

Automatic point cloud segmentation using RANSAC and DBSCAN algorithm for indoor model

Harintaka, Calvin Wijaya

Department of Geodetic Engineering, Faculty of Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia

Article Info

Article history:

Received May 5, 2023

Revised Jul 13, 2023

Accepted Jul 30, 2023

Keywords:

Density-based spatial clustering of application

Low-cost terrestrial laser scanners

Machine learning

Point cloud

Random sample consensus

ABSTRACT

Indoor modeling is a crucial aspect of architecture, engineering, and construction (AEC) projects. While terrestrial laser scanners (TLS) are the most popular tool for acquiring indoor geometry, processing point clouds from TLS scans with manual methods can be inefficient and error-prone. This study proposes a machine learning algorithm to automatically segment point clouds acquired by low-cost TLS. Random sample consensus (RANSAC), a simple yet effective algorithm for segmenting planar surfaces such as walls, ceilings, and floors, is used in the segmentation process. The resulting segmentation is then refined using density-based spatial clustering of application with noise (DBSCAN) to group nearby points that were not segmented correctly by RANSAC into the appropriate segment. The result is a segmented point cloud consisting of five indoor elements: wall, ceiling, floor, column, and interior. The algorithms were found to be effective for segmenting small and simple rooms. For larger or more complex rooms, segmentation can be performed by dividing the room into several parts and applying the algorithms to each partition. Overall, the study demonstrates the potential of machine learning algorithms for automating point cloud segmentation tasks in indoor modeling, especially for low-cost TLS scans.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Calvin Wijaya

Department of Geodetic Engineering, Faculty of Engineering, Universitas Gadjah Mada

Jl. Grafika No.2, Sinduadi, Mlati, Sleman, Daerah Istimewa Yogyakarta 55281, Indonesia

Email: calvin.wijaya@mail.ugm.ac.id

1. INTRODUCTION

The 3D indoor modeling of the building becomes a prominent issue since building information modelling (BIM) was introduced into architecture, engineering, and construction (AEC) industries [1]. The indoor model is important because it serves detailed information about the objects inside the room. The information can be used for various things such as reconstruction, documentation, maintenance, and quality control until simulation [2]. The 3D indoor model gives a digital representation of a room with comprehensive detail so the architect, engineer, and building manager can access information from its digital twin to acquire information. It is shown that indoor modeling become one of the main information in controlling and managing a building [3].

Geometry data of a room for modeling can be acquired with several instruments such as terrestrial instruments. However, with the rapid development of 3D acquisition technology, high-definition instrument (HDS) instruments are the favorite and most popular in urban use. Terrestrial laser scanner (TLS) as one of the HDS instruments can be used to accelerate data acquisition for 3D models. Output or scan result from TLS is a point cloud, which is a collection of points. Point clouds with huge sizes and numbers can be used to create 3D models. With TLS, it is simpler to make a 3D model due to its ability to capture the point cloud, as the point cloud itself is the base data to create a 3D representation. Besides using TLS, point clouds and 3D

models also can be acquired using close-range photogrammetry (CRP) method. CRP is a standard photogrammetric work, but it has done in a short range between camera and object. CRP is a photogrammetric procedure to capture an object within 1 meter to 200 meters from the instrument with an accuracy 0.1 millimeter or below that, used for architecture, engineering, and construction [4]. CRP uses the same algorithm as the standard photogrammetric procedure, the difference is only the distance between the camera and the object [4]–[6]. The output or result of CRP is overlapping photos that can be processed to create a set of point cloud.

Point cloud data, whether from TLS, CRP, or a combination of both have very large and huge numbers up to millions of points. Currently, point cloud processing from raw data into a model that can be served as indoor geometry for modeling still using manual process [7], [8]. This process is inefficient because the number of the point cloud is very large so it is expensive to process and consume many times [9]. Also, the manual process is very vulnerable to human error because needed skills to process point cloud. Therefore, the problem of point cloud processing is to automate the process from raw data point cloud into 3D indoor models or elements efficiently. Automating the process is not an easy task, because point cloud has unique characteristics and it is irregular, unordered, and unstructured. It also has a huge number and also noises inside, so it is difficult to automate the process [10], [11].

