnecessary to the drawing information we want to obtain, such as desks and vehicles. Therefore, we added our own algorithm that can remove these things, leaving only fixed structures such as walls and columns, to create a perfect blueprint image.

3. Proposed Method

The algorithm proposed in this work deals with the process of generating a two-dimensional map based on three-dimensional point cloud data acquired from LiDAR sensors. This process consists of four main steps: map construction, clustering-based denoising, self-developed height-based denoising, and application of the SOR algorithm, as shown in Figure 1. First, the three-dimensional point cloud data are processed to generate a three-dimensional map using the SLAM algorithm. Second, a clustering-based denoising algorithm is applied to remove unnecessary measurement data from the point cloud and noise identified through clustering at an early stage. Third, a height-based filtering algorithm developed in-house by the research team is used to identify and remove noise below a certain height. This unique approach aims to effectively remove noise that does not correspond to building structures, such as desks and cars, in particular. Finally, the SOR algorithm is applied to finally remove noise caused by diffuse reflections. This process further improves the precision of the map. After these steps, the processed data are finally projected into two dimensions to produce a high-quality two-dimensional map. The methodology proposed in this study highlights the importance of denoising and data cleaning in generating high-precision two-dimensional maps using LiDAR data, and shows how to maximize the accuracy and usability of the maps.

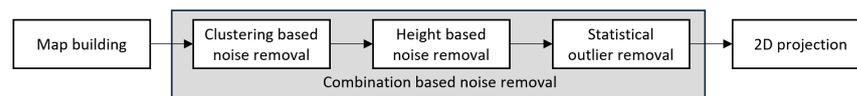


Figure 1. An illustration of the proposed method. In the first step, make a 3D point cloud map using the SLAM algorithm. Subsequently, remove outliers using a clustering algorithm. The denoised point cloud is fed into a height-based algorithm, an algorithm developed in-house, to remove unstructured information. Finally, remove the specular noise using the SOR algorithm.

3.1. Map Building

In this study, LeGO-LOAM (Lightweight and Ground-Optimized LiDAR Odometry And Mapping) [14] was selected for LiDAR-based positioning and mapping in various terrains. Figure 2 shows the system overview of LeGO-LOAM. LeGO-LOAM was chosen because it offers the following advantages over traditional LOAM [15] methodologies' lightweight and real-time processing: LeGO-LOAM significantly reduces the amount of computation, including the ground removal process. This allows real-time position estimation and mapping to be performed in a way that has much lighter weight compared with LOAM. This makes it especially feasible for systems with limited processing power. Robustness to unnecessary measurement data: LeGO-LOAM produces results that are more robust to noise and unnecessary data by filtering out unnecessary measurement data through segmentation. This process improves the accuracy of the data and increases computational efficiency in subsequent processing.

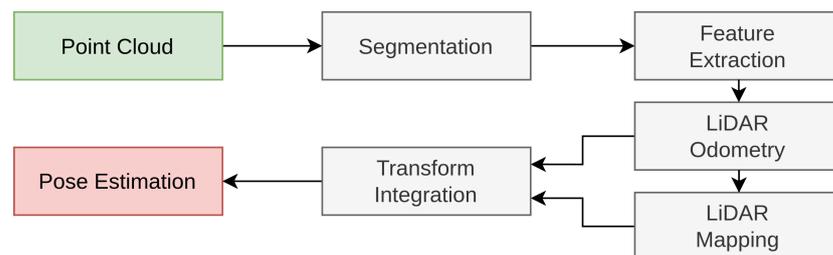


Figure 2. An illustration of the LeGO-LOAM system overview.

3.2. Clustering-Based Noise Removal

The denoising process in this study utilizes a clustering algorithm as the first step. Clustering contributes to the extraction of meaningful information by grouping data points based on similar characteristics or patterns [16,17]. This method effectively removes noise, such as the trajectory of a moving object, by identifying clusters that are not closely spaced. Clustering is an unsupervised learning method, which is ideal for discovering hidden patterns in data without predefined categories.

In this study, we used HDBSCAN for clustering. It is a representative algorithm of density-based clustering. Additionally, due to the nature of point cloud data with complex data, we decided that it is not appropriate to determine the number of clusters, and we chose HDBSCAN because it can reduce the number of parameters by one compared with DBSCAN. The HDBSCAN algorithm operates by initially transforming the space according to data density, followed by constructing a minimum spanning tree based on distance weights. It then builds a hierarchy of clusters from the connected components of the tree. This hierarchy is condensed by enforcing a minimum cluster size, from which stable clusters are subsequently extracted.

3.3. Height-Based Noise Removal

In this section, we introduce a specialized filtering algorithm designed for indoor environments, primarily focusing on static structures, such as walls and columns, while

excluding floors, ceilings, and dynamic objects. This approach is predicated on the observation that permanent architectural elements typically extend vertically from floor to ceiling, a feature not shared by transient or movable objects. By capitalizing on this distinction, the algorithm efficiently filters out irrelevant measurement data, a process succinctly summarized in Algorithm 1.

