Black Sea Journal of Engineering and Science 3(4): 128-137 (2020)
doi: 10.34248/bsengineering.735705

Black Sea Journal of Engineering and Science
Open Access Journal
e-ISSN: 2619 – 8991

BSJournals

Research Article
Volume 3 - Issue 4: 128-137 / October 2020

A COMPARATIVE STUDY FOR INDOOR PLANAR SURFACE SEGMENTATION VIA 3D LASER POINT CLOUD DATA

Eyüp Eymen ERUYAR[1], Metehan YILMAZ[1], Berat YILMAZ[1], Onur AKBULUT[1], Kaya TURGUT[1], Burak KALECİ[1]*

[1]Eskisehir Osmangazi University, Faculty of Engineering, 26000, Eskisehir, Turkey

***Received:** May 11, 2020; **Accepted:** July 20, 2020; **Published:** October 01, 2020*

Abstract

In recent years, point cloud data generated with RGB-D cameras, 3D lasers, and 3D LiDARs have been employed frequently in robotic applications. In indoor environments, RGB-D cameras, which have short-range and can only describe the vicinity of the robots, generally are opted due to their low cost. On the other hand, 3D lasers and LiDARs can capture long-range measurements and generally are used in outdoor applications. In this study, we deal with the segmentation of indoor planar surfaces such as wall, floor, and ceiling via point cloud data. The segmentation methods, which are situated in Point Cloud Library (PCL) were executed with 3D laser point cloud data. The experiments were conducted to evaluate the performance of these methods with the publicly available Fukuoka indoor laser dataset, which has point clouds with different noise levels. The test results were compared in terms of segmentation accuracy and the time elapsed for segmentation. Besides, the general characteristics of each method were discussed. In this way, we revealed the positive and negative aspects of these methods for researchers that plan to apply them to 3D laser point cloud data.

Keywords: 3D laser, Segmentation, Planar surfaces, Indoor, Fukuoka indoor dataset, PCL

***Corresponding author:** Eskisehir Osmangazi University, Faculty of Engineering, 26000, Eskisehir, Turkey
E mail: burakaleci@gmail.com (B. KALECİ)

Eyüp Eymen ERUYAR	https://orcid.org/0000-0003-2775-1798
Metehan YILMAZ	https://orcid.org/0000-0002-3083-2460
Berat YILMAZ	https://orcid.org/0000-0003-0214-2662
Onur AKBULUT	https://orcid.org/0000-0003-2918-2832
Kaya TURGUT	https://orcid.org/0000-0003-3345-9339
Burak KALECİ	https://orcid.org/0000-0002-2001-3381

Cite as: Eruyar EE, Yılmaz M, Yılmaz B, Akbulut O, Turgut K, Kaleci B. 2020. A comparative study for indoor planar surface segmentation via 3D laser point cloud data. BSJ Eng Sci, 3(4): 128-137.

1. Introduction

In past years, the researchers generally exploited range data acquired from ultrasonic sensors and 2D laser scanners and/or visual data captured with cameras to provide information to robots for tasks they are expected to perform. Although the range data is generally accurate, it can only present planar information at the height where the sensor is located. On the other hand, the visual data yields the color information about the scene. However, its reliability heavily relies on the lighting conditions of the environment. Besides, the range and/or visual data may not describe the vicinity of the robot sufficiently due to their natural characteristics. For this reason, in recent years, point cloud data has been employed frequently in robotic applications because of its powerful ability to describe the shape, size, position, and orientation of objects (Grilli et al., 2017). 3D point cloud data can be acquired with various sensors such as RGB-D cameras, 3D lasers, and 3D LiDARs. The most significant advantage of the RGB-D cameras is their low cost compared with the 3D lasers and 3D LiDARs. The point cloud data generated with RGB-D cameras has the same structure with RGB images since it is constructed by regarding camera's position and depth information in each pixel. RGB-D cameras are frequently preferred in robotic applications due to these advantages. However, the foremost disadvantage of these cameras is their short-range. On the other hand, 3D lasers and LiDARs generally can capture long-range measurements when they are compared with RGB-D cameras. For this reason, these sensors are often used in outdoor applications such as autonomous vehicles and urban mapping (Xie et al., 2019).

The segmentation of point clouds can be defined as the process of separating the points that have the same characteristics into homogeneous parts (Nguyen and Le, 2013; Grilli et al., 2017). Point cloud segmentation is an essential preprocessing or post-processing step for activities such as semantic information extraction, object recognition, place classification, and human tracking. Thus, it is a very active research area in a wide range of applications from outdoor to indoor. The point cloud segmentation was used in different outdoor applications such as building roof plane segmentation with the Airborne Laser Scanning (ALS) point cloud data (Tarsha-Kurdi et al., 2007; Xu et al., 2016), building facade segmentation with the Terrestrial Laser Scanning (TLS) point cloud data (Ning et al., 2009; Vo et al., 2015), forest monitoring (Morsdorf et al., 2004; Ferraz et al., 2010), autonomous vehicles (Himmelsbach et al., 2010; Zermas et al., 2017), and 3D silhouette extraction of a street (Mutlu et al., 2014). The point cloud segmentation was also applied in indoor applications such as Building Information Modeling (BIM) (Anil et al., 2013; Qu et al., 2014), robotic applications (Rusu et al., 2008; Koppula et al., 2011), and object detection (Mattausch et al., 2014; Cadena and Košecka, 2015). In the reviews, Nguyen and Le (2013) and Grilli et al. (2017) classified the point cloud segmentation methods into five categories: edge-based, region-based, model-based, clustering-based, and graph-based. Edge-based methods first aim to detect the edges where point characteristics are changed. Then, they group the points between the edges. The edge-based methods are generally weak against noise and they may produce low segmentation accuracy although they provide fast segmentation. Region-based methods exploit the neighbor relation to segment the point clouds. Despite their long segmentation time, the region-based methods present high segmentation accuracy. Model-based methods try to fit a model for each segment. They are fast and robust against noise. The disadvantage of these methods is to produce wrong models when the point cloud has nearly coplanar surfaces and the points in the cloud have uneven density. Clustering-based methods are appropriate for irregular object segmentation since they do not depend on a specific model. Also, they can combine different criteria for segmentation. The main disadvantage of these methods is that they have high computational complexity. The graph-based methods can be considered as the subcategory of the clustering-based methods.