Several algorithms have been developed to process point clouds automatically, including a machine learning algorithm. Algorithms such as region growing, random forest, random sample consensus (RANSAC), density-based spatial clustering of application with noise (DBSCAN), and others have shown significant progress to use as point cloud automatic segmentation algorithms [11]. Region-growing algorithms use similarity criteria between points such as color and position of points that are close together for segmentation criteria. The researches [8], [9] use a region growing for automatic segmentation. However, this algorithm is very susceptible to over-segmentation or under-segmentation problems. This algorithm also heavily depended on the criteria for growth. It must well define before segmentation, and it can be different for a different dataset. Kang *et al.* [12] improve the region-growing algorithm by using Gaussian and average curvature to sort point clouds and set the optimal threshold to optimize the segmentation result. However, this research does not specific to indoor applications and lacks comprehensive evaluation metrics.

Random forest (RF) is a popular machine learning algorithm that is mainly used in classification and regression tasks. Several researchers also use this algorithm to classify point cloud data. Even though Poux *et al.* [9] uses a region growing for unsupervised segmentation, in the final classification it then uses the RF algorithm, so as [13]. Xue *et al.* [14] tries to improve the RF algorithm based on iterative closest spectral framework (ICSF) algorithm and weakly correlated random forest. Another research by Xing *et al.* [15] also uses an RF classifier for the automatic segmentation of point cloud data in urban areas. Ni *et al.* [16] presents an automated and effective framework for classifying airborne point clouds with RF as its classifier. The research that uses the RF algorithm is mostly applied to outdoor and urban areas and very little explanation about indoor situations. Even though it is an accurate machine learning algorithm, it is a supervised method where it has a high sensitivity to the training data.

The RANSAC algorithm is also one of the algorithms commonly used for point cloud segmentation [17]. The concept is to use the fitting model principle where it does point cloud fitting to a specific model [18]. Adam *et al.* [19] combine RANSAC with 2D photo for segmentation, but this method does not apply to indoor model. Li *et al.* [20] develop RANSAC based on normal distribution transformation (NDT) to choose an efficient parameter for segmentation, but this method is hard to implement and needs special knowledge for NDT. Research by Runceanu *et al.* [7] using RANSAC for point cloud segmentation indoors and the research focuses on indoor mapping systems and not the algorithm's efficiency for point cloud segmentation. Isa *et al.* [21] using RANSAC and localization approaches to segment point cloud from TLS data of outdoor and indoor scenes in a bell tower, but the data used in the research has a small area and does not explain it for extensive indoor applications such as for BIM or 3D indoor navigation.

DBSCAN algorithm includes a new algorithm in point cloud segmentation, where it segments points based on search radius (ϵ) and a minimum number of point clouds inside one cluster [22]. The researches [23], [24] uses DBSCAN to segment the point cloud. Wang *et al.* [23] use DBSCAN to segment point cloud data acquired using airborne laser scanner in an urban area while Czerniawski *et al.* [24] for indoor segmentation. Based on both research, the search radius (ϵ) is difficult to define, making it a disadvantage of this algorithm. Research by Zhao *et al.* [25] improves the DBSCAN algorithm to solve the lack of automatic estimation of parameters, but the research focuses on improving the algorithm based on a mathematical approach and not an applicative purpose.

Based on several research above and the option of a point cloud segmentation algorithm, this research uses a combination of RANSAC and DBSCAN to solve the problem of automatic point cloud segmentation. The use of RANSAC and DBSCAN for point cloud segmentation algorithms has been widely used due to their effectiveness and versatility. RANSAC algorithm itself is not a new algorithm but has been

modified and combined with other algorithms for many purposes, including for point cloud segmentation [11]. The use of RANSAC proved more effective for point cloud unsupervised segmentation compared with other algorithms, such as Hough transform, principal component analysis (PCA), FPFH, and region growing [1], [17]. RANSAC is also effective to detect inliers and remove outliers or noise in the dataset [20], which makes it a suitable algorithm for this research given that this research uses low-cost TLS.

DBSCAN algorithm is also an unsupervised segmentation algorithm. The advantage of DBSCAN is it does not need several clusters for segmentation, it works based on several parameters. This makes DBSCAN different from another unsupervised algorithm such as K-means, where the user needs to define several clusters for segmentation. However, DBSCAN still can follow up the K-means in terms of accuracy in large datasets [23]. DBSCAN also can detect noise in datasets [22], by separating low-density or noise regions with high-density point clouds. It also makes this algorithm suitable for this research due to the inconsistency density of the point cloud scanned by low-cost TLS.

2. METHOD

In general, research methodology is divided into four main stages, there are data acquisition, data pre-processing, segmentation process, and quality control. This research methodology has been drawn in the flowchart shown in Figure 1. Data acquisition is done using low-cost TLS and CRP methods. The point cloud from both instruments is then integrated into one dataset. Next, the dataset is pre-processed with several steps such as noise filtering, registration, and georeferencing. The processed point cloud is then segmented using RANSAC and DBSCAN algorithms. This research will segment indoor point clouds into five classes: ceiling, floor, wall, column, and interior. The final stage is to analyze and quality control the segmented point cloud.