Algorithm 1 Height-Based Noise Reduction in Point Clouds

```

1: Input: Point cloud  $P$ 
2: Output: Denoised point cloud  $R$ 
3:
4:  $height \leftarrow \max(P_z) - \min(P_z)$  ▷ Calculate the height of the point cloud
5:  $voxel\_size \leftarrow 0.1$  ▷ Define the voxel size
6:  $threshold \leftarrow$  user-defined threshold ▷ Define threshold for filtering noise
7: Initialize voxel map  $V$  as an empty dictionary
8:
9: for each point  $p \in P$  do
10:    $index \leftarrow$  map  $p$  to voxel index based on  $(x, y)$  coordinates and  $voxel\_size$ 
11:   if  $index$  not in  $V$  then
12:      $V[index] \leftarrow$  empty list
13:   end if
14:   Append  $p$  to  $V[index]$ 
15: end for
16:
17: Initialize an empty list  $R$ 
18: for each  $index$  in  $V$  do
19:   if length of  $V[index] \geq threshold \times height$  then
20:     Append all points in  $V[index]$  to  $R$  ▷ Retain points in voxels that meet the threshold
21:   end if
22: end for
23:
24: return  $R$  ▷ Return the filtered, denoised point cloud

```

Our height-based filtering technique quantitatively assesses each point in the point cloud through the voxelization function:

$$V(p) = \left\lfloor \frac{p}{d} \right\rfloor \quad (1)$$

This Equation (1) systematically reduces the spatial resolution of the point cloud by mapping each point p into a voxel V , indexed by dividing the point's coordinates by a predetermined voxel size d , and applying the floor function. This transformation is instrumental for normalizing the distances between points within the same voxel, thereby simplifying subsequent analyses.

The efficacy of the noise reduction is evaluated through the following criterion:

$$\text{Output} = \{p \in P \mid p \text{ in } V[i] \text{ and } |V[i]| \geq \text{threshold} \times \text{height}\} \quad (2)$$

In Equation (2), $|V[i]|$ represents the number of points within voxel i , and $\text{threshold} \times \text{height}$ is a calculated metric based on the overall vertical extent of the point cloud, defined as the difference between its highest and lowest z-coordinates. Only voxels that meet or exceed this density threshold are retained in the final, denoised point cloud, ensuring that only significant structural elements are preserved.

The algorithm initializes by overlaying a voxel grid onto the point cloud, where each voxel is defined as a cube with a side length typically set to 0.1 units. This grid segments the three-dimensional space into discrete units, facilitating the localized examination of

point distributions. Points are assigned to voxels based solely on their x and y coordinates, preserving their z -coordinate for accurate vertical analysis.

The critical phase of the algorithm involves counting the unique z -values within each voxel and comparing this count against the predetermined threshold. Voxels that surpass this threshold are indicative of dense, structurally significant areas, while those below are considered noise and subsequently discarded.

Filtering indoor structures by height is a prevalent approach in point cloud data processing. Conventional methods [18–20] typically denoise a region by determining the distance of the point with the highest z -value from the ceiling. Additionally, the time complexity of this algorithm is $O(n)$, which makes it very fast and effective. However, this maximum z -value-based method can be problematic in the presence of objects such as ceiling fans, air conditioners, and beam projectors, as it requires precise parameter adjustments for accurate denoising. The method proposed in this study, on the other hand, improves on existing methods by evaluating the volume occupancy of the entire point cloud rather than focusing only on points close to the ceiling. This is accomplished through a voxel-based approach, which simplifies the process while effectively mitigating the limitations associated with traditional height-based filtering.

In comparison with other height-based denoising methods, our approach offers several advantages. An alternative method, grammar-based structure extraction [21,22], requires the prior definition of the geometry and dimensions of architectural elements such as walls, doors, and windows. This method segments only those structures that conform to the predefined grammatical rules and discards all other data points. However, adapting the grammar to accommodate new datasets is often a labor-intensive and impractical process for scenarios not covered by existing grammatical structures. Furthermore, grammar-based methods typically presuppose simplified, orthogonal ('Manhattan world') geometries and demand considerable effort to accurately represent streamlined wall surfaces, thus limiting their universal applicability.

Another way to remove noise is to use Principal Component Analysis (PCA) [23–25]. PCA is a classic dimensionality reduction method that transforms data by finding the directions of greatest variance in the data and transforming them into principal components. This allows you to reduce high-dimensional data to lower dimensions while preserving as much of the important information in the original data as possible. The method removes noise by separating the characteristics of the noisy data from the characteristics of the wall data. PCA has the advantage of being simple to use and is particularly effective when the data have a linear structure. It also reduces the dimensionality of the data, which makes it more computationally efficient. However, it might not work well with all datasets. For example, PCA techniques may not work well with point clouds that have curved or irregular surfaces. In these cases, PCA may not sufficiently reflect the complex shape of the data. Additionally, data measured as reflections from glass or mirrors are not well suited to filtering with PCA. Unlike randomly distributed noise, these reflections are indistinguishable from wall data, making PCA ineffective. Because PCA is a statistical method, its effectiveness can be limited if the distribution of the data does not follow a specific pattern. Because of these limitations, the method may not be suitable for general use on all datasets.

Furthermore, algorithms such as RANSAC [9,10,26] are utilized to align lines on walls by imposing angular constraints to identify planes that are approximately perpendicular to the xy -plane. While this method has the potential to deliver effective results, it is susceptible to noise and frequently mis-segments lines in complexly shaped structures. It also faces challenges in accurately extracting curved and columnar forms, generally restricting its utility to environments characterized by simple, orthogonal geometries.