In this study, we cope with the segmentation of indoor planar surfaces such as wall, floor, and ceiling. Although each segmentation category has advantages and disadvantages, most of the existing studies that address the segmentation of planar surfaces in indoor environments applied the well-known model-based RANSAC and region growing methods. Besides, these studies generally used the point cloud data captured with RGB-D cameras. On the other hand, 3D lasers and LiDARs can offer better performance in indoor robotic applications, especially for BIM, mapping, and semantic information extraction, due to their long-range measurements. In this study, we aim to examine the performance of the segmentation methods that are situated in PCL (Rusu and Cousins, 2011) with 3D laser point cloud data for planar surfaces in the indoor environment. In this way, we investigate the potential of edge-based, clustering-based, and graph-based (supervoxels) methods for planar surfaces with 3D laser point cloud data. To achieve this, the experiments were conducted with Fukuoka indoor laser dataset containing point cloud data, which has different noise levels (Martinez Mozos et al., 2019). After some preprocessing operations were applied to the dataset, point-wise labeling was performed with the RViz cloud annotation tool (Monica et al., 2017). The test results were compared in terms of segmentation accuracy and the time elapsed for segmentation. Besides, the general characteristics of each method were discussed.

The rest of the paper is organized as follows: In section 2, a detailed explanation of the segmentation methods is given. We present the experimental setup and results in section 3 and we conclude with section 4.

2. Material and Methods

The segmentation methods in the PCL are listed as follows: plane model segmentation, cylinder model segmentation, Euclidean cluster extraction, region growing segmentation, color-based region growing segmentation, min-cut based segmentation, conditional Euclidean clustering, the difference of normals based segmentation, supervoxel clustering and progressive morphological filtering (PCL-S, 2020).

In this study, we focus on the segmentation of structural planar surfaces such as wall, floor, and ceiling and we intend to use only range data for this purpose. Therefore, the cylinder model segmentation method (Rabbani et al., 2006) that detects cylinders and spheres is out of the scope of this paper. The Euclidean cluster extraction approach (PCL-ECE, 2020) successfully segments the objects since it separates the point cloud into smaller parts. However, this approach is not appropriate for structural planar surfaces. The color-based region growing segmentation approach (PCL-CBRG, 2020) uses color instead of curvature and normal features while segmenting the point clouds. Hence, this method is also out of the scope of this paper. Since the min-cut based segmentation approach (PCL-MCBS, 2020) merely is designed for object segmentation, we did not examine the method. Lastly, progressive morphological filter segmentation (PCL-PMFS, 2020) is also out of the scope of this paper because it aims to segment the roof of the buildings with LiDAR sensors mounted in aircraft.

As a result, in this study, we mainly concentrate on the following methods: plane segmentation (PCL-PS, 2020), region growing (PCL-RG, 2020), conditional Euclidean clustering (PCL-CEC, 2020), the difference of normals based segmentation (PCL-DONBS, 2020), and supervoxel clustering (PCL-SC, 2020). For this reason, in this section, we briefly discuss the previous studies that employ these methods while considering the positive and negative aspects. Then, the details of the methods are explained.

2.1. RANSAC

The main idea behind the model-based segmentation methods is to group the points that fit the same mathematical model for primitive shapes such as plane, sphere, cone, cylinder, cubes, and torus. Fischler and Bolles (1981) introduced the well-known model-based RANSAC algorithm used in segmenting planar surfaces in point cloud data. Since the RANSAC does not need to know the relation between neighbor points, it could be applied to both organized point clouds acquired by RGB-D camera (Lu and Song, 2015) and unorganized point cloud data captured with 3D LiDAR or laser sensors (Xu et al., 2016) without any preprocessing step. Moreover, Schnabel et al. (2007) presented a RANSAC-based method for automatically segmenting fundamental shapes such as plane, cylinder, and cones in both mesh and point cloud data. They optimized the method for time complexity and they showed the robustness of the proposed method against outliers. There is an enormous number of studies that apply the RANSAC method for segmentation both in indoor and outdoor applications. In the reviews, Nguyen and Le (2013) and Grilli et al. (2017) explained most of these studies. Besides, several reviews for RANSAC-based methods were published (Kim and Yu, 2009; Raguram et al., 2012). RANSAC is an iterative and fast method and also robust against the noise and the outliers in the point cloud. However, RANSAC-based methods can lead to inaccurate results when the point cloud includes complex-shaped objects and the points in the scene do not evenly distribute. Also, the RANSAC-based method may not recognize coplanar surfaces since they only utilized the mathematical model of the planes.