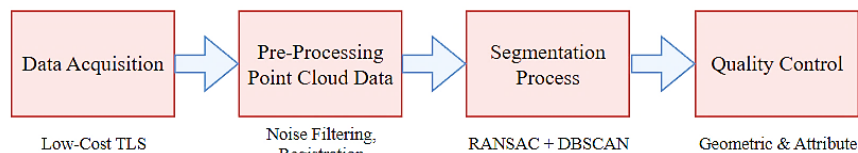


Figure 1. Research workflow

Point cloud data used in this research is primary data collected by author in a building located at Universitas Gadjah Mada. Data acquisition is done using two instruments, there are low-cost TLS and DSLR cameras for the CRP method. Low-cost TLS is used to scan room structures such as walls, ceilings, floors, and columns while a DSLR camera (CRP method) is used to acquire interior point cloud. Acquisition using low-cost TLS is the same as professional TLS where the instrument is set up in the center of the room or in an appropriate overlapping position to acquire point cloud data. Meanwhile for CRP method is done using a convergent or multi-image method where the camera will move rotating the object. Interior or object that will be taken is first stuck with the coded target because many interior objects have smooth and homogeneous surfaces. Low-cost TLS result is raw point cloud data in *.TXT format then converted using calibration data. While the CRP result is an overlap photo that needs to be processed with structure from motion (SfM) algorithm to produce point cloud data. Both point clouds were then integrated and merged into one dataset.

Pre-processing point cloud aims to prepare the point cloud before automatic segmentation. The pre-processing step itself consists of: i) noise filtering, which aims to reduce and minimize noise and outliers in point cloud data. This step takes the longest time, even more, consuming than the main process. This is because low-cost TLS result has a huge number of noises. The process of noise filtering is done using noise and statistical outlier removal (SOR) filters; ii) point cloud registration, which aims to register or combine all point clouds from different rooms into one point cloud dataset only. This step makes all rooms connect. The registration method used in this research is cloud-to-cloud registration; iii) georeferencing, aims to bring point cloud data from a local coordinate into a ground coordinate. This is because there is no position sensor such as GPS inside a low-cost TLS instrument, so the result is a point cloud in local coordinates. The georeferencing process needs ground control points (GCP) that contain real ground coordinates. This research uses four GCP distributed at the corner of the building.

Point cloud segmentation, as the focus of the research, aims to separate or group point clouds into five classes. The process is done automatically using RANSAC and DBSCAN algorithms in the Open3D library. All point cloud data is segmented into five classes, there are ceiling, wall, column, door, and interior. RANSAC and DBSCAN both are unsupervised segmentation algorithms, so it does not need a training

sample. However, it needs to define parameters well for both algorithms. RANSAC and DBSCAN algorithms have parameters to do the segmentation. Both algorithms were implemented using Open3D created by Zhou *et al.* [26], so the parameters will follow the documentation. RANSAC parameter's *distance_threshold* represents the distance threshold from a point into a plane to be stated as inlier, *ransac_n* shows how many points are used to make a plane segment and *num_iteration* shows how many iterations will be done.

While parameters for DBSCAN are *eps* that represent the distance between points that needed to state in one plane and *min_points* represent the minimum number of points in one plane. Each parameter of RANSAC and DBSCAN need to configure first and well-defined to acquire accurate segmentation result. In this research, parameters for RANSAC are *distance_threshold* 0.1, *ransac_n* 3, and *num_iteration* 1000. The distance between points can be measured, interpolated, or known from scan sampling resolution. The number of minimum points used to make a plane is 3 because it is enough points to make a plane. While the number of iterations can be acquired from the formula or calculated, also from trial and error. While parameters for DBSCAN are *d_threshold* 0.1 and *min_point* 10. The process is run iteratively in a specific range until all point cloud data in the area is segmented. The workflow algorithm for RANSAC and DBSCAN is shown in Figure 2.