In contrast, our proposed method can be applied to many datasets in general. The dataset we obtained for our experiments is noisier than typical refined datasets. It contains not only flat walls, but also curved walls and columns, noise points reflected from windows, outdoor points, people, vehicles, etc. The proposed method is specially designed to be less

sensitive to noise interference and does not require precise parameter tuning because it considers the characteristics of the entire dataset instead of focusing on some data points. The proposed height-based filtering is an improvement over existing methods, while keeping the time complexity to $O(n)$, making it very efficient. In conclusion, our proposed method is a highly efficient noise filtering that can be used in general for any dataset and noise, showing a clear improvement over existing methods.

3.4. Statistical Outlier Removal

The Statistical Outlier Removal (SOR) algorithm used in this study is commonly used for denoising three-dimensional point cloud data. The technique calculates the average distance and standard deviation of neighboring points around each point and removes points that exceed a certain threshold as unnecessary measurement data, i.e., outliers. A key assumption of the SOR algorithm is that the distribution of data points follows a Gaussian distribution (normal). Under this assumption, all data points that fall outside a certain distance (a threshold defined as a multiple of the standard deviation) from the mean are considered outliers and are removed. This process is performed using the mean distance and standard deviation, which reflect the global nature of the data.

4. Experiment

4.1. Experiment Environment

To denoise a 3D point cloud, it is important to first obtain a high-quality 3D point cloud map. To do this, we developed a custom sensor system to collect the necessary data. Since using other sensor data would have been an additional cost and process, we utilized only point cloud data to build a detailed 3D point cloud map, which is the basis for the denoising procedure.

4.1.1. Sensor Setting

The dataset used in this study was created by scanning the interior space of the building using 3D LiDAR technology. The dataset also includes a variety of noise generated during the scanning process. By creating our own maps rather than using open-source data, you have the benefit of a deeper understanding of each space and ground truth.

Our experimental driving platform was equipped with a 3D laser scanner and Ouster OS-1 64-channel sensor (Ouster Inc., San Francisco, CA, USA). We secured the LiDAR and power bank to the cart, as shown in Figure 3. To use loop closing to match the start and end points of the scan, we used a cone to mark the start point.



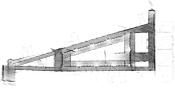
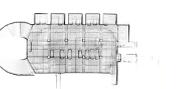
Figure 3. Experimental platform with a 64-channel 3D laser scanner.

4.1.2. Map Building

The maps produced cover a wide range of indoor environments, including building interiors, subway stations, and underground car parks, and contain detailed information about places with complex and varied structural elements, such as glass walls and rounded columns. These characteristics contribute to the versatility and complexity of the maps.

The dataset is presented in Table 1, including distinct environmental attributes pertinent to each location. The Gachon University station area is distinguished by a sequence of architectural columns, offering a structured yet complex navigational landscape. The Gachon Hall is characterized by diverse terrain features, including an array of indoor fixtures such as signboards, sofas, and desks, complemented by elevated ceilings that add to the spatial variability. The AI Building presents a challenging environment for optical sensors due to the prevalence of glass surfaces, which induce significant diffuse reflection, complicating the clarity of sensor readings. The subterranean level of the Vision Tower, functioning as a parking facility, encompasses dynamic elements such as numerous vehicles and pedestrian traffic, introducing variability and movement that are critical for evaluating the robustness of traversable region detection algorithms.

Table 1. Datasets created through map building.

Location	Gachon Uni. Station	Gachon Hall	AI Building 7F	Vision Tower B3F
Point Cloud Image				
Dimensions (mm)	35,800 × 150,200	43,800 × 108,200	42,300 × 96,700	29,000 × 65,500

4.2. Noise Removal

When denoising, it is important to choose the right algorithm for the nature of the noise you want to remove. This is to remove areas within the 3D point cloud where there is insufficient accumulation of points, for example, moving objects or objects detected beyond windows.

4.2.1. Clustering-Based Noise Removal

In Figure 4, we demonstrate the application of the silhouette score [27] to optimize clustering parameters, specifically the *minpts* (minimum points) parameter of HDBSCAN. The *minpts* parameter defines the minimum number of points required to form a dense region, crucial for identifying core clusters. The silhouette score serves as a crucial metric for evaluating the quality of clustering arrangements within the context of 3D point cloud denoising. By assessing how closely each data point relates to others within its cluster compared with neighboring clusters, the silhouette score enables the identification of an optimal *minpts* value that maximizes clustering effectiveness.

It is important to highlight that while the use of the silhouette score in our methodology is not a novel research contribution, its application within the context of 3D point cloud denoising underscores its relevance as a fundamental tool for evaluating clustering performance. By leveraging this established statistical measure, we ensure an approach to parameter optimization, thus contributing to the generation of high-quality clustering results.

This experiment targeted a range of *minpts* values from 5 to 1000, seeking to identify the value that maximizes the overall silhouette score. Due to computational constraints, a sampling strategy was adopted where every tenth point was selected to compute the silhouette score. This approach helped to strike a balance between computational efficiency and accuracy of silhouette score estimation and contributed to the identification of optimal *minpts* values that promote high-quality clustering results in the context of 3D point cloud denoising.