The RANSAC is mainly a prediction approach that is executed iteratively to segment point cloud data. The algorithm starts with the selection of the mathematical model. Then, a small set of feasible points is selected randomly instead of searching the large set of points that fit the model. The small set is enlarged regarding the DISTANCE_THRESHOLD parameter. The distance between a point (p=(x_p,y_p,z_p)) and the model is calculated as given in Equation 1. In the equation, we assume that the model is plane and the a, b, c and d describe plane parameters. The distance value can be considered as the error. If the error is less than the threshold, the point is added to that model and the model is updated. After a model is entirely segmented, the points that belong to the model are extracted from the point cloud and the process is repeated until the number of remaining points reaches the predefined number.

$$distance = \frac{ax_p+by_p+cz_p+d}{\sqrt{a^2+b^2+c^2}} \qquad (1)$$

2.2. Region Growing

The region-based methods utilize the local information to cluster the points into regions. To determine the points to be added to a region, the features such as surface orientation, curvature, normal, etc. are investigated for points in a predetermined radius or a certain number of neighbors (Rabbani et al., 2006; Jagannathan and Miller, 2007). Therefore, a preprocessing step is required to define neighborhood relationships before using these methods with unorganized point cloud data (Vo et al., 2015). In the reviews, Nguyen and Le (2013) and Grilli et al. (2017) divide the region-based methods into two categories: Top-down methods (unseeded) and bottom-up (seeded) methods. The top-down methods start with one region that includes all points in the point cloud data. Then, they separate the region into subregions according to a criterion. The success of these methods highly depends on the selected criterion. On the contrary, the bottom-up methods first, select some seed points, and then points that satisfy a predefined condition, are joined to seeds to construct the regions (Besl and Jain, 1988). The region-based methods are generally robust against the noise. The success of these methods relies on selecting seed points and the merging criterion (Xie et al., 2019). Besides, these methods are sensitive to inaccurate normal and curvature values and this situation can cause incorrect results at points in the boundary of the regions where normal and curvature values change very quickly. To avoid these incorrect results, the search radius or number of neighbors around a point can be increased. In that case, the elapsed segmentation time increases. Therefore, there is a trade-off between the success of segmentation and the segmenting time in region-based approaches (Nguyen and Le, 2013).

Although the region-based methods are separated into two categories, the researchers generally prefer to use bottom-up (seeded) methods. Besl and Jain (1988) introduced the first seeded region growing algorithm. The algorithm consists of two steps: 1) Determining the seed points, 2) growing the seed to form regions. The selection of seed points is an essential step for the success of the region-based methods. Besl and Jain (1988) determined the seed points according to the curvature value of the points. However, seed selection may vary for other methods. For example, Rabbani et al. (2006) and Ning et al. (2009), first, determined an appropriate model for a point and its neighbors to select a seed point. Then, the point that has the smallest distance with the plane created with the neighbor points is chosen as a seed point. Once the seed point is specified, a search list is created as pushing the seed point and its neighbors. At that point, the search begins to examine the similarities between the seed point and other list items in terms of local features such as curvature, surface orientation, smoothness, normal, etc. Besl et al. (1988) preferred normal of the points as the local feature and the angle between normal vectors (n_i and n_j) was calculated. If the angle is less than a predefined threshold value, the neighbor point is added to the region corresponding to the seed point (Equation 2). After that, the seed point is removed from the search list, and the recently added point becomes the new seed. The process is repeated for new seed points until no seed point for that region is left. The method continues with a new seed point for a new region.

$$\cos^{-1}(\boldsymbol{n_i}.\boldsymbol{n_j}) < threshold \qquad (2)$$

2.3. Conditional Euclidean Clustering

Conditional Euclidean clustering is a variant of the region growing approach, in which local features that are employed to cluster the neighbor points can be customized according to the application. In this way, different features such as color, normal, smoothness, and distance can be aggregated for segmentation (PCL-CEC, 2020). For example, Wu et al. (2019) combined color and distance features in conditional Euclidean clustering to segment the juicy peaches on trees. Zhou et al. (2018) generated topological maps by means of signed surface variation and distance-based conditions. Besides, conditional Euclidean clustering allows determining the number of points in a cluster and applying different local features for different regions of the point cloud or same clustering criterion with varying parameters to segment different-sized objects in a point cloud data. For these reasons, conditional Euclidean clustering method is appropriate for object segmentation and planar surface segmentation. The main disadvantage of the method is that its success depends on the merging condition of region growing.

In this study, apart from the region growing method that uses similarities between normal vectors of neighbors as a clustering constraint, we opted to consider the x, y, or z components of the point's normal to merge the neighbor points into the same segment in conditional Euclidean clustering method. To do that, for each point, we first determined the largest component of its normal vector according to the Equation 3. We called that component as the dominant axis of the normal and employed it for planar surface segmentation. If neighbor points have the same dominant axis, these points are added to the same segment. The idea behind this is that the normal vectors of the planar surfaces generally parallel to one of the x, y, or z-axis. In this way, we aimed to reduce the segmentation errors that occur at the boundary regions of planar surfaces when the similar normal vectors criterion is applied for segmentation like in the region growing method.

$$\text{dominant axis} = \arg max_n(n_x, n_y, n_z) \qquad (3)$$

2.4. The Difference of Normal Based Segmentation

In computer vision applications, the operators such as the difference of Gaussians (DoG) and Laplacian of the Gaussian (LoG) have been widely used to detect edges, to find salience points, and to pre-segment images, before complex algorithms are executed. In a similar fashion, the difference of normals (DON) operator, which calculates normal vectors for large-scale and small-scale support radiuses, is mainly designed to identify essential and distinguishing points needed for object identification within large unorganized the point cloud data for outdoor applications (Ioannou et al., 2012). For example, Su et al. (2018) applied the difference of normals method to extract corn leaves. The main advantage of the method is that the support radiuses can be adjusted depending on the object to be segmented. However, the size of the support radius should be selected carefully since the calculation time of the normal vectors highly depends on this selection. Besides, apart from the region-based methods, the difference of normals does not need the local information to determine the segments.