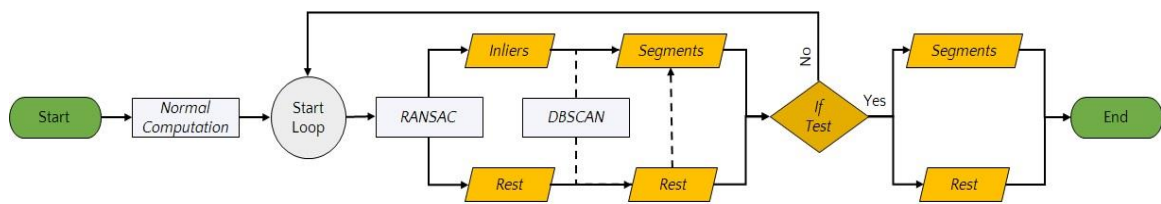


Figure 2. Algorithm workflow

Automatic segmentation is done using RANSAC and DBSCAN algorithms. RANSAC is a model-fitting algorithm where it fits point cloud data into a primitive shape such as a plane, sphere, and rectangle. In one iteration of the RANSAC algorithm it will segment one plane so to segment n plane inside a room, it needs n repeat of RANSAC. It is done using a looping algorithm inside the script (Figure 3). RANSAC is used to segment point clouds with repetition or in looping to acquire sufficient planes inside the room. Remains of the point cloud after the first looping is input to the next looping process. For one loop it will segment one plane, the rest of the unsegmented point cloud is used in the next loop to acquire another segment, and so on until n segment is received and the final rest of data is considered as outlier or noise in data. So, the result of RANSAC is inliers and outliers where inliers are the plane consisting of segmentation plane results while outliers are noise and point cloud data that are not fit with any plane.

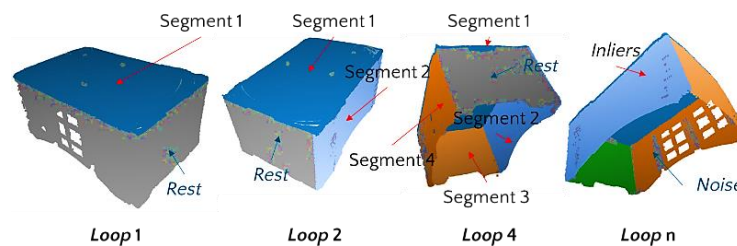


Figure 3. Segmentation process

After segmentation, there are quality control (QC) steps in aim to keep and ensure the quality of the segmented result is correct and accurate. There are two QC used in this research, geometric QC and attribute QC. Geometric QC is used to ensure geometric of the point cloud is accurate. It was checked with a comparison between real-world geometry and segmented point cloud geometry. The difference between them was then calculated to find root mean square error (RMSE). The formula of RMSE is shown in (1):

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\mathcal{Y}_i - y_i)^2}{n}} \quad (1)$$

Where \hat{y}_i represents segmented geometry, y_i represent ground truth geometry and n is number of samples. While the attribute QC is used to check the correctness of the attribute after segmentation. It was checked with a comparison between ground truth and segmented point cloud. The ground truth image as the comparison was taken as a photo, then the segmented point cloud was compared with the ground truth visually. The process is done using a confusion matrix to compare real attributes or the ground truth with the predicted attribute after segmentation. This research uses 50 samples distributed in five rooms for both QC.

3. RESULTS AND DISCUSSION

Point cloud processing aims to segment the point cloud automatically so as a result, it is grouped into five indoor classes. Segmentation is done using RANSAC and DBSCAN to segment all point cloud data in five rooms in the research location. The analyzing process aims to find the effectivity of both algorithms in huge and complex rooms, continued with quality control after segmentation.

3.1. Automatic segmentation result

The process of automatic segmentation from indoor point cloud data is done by using RANSAC and DBSCAN algorithms. Both algorithms have a parameter that needs to be defined before segmenting the point cloud data. After experimenting with the parameters, the *distance_threshold* parameter plays an important role in the process. The parameter is the reason to define which inliers, segmented plane, and outliers. The smaller the value of *distance_threshold*, the more outliers remain, because the required distance between points to form inliers becomes smaller. Conversely, the larger the *distance_threshold* value, it easier to form the segments, because the required distance between points to form inliers becomes larger, as shown in Figure 4. The use of this parameter then needs to be defined and configured precisely before doing the segmentation process.

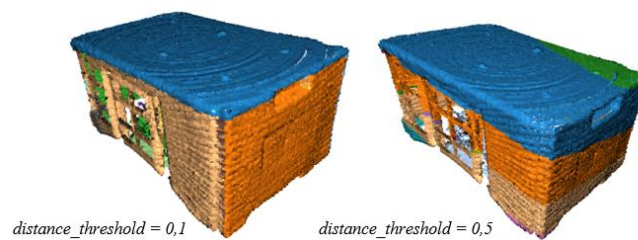


Figure 4. Effect of difference *distance_threshold* for segmentation result

For each RANSAC inlier or segmented plane, then DBSCAN will do the refinement by removing the lines that split planes creating a clear and smooth segmented result. However, it does not depend on training data due both of them are unsupervised algorithms, which gives advantages for fast processes. For each room, the process for RANSAC and DBSCAN to automatically segment the point cloud data is quite fast. It shows time efficiency for the process. RANSAC algorithm does not require complex optimization or high memory resources, which makes it efficient compared with other algorithms [11]. The use of RANSAC and DBSCAN is efficient in time to segment point cloud data, because of its simplicity but effective in determining the segments inside. Figure 5 shows the result of automatic segmentation using this method, where the left shows the ground truth and the right shows the predicted results.