As illustrated in Figure 4, optimal performance was observed within the range of 400 to 500 for the *minpts* parameter. Subsequent experimentation, focused specifically on this narrowed interval, revealed that the silhouette score reached its apex at a *minpts* value of 443. This outcome underscores the significance of fine-tuning the *minpts* parameter to enhance the clustering algorithm's efficacy.

An image comparison of the original PCD and the PCD clustered with the best parameter values from HDBSCAN found through silhouette scoring, as shown in Figure 5.

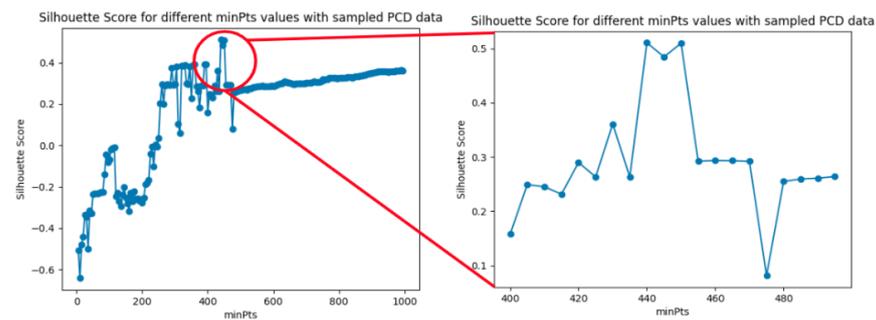


Figure 4. Silhouette score to find optimal HDBSCAN parameter.

A substantial mitigation of noise artifacts has been accomplished, notably the erratic signals attributable to pedestrian movement and the specular noise emanating from the glass windows. Despite these advancements, residual noise within the dataset persists and warrants further noise removal to achieve an even cleaner PCD.

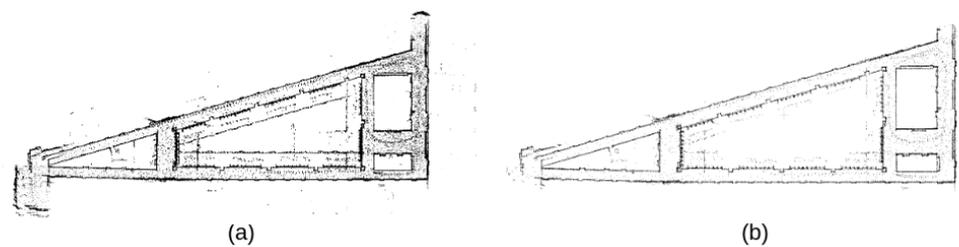


Figure 5. (a) Original PCD and (b) PCD clustered by an HDBSCAN optimal parameter.

4.2.2. Height-Based Noise Removal

In the preprocessing stage, the denoised dataset using HDBSCAN is used as input data. The point cloud data, initially dense, are transformed into a voxel representation to enhance computational efficiency and reduce storage overhead. This conversion leverages the inherent benefits of voxelization in handling extensive point cloud data. Subsequently, we compute the aggregate height of the processed point cloud. A proportion of this total height is designated as a threshold to discern significant structural elements from negligible entities. We then conduct a voxel-based analysis, focusing on voxels sharing identical x and y coordinates but varying in z -axis values. The occupancy count of these vertical voxel stacks is tallied; voxel stacks falling below the established height threshold are deemed non-essential and subsequently purged, eliminating all associated points.

To determine the optimal threshold, a heuristic approach was employed, identifying the most effective value as one where one-third of all voxels possessed valid data. This criterion facilitated the isolation of pertinent information while effectively reducing noise in the dataset. As demonstrated in Figure 6, the application of the HDBSCAN algorithm alone was insufficient for the effective removal of noise originating from ceilings and vehicles. However, when integrated with our proposed height-based algorithm, the noise was effectively mitigated, resulting in a dataset where only essential structures such as pillars, parking spots, and walls remained discernible. This enhancement underscores the efficacy of our height-based approach in isolating key architectural elements.

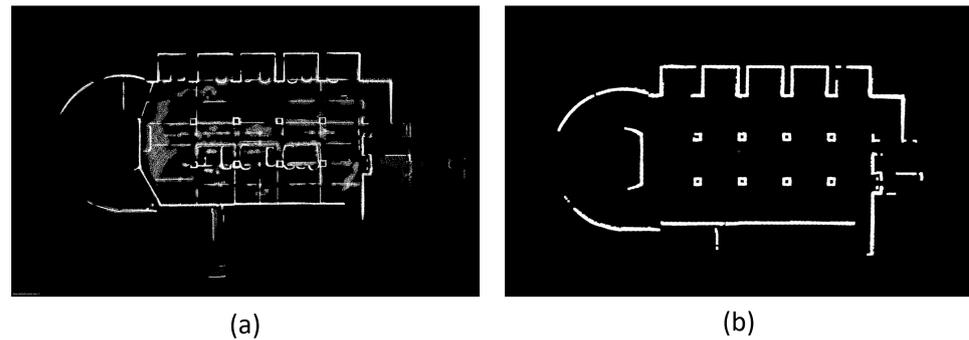


Figure 6. (a) PCD clustered by HDBSCAN and (b) PCD noise removed by HDBSCAN, height-based algorithm.