The main idea behind the difference of normals method is to observe the surface normal vectors that describe the surface geometry. The researchers generally prefer to use the support radius or a fixed number of neighbors to determine the surface normal in spite of the existence of many different methods. The radius or the number of neighbors describes the size of the surface that the normal represents. The difference of normals method calculates surface normal vectors for the small support radius (r_s), which reflects small changes on the surface geometry, and the large support radius (r_l) that depicts the general character of the surface. If the small support radius normal and large support radius normal for a point are similar to each other, it indicates that the normal vectors reflect the character of the surface. However, the difference between small support radius normal and large support radius normal increases when a point lies at the edge of a planar surface due to the rapid changes in small support radius normal. The difference of normals based segmentation method utilizes from that idea. First, the method executes the difference of normals operator to calculate the arithmetic average of the small and large support radius normal vectors for each point (Equation 4). Then, conditional removal filtering (PCL-CRF, 2020) is applied to remove the points that the difference between two normal vectors is greater than a predefined threshold value (FILTER_THRESHOLD). In other words, the filtering step eliminates the points located at the edge of the planar surfaces. Finally, the Euclidean cluster extraction method is employed to cluster the remaining points (Ioannou et al., 2012).

$$\Delta\boldsymbol{n}\ (p, r_s, r_l) = \frac{\boldsymbol{n}(p,r_s) - \boldsymbol{n}(p,r_l)}{2} \tag{4}$$

2.5. The Supervoxel Clustering

In recent years, graph-based methods have gained popularity in 2D image segmentation due to their pixel-based strategy that significantly increases the segmentation success (Egger et al., 2012). These methods construct graphs, in which each pixel is represented with a node. Then, the nodes are processed to enhance pixel-based low-level data to object-scale high-level information. On the other hand, the graph-based methods suffer from high time complexity since they handle all pixels in the image. To reduce the computational time of the graph-based methods, the superpixel concept, which considers a group of pixels instead of each pixel as a node, is revealed. After the successful implementation of the superpixel concept for 2D image segmentation, the supervoxel concept has been begun to be used for segmentation of point cloud data. For example, Lin et al. (2018) proposed an improved supervoxel implementation to diminish the segmentation errors at the supervoxel boundaries. They showed the effectiveness of their work on three different datasets, including different types of point cloud data.

The supervoxel clustering method first creates the voxel octree structure regardless of whether the point cloud data is organized or unorganized. Moreover, an adjacency graph with a resolution of R_{voxel} is constructed, this is essential for segmentation of the point cloud data. Although these steps seem to be time-consuming and can be considered a negative aspect of the method, they allow utilizing neighbors relation and speeding up the segmentation process. Then, the seed points must be selected because the supervoxel clustering method is a variant of the region-based methods. It is clear that the selection of the seed points profoundly affects the segmentation accuracy. For that reason, R_{seed} parameter, which should be chosen greater than R_{voxel} value, is used to determine the evenly distributed seed points in the voxelized grid. The seed voxels and its neighbors form the supervoxels that substantially are the cluster of features such as color, spatial distance, and normal. After supervoxels are determined, the growing process is executed with the k-means clustering algorithm and breadth-first search. First, the nearest voxel to the supervoxel and its neighbors are handled. Then, the similarity between these voxels and the supervoxel in the feature domain is investigated. Since the supervoxels consist of color, distance, and normal features, the distance between supervoxel and the voxels is calculated with a weighted distance function as given in Equation 5:

$$D = \sqrt{w_c D_c^2 + \frac{w_s D_s^2}{3R_{seed}^2} + w_n D_n^2} \tag{5}$$

where D_c, D_s, and D_n represent the color, spatial, and normal distance, respectively. Also, weights of these distances are described with w_c, w^s, and w_n, respectively. The voxel that has the smallest distance value is joined to the supervoxel and its neighbors in the adjacency graph are added to the search queue. The same process is performed for all supervoxels and the depth of the examined voxels remains the same. The search is terminated when the boundaries of supervoxel are reached or no neighbors are available for search. Lastly, the k-means clustering algorithm is applied to update the cluster centers. The process continues until either the supervoxels stabilize or a predefined number of iterations is performed (Papon et al., 2013).

3. Results

3.1. Experimental Setup

The publicly available Fukuoka indoor dataset was used to analyze the performance of the segmentation methods that are situated in PCL. The dataset consists of panoramic point cloud data captured with the SICK LMS-151 laser sensor. The point clouds are separated into five categories: Corridor, kitchen, laboratory, study room, and office. In the laboratory, kitchen, and study room categories, samples generally include objects such as tables, computer screens, and chairs. However, in this study, we intend to segment structural planar surfaces such as wall, floor, and ceiling. For this reason, we selected 30 samples from the corridor and office categories that mainly include points representing these planar surfaces. Besides, we elaborated on the selected samples containing different levels of noise. Each of the selected point clouds has approximately 474200 points. Some preprocessing steps were performed to prepare the point cloud data for segmentation. First, the points that belong to non-planar surfaces such as flowerpot, air conditioning parts, and fire alarms were removed because these points may adversely affect the performance of the methods. For this reason, the natural characteristic of these methods can be misunderstood. Then, the normal vector of each point in the point cloud data was calculated since all methods use normal information as a local feature for segmentation, excluding the RANSAC. For each point, to determine the normal vector, firstly, a surface is fitted with the neighbors around the point. Then, the surface normal is considered the normal vector of the point. Thus, this process allowed us to compare the methods fairly because all methods employed the same normal vectors (PCL-SS, 2020). The experiments were conducted on a PC with the Intel i7-7700HQ processor, 16GB RAM, and Ubuntu 16.04 operating system.