The result of automatic segmentation is then saved into polygon file format (PLY) to be visualized. For each room, the number of segmented planes or inliers varies from six to eight segments, representing the number of classes also noises or outliers that are excluded from any classes. The segmentation process is done room per room due to the different sizes and complexity of each room. The result shows that the simpler room is more accurately segmented compared with complex rooms. For classes after segmentation, an indoor structure such as a wall and ceiling gets predicted well compared with other classes. The reason for this is that the RANSAC works on fitting the point cloud data into a specific geometry, while for walls and ceilings, it is easier to segment due to their shape being flat already. The result is accordance with Sanchez and Zakhor [27] that RANSAC works more accurately for indoor structural. However, wrong-defined parameters still create errors in segmentation.

The use of DBSCAN to group the inliers detected into a segmented point cloud has been proven effective and efficient due to fast and accurate segmentation results. The use of DBSCAN makes the user not need to explicitly define several segments, avoiding segmentation restricted or limited a k number of planes only. It differs from another segmentation or classification such as the K-means or k-nearest neighbors

(KNN) algorithm needs a k fix number of clusters to do the process. The use of DBSCAN in this research also overcomes the inconsistency density of point cloud acquired with low-cost TLS, by separating dense regions of point cloud data with low density of noise and outliers. While KNN probably may struggle to handle different densities in point cloud data. In addition, the point cloud data has suffered from noise due to being scanned with low-cost instruments, where the DBSCAN is suitable due to its robustness to detect and determine noises.

With the advantage of time processing or efficiency, the method to use RANSAC and DBSCAN for point cloud segmentation can be used for several indoor applications such as the preliminary indoor model for indoor navigation. The unsupervised method gives this method the advantage of depending on training data. It is a robust algorithm and particularly useful when dealing with noisy data such as point clouds. However, the parameters need to be well configured before starting the segmentation process. It also needs another advanced algorithm whenever need to complex geometric indoor structures.



Figure 5. Automatic segmentation result using RANSAC and DBSCAN

3.2. Segmentation for large and complex rooms

Segmentation for a large room is done by splitting the room into several parts first and automatically segmenting each part. The reason to split the large room is its limited ability of used machine learning algorithms to directly segment large rooms. If RANSAC and DBSCAN algorithms are used to segment large rooms directly, the segmentation result is not correct or does not represent real plane indoor. Segmentation results in a large room directly making segmented planes into many clusters. An example is ground truth is one wall but predicted as several walls. This error of segmentation was caused by the distance_threshold parameter in RANSAC that was not represented correctly due large room, causing a false segmentation result. The solution is to divide the large room into several parts and each of them is segmented automatically.

From segmentation results in the large rooms show that the algorithms used in this research, RANSAC, and DBSCAN are ineffective to segment point clouds in a large room. The strategy is to pre-process the point cloud data by dividing it into smaller parts. For each part then segmentation, but this strategy needs more time due to taking longer pre-processing time to split and divide the data into a smaller area. This means inefficient time to use the algorithms for large rooms.

Besides the size of the room, the complexity factor or level of detail also affects the segmentation result, as shown in Figure 6. This research uses five rooms where each of which represents a different detail. Level of detail (LoD) is commonly used to represent different details of buildings where LoD1 represents the simplest one and otherwise LoD5 represents the complex room. However, LoD terms in this research do not relate to CityGML LoD, because it does not acquire outdoor or outside parts of the rooms. The LoD used means different complexity of the rooms inside, aim to simplify analysis process of finding effectivity of machine learning algorithm from simple to detailed rooms.

LoD1 in this research is represented by a small rectangular room, filled with a little interior of chairs and tables. LoD2 is represented by a combine shape room, a combination of rectangle and trapezium also filled with a table and chair. LoD3 is represented by a corridor or long room, a combination of many shapes but there is no interior inside. LoD4 is represented by a lobby that is large and full of the interior. LoD5 also has huge geometry with many columns spread across the rooms and interiors. For LoD1 until LoD3, the algorithms successfully segment the point cloud, separate ceiling, wall, floor, and interior, meanwhile, it is

inaccurate to segment for LoD4 and LoD5. RANSAC and DBSCAN successfully segment simple LoD and inaccurate for complex LoD. This algorithm has limitations that cannot be used to segment complex rooms [1], [11], [27]. The solution proposed from this research is to split or divide the rooms into smaller parts and do segmentation for each part. The purpose of dividing the complex rooms is to simplify the rooms so algorithms can segment them correctly.