4.2.3. Statistical-Based Noise Removal

Despite dual denoising efforts, instances of partial noise contamination persisted within the dataset, necessitating the implementation of the Statistical Outlier Removal (SOR) algorithm for outlier mitigation. This algorithm operates on the principle of spatial consistency, assessing the local neighborhood of each point across the dataset, indexed from 1 through N . For a given point n , the SOR algorithm computes the mean and variance of the distances to its k -nearest neighbors. Points exhibiting a variance exceeding a pre-defined threshold are classified as outliers and excised from the dataset.

The SOR procedure is contingent upon two critical parameters: the count of neighboring points (k) and the threshold for the standard deviation variance. Extensive experimentation was conducted to ascertain the optimal parameter pairing. Our findings indicate that a reduction in the neighbor count adversely affected the results, whereas an increment in neighbors contributed to improved noise reduction. Through a heuristic approach, we established that a neighbor count of approximately 100 was universally optimal across four-point cloud datasets (PCDs). The standard deviation variance parameter, conversely, exhibited negligible influence on the outcome within the context of our experimentation.

The results are shown in Figure 7. Previous combinations of algorithms failed to eliminate noise reflected from glass windows. However, the application of the Statistical Outlier Removal (SOR) algorithm substantially enhanced noise mitigation, effectively removing nearly all such disturbances. This included not only reflections but also minor noise and trajectories, which were all thoroughly eliminated.

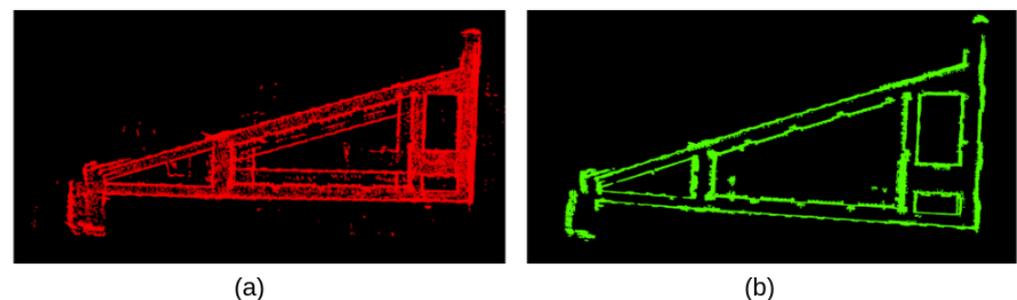


Figure 7. (a) Original PCD and (b) PCD noise removed by SOR optimal parameter.

4.2.4. Combination-Based Noise Removal

This combination-based noise filtering step combines the three steps described earlier into one. The settings of the parameters that remove noise were refined through empirical testing. Initially, clustering-based methods filtered out low-density noise, but struggled with noise on some floor and ceiling surfaces. Then, height-based algorithms removed noise such as pedestrian paths and low objects, but were vulnerable to vertical noise from

reflections. Statistical Outlier Removal (SOR) algorithms were excellent at removing dense noise near walls, but were less effective for high-density objects such as vehicles.

Therefore, we employed a composite strategy combining clustering-based, height-based, and SOR algorithms to optimize noise reduction across varying conditions. This integrated approach resulted in more effective noise filtration, significantly enhancing the clarity of the point cloud data for detailed analysis.

The effectiveness of our proposed methodologies is illustrated in Figure 8. For dataset (a), the primary challenge involved eliminating noise reflected through windows, which proved difficult with a height-based algorithm alone. However, as depicted in (b), the integration of multiple algorithms facilitated the clean removal of this noise. In (c), the presence of glass, a couch with a sign, and various other interior features posed additional complexities, yet the amalgamated approach effectively eliminated these elements, as evidenced in (d). In scenario (e), where rows of pillars needed to be preserved while removing adjacent noise, the traditional plane segmentation algorithm [28] inadvertently erased these structural features shown in Figure 9. In contrast, our refined algorithm successfully maintained the pillars, enhancing the dataset's fidelity, as shown in (e). Furthermore, in (g), the dataset included dynamic noise sources such as moving and parked vehicles and pedestrians. Our algorithm outperformed standard methods like SOR and HDBSCAN in removing these transient elements, with the results displayed in (h) reaffirming the robustness and precision of our approach.