The test results were compared in terms of segmentation accuracy and the time elapsed for segmentation. In order to calculate the segmentation accuracy, point-wise labeling was performed for each point cloud data through the RViz cloud annotation tool (Monica et al., 2017). Then, we determined paired segments between the segmentation result of a method and the labeled data. The segmentation accuracy of a sample is calculated by dividing the number of correctly segmented points to the total number of points. The average of all samples is considered the average segmentation accuracy.

In the RANSAC method, the DISTANCE_THRESHOLD parameter that determines whether a point is in the plane. The parameter should be selected as high as possible to cope with the noise. However, all points could be clustered into a segment when the parameter is too high. The parameter was selected empirically as 0.03 meters to consider these circumstances. In the region growing method, we used the default values at the PCL. The number of neighbors that examined in segmentation was 30. Also, the curvature and smoothness threshold were set to 1 and 3 degrees, respectively. In the conditional Euclidean clustering method, the neighborhood relationship is established with the search radius. We tried to set the radius of the conditional Euclidean clustering method, in which the number of neighbors to be approximately 30. In the experiments, the radius was selected as 0.015 meters. Also, the minimum and the maximum number of points that a segment can possess were 3000 and 1000000, respectively. In the difference of normal based segmentation method, small support radius (r_s) and large support radius (r_l) were 0.015 and 0.075 meters, respectively. Notice that, we selected the small support radius the same as the radius in the conditional Euclidean clustering method to compare the methods fairly. Besides, Ioannou et al. (2012) recommended selecting the large support radius five times the small support radius. There are five main parameters used in supervoxel clustering. Voxel size value (R_{voxel}), which determines octree leaf resolution, was selected empirically as 0.01 meters. The seed size value (R_{seed}) that describes the maximum size of the supervoxel was chosen 5.0 meters to contain all points in the point cloud data. In this study, we used only normal values in the supervoxel clustering method. Therefore, weights for color (w_c), spatial distance (w_s), and normal (w_n) were 0.0, 0.0, and 1.0, respectively.

3.2. Results

3.2.1. Segmentation time

The segmentation time is one of the essential metrics to evaluate the segmentation methods. The segmentation methods mentioned in Section 2 were applied to the selected 30 scenes for examining the performance of these methods in terms of segmentation time. Table 1 shows the average elapsed time to segment a scene for each method.

Table 1. Average segmentation time of a scene

Methods	Average Time (Seconds)
RANSAC	0.4285
Region Growing	6.2855
Conditional Euclidean Clustering	5.6124
Difference of Normal Based Segmentation	3.2140
Supervoxel Clustering	5.2726

The conditional Euclidean clustering and the supervoxel clustering methods are variants of the region-based methods. These methods utilize the local information to form a region; therefore, segmentation time for these methods was longer when they are compared with the RANSAC and the difference of normals based segmentation. This is an expected result because searching for neighbors is a time-consuming operation and the RANSAC and the difference of normals based segmentation do not perform the operation for segmentation. The segmentation time of region-based methods was also different from each other. The slight difference between the region growing and the conditional Euclidean clustering could be explained with different search approaches that these methods use. Generally, region growing based methods search their vicinity with a radius or the number of neighbors. As applying the region growing, the number of neighbors was selected 30. On the other hand, the conditional Euclidean clustering method uses the radius, which was chosen as 0.015 meters. Although we tried to set the radius of the conditional Euclidean clustering method, in which the number of neighbors to be approximately 30, the number of neighbors may vary according to the density of the points. Apart from the region growing and the conditional Euclidean clustering, which are utilized point-wise neighborhoods, the supervoxel clustering method uses voxel-wise neighborhoods. Therefore, in the worst-case scenario, each voxel includes only one point and the performance of the region growing and the supervoxel method could be the same in terms of the segmentation time. As a result, the number of points in a supervoxel that is adjusted with the Rvoxel parameter determines the segmentation time. The difference of normals based segmentation method could be considered as the edge-based methods since the method essentially is designed for the detection of the edges. In theory, the main advantage of the edge-based methods is their low computational complexity. Once the segmentation time performance of the difference of normals based segmentation method is analyzed, we separated elapsed time into three stages. In the first stage, the difference of normal vectors was calculated in 0.1265 seconds. Then, conditional removal filtering took 0.016 seconds. In the last stage, Euclidean clustering was applied and it required 3.0715 seconds. Thus, the time-consuming part of the method is clustering and if an appropriate clustering method is chosen, the segmentation time could be decreased. Finally, RANSAC was the fastest one among the methods. It is an expected result because the RANSAC tries to fit a mathematical model and it does not utilize the local information to segment the planes.

3.2.2. Segmentation accuracy

In order to examine the success of methods that are situated in PCL, the segmentation accuracy was calculated for each method. Table 2 shows the average segmentation accuracy for a scene. Although RANSAC, region growing, and the conditional Euclidean clustering methods yielded approximately the same segmentation accuracy, they behaved differently in different scenes. Also, the reasons why the difference of normals based method and the supervoxel clustering method produce low segmentation accuracy when they are compared to the other methods were discussed.