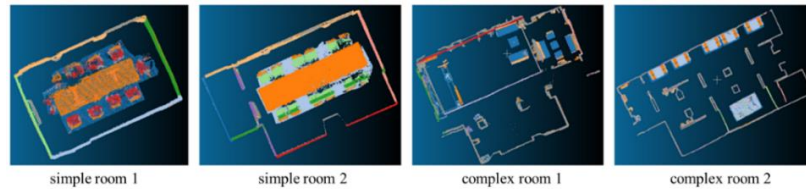


Figure 6. Segmentation result for simple and complex rooms

3.3. Quality control result

Geometric QC aims to ensure geometric of the segmented point cloud is accurate. The process of geometric QC is to compare real-world geometry such as the length and width of a room with model geometry. Real-world geometry was measured with measured tape while model geometry was measured with a ruler tool in CloudCompare. The difference between real-world geometry and model geometry is then calculated statistically to find the maximum, minimum, mean, standard deviation, and RMSE as shown in Table 1.

Table 1. Statistic of geometric quality control result

Statistic result	Value (cm)
Maximum difference	4.3
Minimum difference	0.1
Mean difference	2.3
Deviation difference	2.5
RMSE	2.5

Table 1 shows that the difference between real-world geometry and model geometry is very small. In other words, the model has accurate geometry. The maximum difference found in 50 samples of geometry is only 4 cm and the minimum is 0.1 cm. It shows very little difference between them. The mean, standard deviation and RMSE are relatively the same in approximation ± 2 cm. It shows the high accuracy of the model where the difference with real-world is in centimeter fractions. From this QC then concluded that the model has the correct scale, shape, and geometry. To elaborate, this geometric QC is heavily dependent on:

- Point cloud acquisition instrument used. The accuracy of this instrument is affecting the accuracy of point cloud results. Low-cost TLS used in this research has 1 cm accuracy.
- Georeferencing accuracy. This process also can create errors while georeferencing due to the process being done manually where an operator must select a tie point located in a dataset consisting of millions of points.
- Accuracy of geometric QC instrument, that is measured tape in this measured is 1 mm.
- Accuracy of model measurement in CloudCompare, because it is done manually to select which point to start and end of measured geometry.

Attribute QC aims to check correctness after the segmentation process. It is checked using a confusion matrix to compare ground truth and after segmentation. For this QC, 108 samples have been chosen from 5 rooms divided into 5 classes according to classes for segmentation. Matrix confusion can be calculated by comparing ground truth attribute and class after segmentation. Although it is attributed to QC, content that is checked in the model is whether the point cloud is segmented into different segments or not.

From attribute QC, it receives 80.56% overall accuracy where 87 samples are correct from 108 samples. It shows that the segmentation process generally is correct, and the point cloud data is well segmented. It also concludes that RANSAC and DBSCAN can effectively segment point clouds for indoor. Classes like ceiling have 100% user accuracy means all samples of those classes are correct. Producer accuracy of ceiling class is also high at 88%. Another accurate segment is wall and floor, which means for a simple object that has a primitive shape (plane) is accurately segmented. This is because the concept of RANSAC fits the point cloud into a plane model to segment.

Compared with another unsupervised algorithm, the use of RANSAC and DBSCAN in this research has proven more effective and efficient. For comparison, the research data was segmented with two other unsupervised

segmentation algorithms, K-means and spectral clustering. The K-means algorithm is simpler and easier to implement compared with DBSCAN. DBSCAN is more complicated than K-means [28], but it makes the DBSCAN can be used more powerful to segment point cloud data compared with K-means. Figure 7 shows a comparison of the segmentation results of this research algorithm with K-means showing the inability of K-means to segment indoor elements such as walls. The walls of the room are segmented into two or more clusters with K-means, while DBSCAN successfully groups the point cloud in one wall into one group cluster.