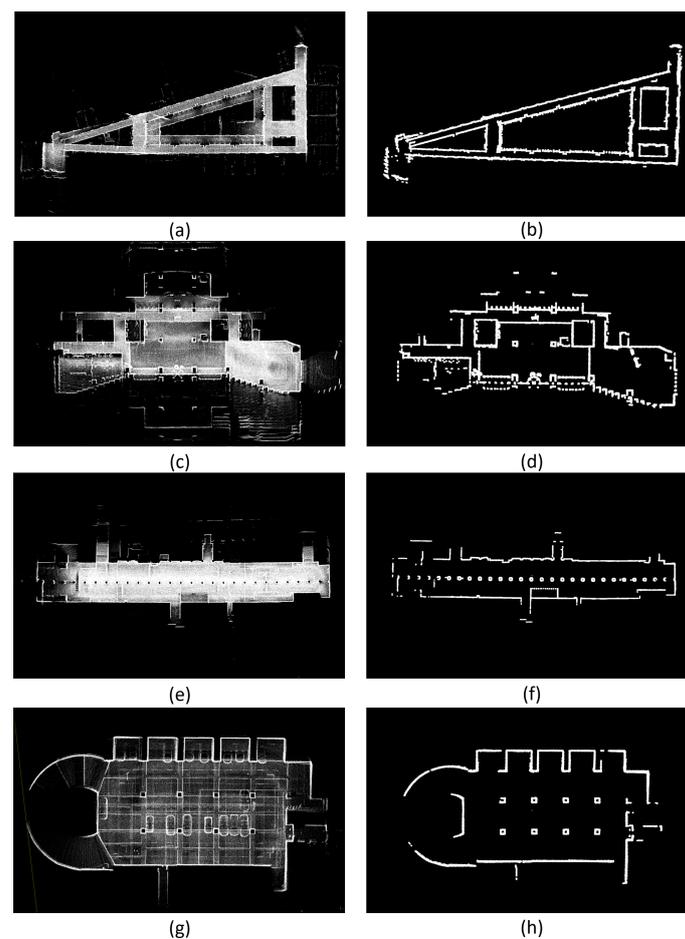


Figure 8. (a) Original AI 7f PCD, (b) final AI 7f PCD noise removed, (c) original Gachon hall PCD, (d) final Gachon hall PCD noised removed, (e) original Gachon uni station PCD, (f) final Gachon uni station PCD noised removed, (g) original Vision Tower PCD, and (h) final Vision Tower PCD noised removed.

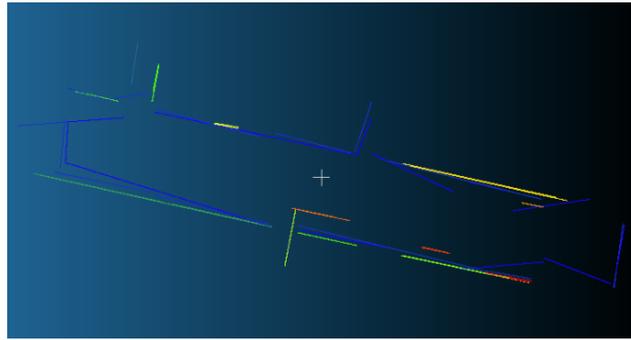


Figure 9. Gachon station plane segmentation result.

4.3. Evaluation and Results

4.3.1. Pixel-Based Evaluation

This metric assumes that the output is rendered in a two-dimensional format, specifically as an image. If a pixel accurately reflects the expected condition, it is assigned a value of '1' (full); otherwise, it is assigned a value of '0' (empty). Utilizing the design drawings of a building as the GT, the match between the ground truth and the image is evaluated based on exact pixel matching. A pixel is considered a match if both the pixel in the GT and the corresponding pixel in the image are '1' or both are '0'. If a pixel is a '1' in the GT and a '0' in the image, or vice versa, it is a mismatch, and that pixel is not included in the accuracy measurement. This method provides a simple pixel-by-pixel assessment of the fidelity of the image to the expected result, as follows:

$$\text{Accuracy} = \frac{\sum_{i=1}^n \text{match}(p_i, gt_i)}{N} \quad (3)$$

where p_i represents the predicted value of the i -th pixel, gt_i is the ground truth value of the i -th pixel, and $\text{match}(p_i, gt_i)$ is a function that returns 1 if $p_i = gt_i$ and 0 otherwise. Here, N is the total number of pixels in the image.

Table 2 shows the Intersection over Union (IoU) for the AI Buildings dataset using Equation (3). The table presents a summary of the IoU metric evaluated through a pixel-based approach AI 7F dataset. At each step of our algorithmic pipeline—encompassing clustering, height-based filtering, and Statistical Outlier Removal (SOR)—we preserved images for subsequent analysis. By comparing the pixel correspondence with the ground truth, we quantified the degree of match. What we wanted to know from this experiment was a quantitative assessment of whether the combination of the SOR, HDBSCAN, and height-based algorithms was more effective than any of the algorithms alone or in some combination.

Table 2. Quantitative results for pixel IoU.

	Combination 1	Combination 2	Combination 3
Clustering-Based Noise Removal [13]	○	○	○
Statistical-Based Noise Removal [7]	×	×	○
Proposed Height-Based Noise Removal	×	○	○
Result [%]	72	80	89

In the first combination, we used only cluster-based denoising to remove noise, and the result was about 72% matching with the GT image. This was due to the fact that only very basic noise was removed, and all the reflections of glass and easily moving objects were reflected. In the second combination, we used the cluster-based denoising technique and the height-based denoising technique we developed to remove the noise. The result was about 80% matching with the GT image, which is higher than before. We can conclude that the addition of the height-based algorithm removes the information of moving objects, which is the reason for the increase. However, there is still a strong noise, such as the reflective wall, which is not the highest. The third combination is the system we propose in

our paper, which combines cluster-based denoising, height-based denoising, and SOR. This combination has a matching rate of 89% with GT images, which is higher than those of the previous two combinations, because it removes the noise of diffuse reflections and moving objects, which were the cause of poor performance in the previous combinations. This result confirms that the combination of algorithms we used in our paper is the most effective.