Table 2. Average segmentation accuracy of a scene

Methods	Average Accuracy
RANSAC	94.33%
Region Growing	94.19%
Conditional Euclidean Clustering	93.54%
Difference of Normal Based Segmentation	89.50%
Supervoxel Clustering	64.49%

To analyze the characteristics of each method, we selected four scenes among the 30 scenes, which are shown in columns of Figure 1. Another criterion of the selection of these scenes was the noise level of the data. We selected scenes that include different noise levels. Scene 1 and scene 2 include low-level noise when they are compared with scene 3 and scene 4. The first row in the figure shows the ground truth of the scenes. The remaining rows depict the results for each segmentation method. Also, the segmentation accuracy and time for each scene is given at the right of the scenes. In figures, the incorrect or unsegmented regions are represented with white ellipses. The RANSAC method suffers from combining the coplanar surfaces that are located in different locations on the same surface. For example, in scene 1, there are two vertical planes, which lie on two sides of the door plane. These planes approximately fit the same mathematical model and the RANSAC does not utilize the local information. Thus, the RANSAC added these planes into one segment, which is shown with yellow color. A similar problem also occurred in scene 4 and the points in the ellipses were incorrectly joined into the wall plane that is depicted in yellow color. In order to deal with the high noise, the DISTANCE_THRESHOLD parameter was selected as high as possible. Therefore, the high value provides accurate segmentation at intersection points as the noise level increases and scene 3 and scene 4 show examples of this situation. On the other hand, the incorrect results were produced at the intersection points of two planar surfaces in scene 2 due to the high value in the selection DISTANCE_THRESHOLD parameter. The well-known negative aspect of the region growing methods is that they may lead to incorrect results at points in the boundary of the regions where normal and curvature values change very quickly. The ellipses in the figures show the unsegmented regions. As the noise level increases just like scene 3 and scene 4, the normal and the curvature values are more corrupted than the values in scene 1 and scene 2. Thus, the segmentation accuracy for scene 3 and scene 4 are low since the number of unsegmented points directly proportional to the noise level. Besides, the required time for segmentation increases as the noise level rises. In this study, the dominant axis concept was suggested for the merging constraint of the conditional Euclidean clustering method. Therefore, the success of the method heavily depends on the accuracy of the normal. The method offered promising results for scene 1 and scene 2. Moreover, for these scenes, the method outperformed the RANSAC and region growing methods in terms of segmentation accuracy. However, as the noise level increases, the success of the method dramatically decreases. Nevertheless, the potential of the method should not be underestimated. The difference of normals based segmentation method employs the filtering process after small and large support radius normal vectors are calculated. Thus, the success of the method heavily depends on the filtering process. The filtering algorithm uses a threshold to determine whether a point is filtered. If the threshold is low, all points that the normal values are changed larger than the threshold are eliminated. All points could be combined into one segment when the threshold is high. As a result, we observed that the best accuracy success is obtained when the threshold was set to 0.1. However, the threshold could be adaptively adjusted according to the noise level to increase the segmentation accuracy. The method can always be considered for the applications that need fast segmentation and when the data contains coplanar surfaces that the RANSAC prone to error. The supervoxel clustering method was not appropriate for the segmentation of planar surfaces since it only allows adjusting the R_{voxel} parameter. The parameter determines the number of supervoxel and their boundaries. Then, the method clusters the points in each supervoxel according to the normal vectors. The points that have similar normal vectors are added to the same region and this causes incorrect segmentation results.