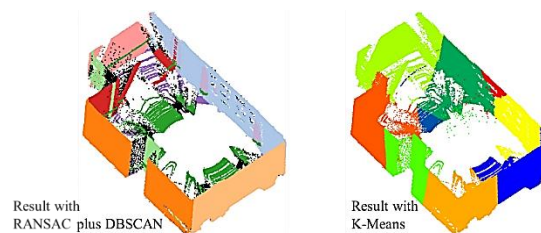


Figure 7. Segmentation results with RANSAC plus DBSCAN and K-means

On the other side, the use of spectral clustering is not efficient due to the time processes taking a very long time, up to minutes while the use of DBSCAN works in seconds. Segmentation results with the Spectral Clustering algorithm also were not accurate as expected, showing RANSAC and DBSCAN used in this research outperforms both popular unsupervised algorithms. The use of RANSAC and DBSCAN is effective in this research due to their robustness to noise and outliers, remembering the point cloud data used in this research was acquired using low-cost TLS which has many noises and density inconsistencies in the dataset.

4. CONCLUSION

Based on research and processing data, point cloud acquired using low-cost TLS has several limitations such as diverse pattern and distribution, also has a lot of noise. For that, the point cloud needs to be well pre-processed before being used as input for the automatic segmentation process. This become one of the disadvantages of the automatic process where the pre-processing stage is needing more time to prepare the data. From the research to automate point cloud segmentation using RANSAC and DBSCAN algorithms, it is known that both algorithms can have maximum use to segment small and less detailed indoor point clouds. While for large and complex indoor point cloud data, it still has issues with false segmentation. The strategy is to divide the large room into smaller parts and segment per partition. From control quality results show high accuracy, both for geometric accuracy and attribute accuracy after segmentation showing that this machine learning algorithm is effective in several use or condition for point cloud automatic segmentation.

ACKNOWLEDGEMENTS

This research can be done with the help of Dr. Catur Aries Rokhmana who provides the low-cost TLS instrument and its equipment. The authors also thank to Universitas Gadjah Mada for providing a place and funding for this research.




REFERENCES

- [1] R. R.-Jarén and J. J. Arranz, "Automatic segmentation and classification of BIM elements from point clouds," *Automation in Construction*, vol. 124, Apr. 2021, doi: 10.1016/j.autcon.2021.103576.
- [2] H. Macher, T. Landes, and P. Grussenmeyer, "From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings," *Applied Sciences*, vol. 7, no. 10, pp. 1–30, Oct. 2017, doi: 10.3390/app7101030.
- [3] A. Borrmann, M. König, C. Koch, and J. Beetz, "Building information modeling: Technology foundations and industry practice," *Building Information Modeling: Technology Foundations and Industry Practice*, no. September, pp. 1–584, 2018, doi: 10.1007/978-3-319-92862-3.
- [4] T. Luhmann, S. Robson, S. Kyle, and I. Harley, *Close Range Photogrammetry Principles, techniques and applications*. Scotland, UK, 2011, doi: 10.1111/phor.12114.
- [5] W. Linder, *Digital photogrammetry: A practical course, fourth edition*. 2016, doi: 10.1007/978-3-662-50463-5.
- [6] T. Luhmann, S. Robson, S. Kyle, and J. Boehm, *Close-Range Photogrammetry and 3D Imaging*, 2nd editio. Jerman: De Gruyter, 2019, doi: 10.1515/9783110607253.
- [7] L. S. Runceanu, S. Becker, N. Haala, and D. Fritsch, "Indoor Point Cloud Segmentation for Automatic Object Interpretation," *Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V.*, pp. 147–159, 2017.