4.3.2. Density-Based Evaluation

We introduce a methodology designed to quantitatively evaluate the effectiveness of the denoising process implemented in this study. The calculation of this denoising metric is encapsulated by the following simplified formula:

$$S = \frac{1}{|P|} \sum_{i=1}^{|P|} \left(\frac{1}{|I_i|} \sum_{j \in I_i, j \neq i} \exp(-\psi \|P_i - P_j\|^2) \right) \quad (4)$$

In a refined point map, unnecessary elements are removed so that fixed structures, such as walls or columns, are visible. If there is noise, the point density will appear lower. This makes it easy to tell the difference between well-denoised and poorly denoised point sets by looking at the point density per unit area. A well-denoised point set has a high point density because only information about the main structures remains, whereas a poorly denoised set has a low point density because the point set includes extraneous noise that is scattered throughout. The density-based evaluation method we invented was devised using features from a well-denoised set of points. Before calculating density, project the point cloud data onto the z axis to distinguish between points close to the center and those on the edges. The quality of the denoised point cloud is denoted by the score S , and is quantified using Equation (4).

The score is retrieved for every point P in the dataset, where P_i is the i_{th} point. To obtain S , a density-based evaluation method, a unit region of the radius r is required. The set of points in the unit region centered at P_i is called I_i . If the number of points in I_i is insufficient, the set I_i is set to the nearest n points. Subsequently, the distances between P_i and every other point P_j in I_i are computed. The weight function $\exp(-\psi \|P_i - P_j\|^2)$ is a y -axis symmetric, ranges from 0 to 1, and has the shape of a Gaussian distribution. The weighting function stretches depending on the value of ψ , which is similar to the standard deviation in a normal distribution. Despite equivalent point counts within a given radius, the spatial distribution—whether clustered centrally or dispersed—may vary significantly. Thus, a weighting function is applied to points nearer to the center to reflect their contribution more accurately, and the averaged results provide a measure for assessing the overall denoising efficacy of the map. Since the maximum value of S is fixed at 1 and the minimum at 0, it can be used to quantitatively compare the denoising of the same dataset. Using this, we performed a quantitative evaluation on one of our datasets, AI 7f, and the results are in Table 3.

Table 3. Quantitative results based on density and distance.

	Combination 1	Combination 2	Combination 3
Clustering-Based Noise Removal [13]	○	○	○
Statistical-Based Noise Removal [7]	×	×	○
Proposed Height-Based Noise Removal	×	○	○
Result [%]	65.4	68.2	69.6

We can account for the spatial distribution by giving a higher weight to points closer to the center and a lower weight to points farther away from the center. This will provide a more accurate measure of the overall denoising effectiveness of the map. The results are shown below. The first combination only applied the clustering algorithm, which means that the basic noise was removed, not the diffuse reflection noise and moving object noise. Since there is a lot of noise left, the points are scattered in the space, and we can see that the density-based evaluation value of our proposed algorithm is 65.4%, which is lower

than the other two combinations. The second combination used the clustering algorithm and our proposed algorithm, the height-based denoising algorithm. The density-based evaluation value of PCD using this algorithm is 68.2%, which shows an increase in density between points compared with the first combination. This is due to the removal of noise from moving objects such as cars, which are easy to move, as explained earlier. The third combination had the highest density among the three combinations, with a density of 69.6%. This is because the noise from diffuse reflections and moving objects, which caused the performance to decrease in the previous combination, was removed. The density-based performance evaluation also confirmed that the combination of algorithms we used in our paper was the most effective.

5. Conclusions

In conclusion, this paper has successfully developed a novel denoising system tailored for the efficient transformation of intricate 3D LiDAR point clouds into user-friendly 2D maps. Our comprehensive system integrates the LeGO-LOAM algorithm for map construction with a multi-stage noise filtration process, including clustering-based denoising, height-based filtering, and the application of the Statistical Outlier Removal (SOR) algorithm. These methodologies collectively enhance the precision of 2D cartographic outputs.

The study's findings underscore the pivotal role of noise reduction in the interpretation of LiDAR-generated spatial data, thereby facilitating its broader application in fields such as robotics, architecture, and emergency management. Looking ahead, the focus will be on augmenting the efficiency of the denoising process, expanding the integration with diverse sensing technologies, and thoroughly evaluating the system's practicality in real-world scenarios. Anticipated future developments are aimed at advancing the commercial viability of LiDAR-based 2D mapping, broadening its accessibility and utility across various industries.