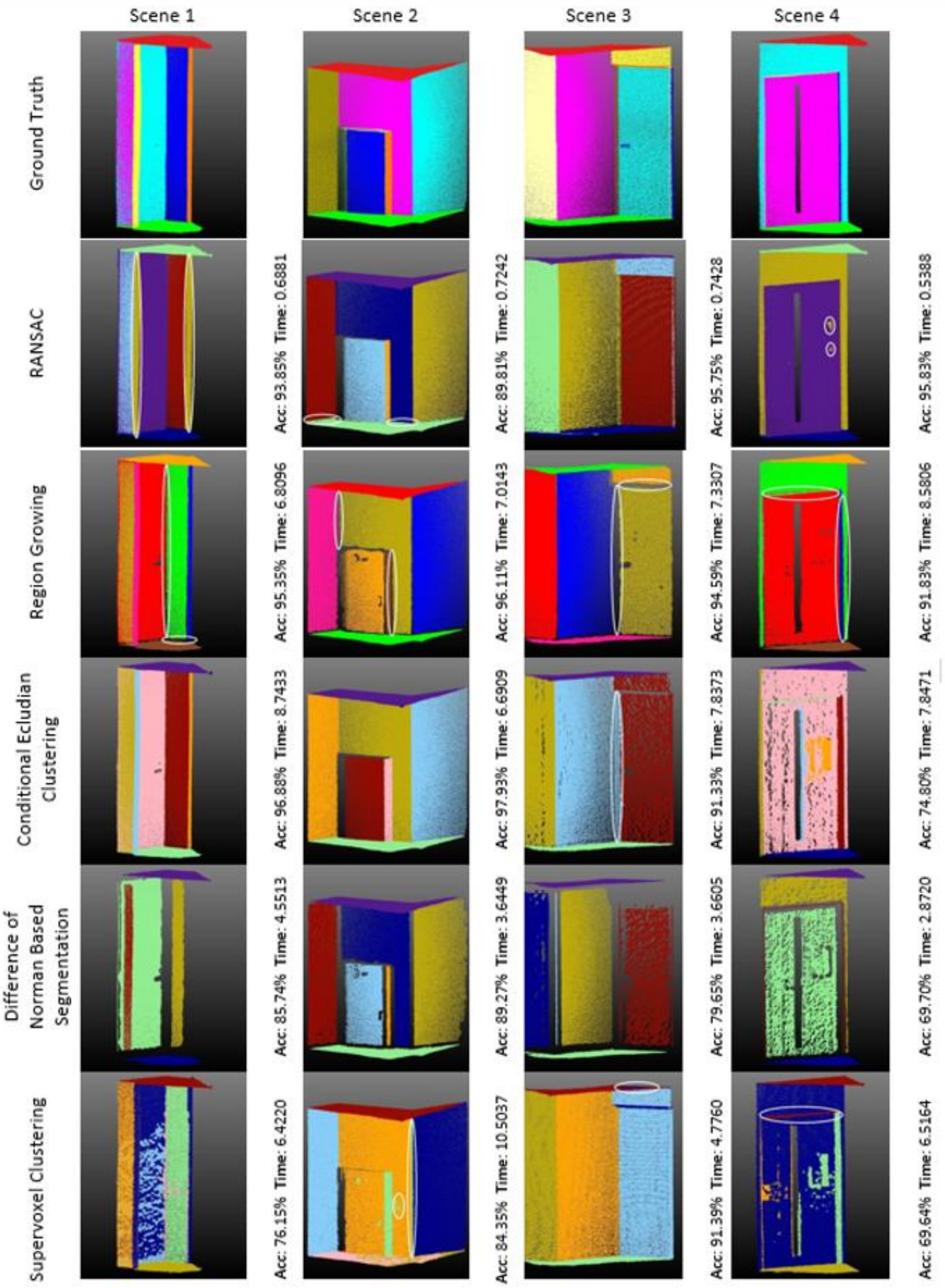

Figure 1. The segmentation results.

4. Conclusion

In this study, the segmentation of indoor structural planar surfaces such as wall, floor, and ceiling via point cloud data that is captured with the 3D laser range scanner was considered. In previous studies, the researchers generally used RANSAC and region-growing methods. However, other segmentation methods such as conditional Euclidean clustering, the difference of normals based segmentation, and the supervoxel clustering may produce satisfactory results for indoor applications. For this reason, we examined the performance of these methods in terms of segmentation accuracy and segmentation time to reveal the positive and negative aspects of these methods for researchers that plan to apply them to 3D laser point cloud data. To do that, the publicly available Fukuoka indoor laser dataset, which has point clouds with different noise levels, was used. The test results were evaluated regarding the noise level of point clouds and the behaviors of each method against the noise were discussed.

Conflict of interest

The authors declare that there is no conflict of interest.

References

Anil EB, Tang P, Akinci B, Huber D. 2013. Deviation analysis method for the assessment of the quality of the as-is Building Information Models generated from point cloud data. Aut in Construc, 35: 507-516.

Besl PJ, Jain RC. 1988. Segmentation through variable-order surface fitting. IEEE Transact on Pat Analy Machine Intel, 10(2): 167-192.

Cadena C, Košecka J. 2015. Semantic parsing for priming object detection in indoors RGB-D scenes. Int J Robotics Res, 34(4-5): 582-597.

Egger J, Colen RR, Freisleben B, Nimsky C. 2012. Manual refinement system for graph-based segmentation results in the medical domain. J Medic Sys, 36(5): 2829-2839.

Ferraz A, Bretar F, Jacquemoud S, Gonçalves G, Pereira L. 2010. 3D segmentation of forest structure using a mean-shift based algorithm. IEEE Int Conference on Image Processing, 1413-1416.

Fischler MA, Bolles RC. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun of the ACM, 24(6): 381-395.

Grilli E, Menna F, Remondino F. 2017. A review of point clouds segmentation and classification algorithms. The Intl Archives of Phot, Remote Sensing and Spatial Inf Sci, 42: 339.

Himmelsbach M, Hundelshausen FV, Wuensche HJ. 2010. Fast segmentation of 3D point clouds for ground vehicles. IEEE Intelligent Vehicles Symposium, 560-565.

Ioannou Y, Taati B, Harrap R, Greenspan M. 2012. Difference of normals as a multi-scale operator in unorganized point clouds. Second Int Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, 501-508.

Jagannathan A, Miller EL. 2007. Three-dimensional surface mesh segmentation using curvedness-based region growing approach. IEEE Transactions on Pat Analy Machine Intel, 29(12): 2195-2204.

Kim T, Yu W. 2009. Performance evaluation of ransac family. In Proceedings of the British Machine Vision Conference (BMVC), 1-12.

Koppula HS, Anand A, Joachims T, Saxena A. 2011. Semantic labeling of 3d point clouds for indoor scenes. In Adv in Neural Inf Processing sys, 244-252.

Lin Y, Wang C, Zhai D, Li W, Li J. 2018. Toward better boundary preserved supervoxel segmentation for 3D point clouds. ISPRS J Photogrammetry and remote sensing, 143: 39-47.

Lu Y, Song D. 2015. Robustness to lighting variations: An RGB-D indoor visual odometry using line segments. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 688-694.

Martinez Mozos O, Nakashima K, Jung H, Iwashita Y, Kurazume R. 2019. Fukuoka datasets for place categorization. Int J Robotics Res, 38(5): 507-517.

Mattausch O, Panozzo D, Mura C, Sorkine-Hornung O, Pajarola R. 2014. Object detection and classification from large-scale cluttered indoor scans. Computer Graphics Forum, 33(2): 11-21.

Monica R, Aleotti J, Zillich M, Vincze M. 2017. Multi-label point cloud annotation by selection of sparse control points. International Conference on 3D Vision (3DV), 301-308.

Morsdorf F, Meier E, Kötz B, Itten KI, Dobbertin M, Allgöwer B. 2004. LIDAR-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. Remote Sensing of Envir, 92(3): 353-362.

Mutlu B, Hacıömeroğlu M, Serdar GM, Dikmen M, Sever H. 2014. Silhouette extraction from street view images. Int J Advanced Robotic Sys, 11(7): 114.

Nguyen A, Le B. 2013. 3D point cloud segmentation: A survey. 6th IEEE conference on robotics, automation and mechatronics (RAM), 225-230.

Ning X, Zhang X, Wang Y, Jaeger M. 2009. Segmentation of architecture shape information from 3D point cloud. Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry, 127-132.

Papon J, Abramov A, Schoeler M, Worgotter F. 2013. Voxel cloud connectivity segmentation-supervoxels for point clouds. Proceedings of the IEEE conference on computer vision and pattern recognition, 2027-2034.

PCL-Color-Based Region Growing (PCL-CBRG). 2020. Point Cloud Library Color-Based Region Growing Code. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/region_growing_rgb_segmentation.html#region-growing-rgb-segmentation (access date: 01.05.2020)

PCL- Conditional Euclidean Clustering (PCL-CEC). 2020. Point Cloud Library Conditional Euclidean Clustering Segmentation Code. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/conditional_euclidean_clustering.html#conditional-euclidean-clustering (access date: 01.05.2020).

PCL- Conditional Removal Filtering (PCL-CRF). 2020. Point Cloud Library Conditional Removal Filtering Code. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/conditional_removal.html#conditional-removal (access date: 01.05.2020).

PCL- Difference of Normal Based Segmentation (PCL-DONBS). 2020. Point Cloud Library Difference of Normal Based Segmentation Codes. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/don_segmentation.html#don-segmentation (access date: 01.05.2020).

PCL-Euclidean Clustering Extraction (PCL-ECE). 2020. Point Cloud Library Euclidean Clustering Extraction Code. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/cluster_extraction.html#cluster-extraction (access date: 01.05.2020).

PCL-Min-Cut Based Segmentation (PCL-MCBS). 2020. Point Cloud Library Min-Cut Based Segmentation Code. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/min_cut_segmentation.html#min-cut-segmentation (access date: 01.05.2020).

PCL-Plane Segmentation (PCL-PS). 2020. Point Cloud Library Plane Segmentation Code. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/planar_segmentation.html#planar-segmentation (access date: 01.05.2020).

PCL-Progressive Morphological Filter Segmentation (PCL-PMFS). 2020. Point Cloud Library Progressive Morphological Filter Segmentation Code. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/progressive_morphological_filtering.html#progressive-morphological-filtering (access date: 01.05.2020).

PCL-Region Growing (PCL-RG). 2020. Point Cloud Library Region Growing Segmentation Code. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/region_growing_segmentation.html#region-growing-segmentation (access date: 01.05.2020).

PCL-Segmentation (PCL-S). 2020. Point Cloud Library Segmentation Codes. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/index.html#segmentation (access date: 01.05.2020).

PCL- Supervoksel Clustering (PCL-SC). 2020. Point Cloud Library Supervoksel Clustering Segmentation Code. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/supervoxel_clustering.html#supervoxel-clustering (access date: 28.04.2020).

PCL-Surface Smoothing (PCL-SS). 2020. Point Cloud Library Surface Smoothing Code. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/resampling.html?highlight=resampling#smoothing-and-normal-estimation-based-on-polynomial-reconstruction (access date: 02.05.2020).

Qu T, Coco J, Rönnäng M, Sun W. 2014. Challenges and trends of implementation of 3D point cloud technologies in building information modeling (BIM): case studies. Computing in Civil and Build Eng, 809-816.

Rabbani T, Van Den Heuvel F, Vosselmann G. 2006. Segmentation of point clouds using smoothness constraint. Int archives of Photogrammetry, Remote Sens and spatial Inf Sci, 36(5): 248-253.

Raguram R, Chum O, Pollefeys M, Matas J, Frahm JM. 2012. USAC: a universal framework for random sample consensus. IEEE Trans on Pattern Analy and Machine Int, 35(8): 2022-2038.

Rusu RB, Cousins S. 2011. 3d is here: Point cloud library (pcl). IEEE international conference on robotics and automation, 1-4.

Rusu RB, Marton ZC, Blodow N, Dolha M, Beetz M. 2008. Towards 3D point cloud based object maps for household environments. Robotics and Autonomous Sys, 56(11): 927-941.

Schnabel R, Wahl R, Klein R. 2007. Efficient RANSAC for point-cloud shape detection. Comp Graphics Forum, 26(2): 214-226.

Su W, Zhang M, Liu J, Sun Z. 2018. Automated extraction of corn leaf points from unorganized terrestrial LiDAR point clouds. Int J Agri and Biol Eng, 11(3): 166-170.

Tarsha-Kurdi F, Landes T, Grussenmeyer P. 2007. Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data. ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, Sep 2007, Espoo, Finland, 407-412.

Vo AV, Truong-Hong L, Laefer DF, Bertolotto M. 2015. Octree-based region growing for point cloud segmentation. ISPRS J Photoy and Remote Sensing, 104: 88-100.

Wu G, Zhu Q, Huang M, Guo Y, Qin J. 2019. Automatic recognition of juicy peaches on trees based on 3D contour features and colour data. Biosystems Eng, 188: 1-13.

Xie Y, Tian J, Zhu XX. 2019. A review of point cloud semantic segmentation. arXiv preprint arXiv:1908.08854.

Xu B, Jiang W, Shan J, Zhang J, Li L. 2016. Investigation on the weighted ransac approaches for building roof plane segmentation from lidar point clouds. Remote Sens, 8(1): 5.

Zermas D, Izzat I, Papanikolopoulos N. 2017. Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications. IEEE International Conference on Robotics and Automation (ICRA), 5067-5073.

Zhou W, Peng R, Dong J, Wang T. 2018. Automated extraction of 3D vector topographic feature line from terrain point cloud. Geocarto Int, 33(10): 1036-1047.