Author Contributions: Methodology, S.Y. and S.C.; Software, S.Y. and S.C.; Investigation, J.A.; Writing—original draft, S.Y. and S.C.; Writing—review & editing, J.A.; Visualization, S.C.; Supervision, J.A.; Project administration, J.A.; Funding acquisition, J.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2022-00165871) and supported by the Gachon University research fund (202400470001) and LIG NEX1.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cole, D.M.; Newman, P.M. Using laser range data for 3D SLAM in outdoor environments. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, Orlando, FL, USA, 15–19 May 2006; pp. 1556–1563.
2. Huang, L. Review on LiDAR-based SLAM techniques. In Proceedings of the 2021 International Conference on Signal Processing and Machine Learning (CONF-SPML), Online, 11–16 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 163–168.
3. Gao, R.; Li, M.; Yang, S.J.; Cho, K. Reflective noise filtering of large-scale point cloud using transformer. *Remote Sens.* **2022**, *14*, 577. [[CrossRef](#)]
4. Gao, R.; Park, J.; Hu, X.; Yang, S.; Cho, K. Reflective noise filtering of large-scale point cloud using multi-position LiDAR sensing data. *Remote Sens.* **2021**, *13*, 3058. [[CrossRef](#)]
5. Charron, N.; Phillips, S.; Waslander, S.L. De-noising of lidar point clouds corrupted by snowfall. In Proceedings of the 2018 15th Conference on Computer and Robot Vision (CRV), Toronto, ON, Canada, 9–11 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 254–261.
6. Fang, H.T.; Huang, D.S. Noise reduction in lidar signal based on discrete wavelet transform. *Opt. Commun.* **2004**, *233*, 67–76. [[CrossRef](#)]
7. Balta, H.; Velagic, J.; Bosschaerts, W.; De Cubber, G.; Siciliano, B. Fast Statistical Outlier Removal Based Method for Large 3D Point Clouds of Outdoor Environments. *IFAC-PapersOnLine* **2018**, *51*, 348–353. [[CrossRef](#)]
8. Duan, Y.; Yang, C.; Li, H. Low-complexity adaptive radius outlier removal filter based on PCA for lidar point cloud denoising. *Appl. Opt.* **2021**, *60*, E1–E7. [[CrossRef](#)] [[PubMed](#)]

9. Raguram, R.; Chum, O.; Pollefeys, M.; Matas, J.; Frahm, J.M. USAC: A Universal Framework for Random Sample Consensus. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2022–2038. [[CrossRef](#)] [[PubMed](#)]
10. Schnabel, R.; Wahl, R.; Klein, R. Efficient RANSAC for point-cloud shape detection. *Comput. Graph. Forum* **2007**, *26*, 214–226. [[CrossRef](#)]
11. Krishna, K.; Narasimha Murty, M. Genetic K-means algorithm. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **1999**, *29*, 433–439. [[CrossRef](#)] [[PubMed](#)]
12. Schubert, E.; Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Trans. Database Syst.* **2017**, *42*, 1–21. [[CrossRef](#)]
13. McInnes, L.; Healy, J.; Astels, S. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* **2017**, *2*, 205. [[CrossRef](#)]
14. Shan, T.; Englot, B. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765. [[CrossRef](#)]
15. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. In Proceedings of the Robotics: Science and Systems, Berkeley, CA, USA, 12–16 July 2014; Volume 2, pp. 1–9.
16. Xu, R.; Wunsch, D. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678. [[CrossRef](#)] [[PubMed](#)]
17. Saxena, A.; Prasad, M.; Gupta, A.; Bharill, N.; Patel, O.P.; Tiwari, A.; Er, M.J.; Ding, W.; Lin, C.T. A review of clustering techniques and developments. *Neurocomputing* **2017**, *267*, 664–681. [[CrossRef](#)]
18. Jung, J.; Stachniss, C.; Kim, C. Automatic Room Segmentation of 3D Laser Data Using Morphological Processing. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 206. [[CrossRef](#)]
19. 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. [[CrossRef](#)]
20. Hong, S.; Jung, J.; Kim, S.; Cho, H.; Lee, J.; Heo, J. Semi-automated approach to indoor mapping for 3D as-built building information modeling. *Comput. Environ. Urban Syst.* **2015**, *51*, 34–46. [[CrossRef](#)]
21. Cai, Y.; Fan, L. An Efficient Approach to Automatic Construction of 3D Watertight Geometry of Buildings Using Point Clouds. *Remote Sens.* **2021**, *13*, 1947. [[CrossRef](#)]
22. Bassier, M.; Vergauwen, M. Topology Reconstruction of BIM Wall Objects from Point Cloud Data. *Remote Sens.* **2020**, *12*, 1800. [[CrossRef](#)]
23. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [[CrossRef](#)]
24. Cheng, D.; Zhao, D.; Zhang, J.; Wei, C.; Tian, D. PCA-Based Denoising Algorithm for Outdoor Lidar Point Cloud Data. *Sensors* **2021**, *21*, 3703. [[CrossRef](#)]
25. Cheng, D.; Zhang, J.; Zhao, D.; Chen, J.; Tian, D. Automatic Extraction of Indoor Structural Information from Point Clouds. *Remote Sens.* **2021**, *13*, 4930. [[CrossRef](#)]
26. Han, J.; Rong, M.; Jiang, H.; Liu, H.; Shen, S. Vectorized indoor surface reconstruction from 3D point cloud with multistep 2D optimization. *ISPRS J. Photogramm. Remote Sens.* **2021**, *177*, 57–74. [[CrossRef](#)]
27. Shahapure, K.R.; Nicholas, C. Cluster quality analysis using silhouette score. In Proceedings of the 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), Sydney, Australia, 6–9 October 2020; pp. 747–748.
28. Lu, X.; Yao, J.; Tu, J.; Li, K.; Li, L.; Liu, Y. Pairwise linkage for point cloud segmentation. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 201–208. [[CrossRef](#)]

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.