- [8] E. Grilli, F. Poux, and F. Remondino, "Unsupervised object-based clustering in support of supervised point-based 3D point cloud classification," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 43, no. B2-2021, pp. 471–478, 2021, doi: 10.5194/isprs-archives-XLIII-B2-2021-471-2021.
- [9] F. Poux, C. Mattes, and L. Kobbelt, "Unsupervised Segmentation of Indoor 3D Point Cloud: Application to Object-Based Classification," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 44, no. 4/W1, pp. 111–118, 2020, doi: 10.5194/isprs-archives-XLIV-4-W1-2020-111-2020.
- [10] S. A. Bello, S. Yu, C. Wang, J. M. Adam, and J. Li, "Review: Deep Learning on 3D Point Clouds," *Remote Sensing*, vol. 12, no. 11, pp. 1–34, May 2020, doi: 10.3390/rs12111729.
- [11] Y. Xie, J. Tian, and X. X. Zhu, "Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation," *arXiv*, Aug., 2019.
- [12] C. L. Kang, F. Wang, M. M. Zong, Y. Cheng, and T. N. Lu, "Research on Improved Region Growing point Cloud Algorithm," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 42, no. 3/W10, pp. 153–157, 2020, doi: 10.5194/isprs-archives-XLII-3-W10-153-2020.
- [13] E. Grilli, E. M. Farella, A. Torresani, and F. Remondino, "Geometric Features Analysis For The Classification Of Cultural Heritage Point Clouds," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 42, no. 2/W15, pp. 541–548, 2019, doi: 10.5194/isprs-archives-XLII-2-W15-541-2019.
- [14] D. Xue, Y. Cheng, X. Shi, Y. Fei, and P. Wen, "An Improved Random Forest Model Applied to Point Cloud Classification," in *IOP Conference Series: Materials Science and Engineering*, Mar. 2020, vol. 768, no. 7, pp. 1–7, doi: 10.1088/1757-899X/768/7/072037.
- [15] X. F. Xing, M. A. Mostafavi, G. Edwards, and N. Sabo, "An improved automatic pointwise semantic segmentation of a 3D urban scene from mobile terrestrial and airborne LiDAR point clouds: A machine learning approach," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, no. 4/W8, pp. 139–146, 2019, doi: 10.5194/isprs-annals-IV-4-W8-139-2019.
- [16] H. Ni, X. Lin, and J. Zhang, "Classification of ALS Point Cloud with Improved Point Cloud Segmentation and Random Forests," *Remote Sensing*, vol. 9, no. 3, pp. 1–34, Mar. 2017, doi: 10.3390/rs9030288.
- [17] S. Liu, M. Zhang, P. Kadam, and C.-C. J. Kuo, *3D point cloud analysis: traditional, deep learning, and explainable machine learning methods*. Switzerland: Springer International Publishing, 2021.
- [18] M. Zuliani, *RANSAC for Dummies*. Santa Barbara: Vision Research Lab, 2012.
- [19] A. Adam, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "H-RANSAC: A Hybrid Point Cloud Segmentation Combining 2D and 3D Data," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, no. 2, pp. 1–8, 2018, doi: 10.5194/isprs-annals-IV-2-1-2018.
- [20] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, and L. Tang, "An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells," *Remote Sensing*, vol. 9, no. 5, pp. 1–16, May 2017, doi: 10.3390/rs9050433.
- [21] S. N. Mohd Isa *et al.*, "Point Cloud Data Segmentation Using RANSAC and Localization," *IOP Conference Series: Materials Science and Engineering*, vol. 705, no. 1, pp. 1–9, Nov. 2019, doi: 10.1088/1757-899X/705/1/012004.
- [22] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python and Scikit-Learn*. 2015.
- [23] C. Wang, M. Ji, J. Wang, W. Wen, T. Li, and Y. Sun, "An Improved DBSCAN Method for LiDAR Data Segmentation with Automatic Eps Estimation," *Sensors*, vol. 19, no. 1, pp. 1–26, Jan. 2019, doi: 10.3390/s19010172.
- [24] T. Czerniawski, B. Sankaran, M. Nahangi, C. Haas, and F. Leite, "6D DBSCAN-based segmentation of building point clouds for planar object classification," *Automation in Construction*, vol. 88, pp. 44–58, Apr. 2018, doi: 10.1016/j.autcon.2017.12.029.
- [25] J. Zhao *et al.*, "An Automatic Density Clustering Segmentation Method for Laser Scanning Point Cloud Data of Buildings," *Mathematical Problems in Engineering*, vol. 2019, pp. 1–13, Jul. 2019, doi: 10.1155/2019/3026758.
- [26] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A Modern Library for 3D Data Processing," *arXiv-Computer Science*, pp. 1–6, 2018.
- [27] V. Sanchez and A. Zakhor, "Planar 3D modeling of building interiors from point cloud data," *Proceedings - International Conference on Image Processing, ICIP*, pp. 1777–1780, 2012, doi: 10.1109/ICIP.2012.6467225.
- [28] Y. Chen, W. Ruys, and G. Biros, "KNN-DBSCAN: a DBSCAN in high dimensions," *arXiv-Computer Science*, pp. 1–20, 2020, doi: 10.48550/arXiv.1801.09847.

BIOGRAPHIES OF AUTHORS



Harintaka    received the Dr. degree (Doctor) with distinction in photogrammetry from Universitas Gadjah Mada in 2012, with the Dissertation "The Development of Digital Photogrammetric Processing of Small Format Aerial Photography for Spatial Data Provision". He is an Assoc.Professor of Photogrammetry in the Department of Geodetic Engineering, Universitas Gadjah Mada since 2007. In addition, he is serving as Head of Laboratory of Photogrammetry and Remote Sensing (2022–2025). His research interests are in photogrammetry, digital image processing, and computer vision. He can be contacted at email: harintaka@ugm.ac.id.



Calvin Wijaya    received the S.T. degree in Geodetic Engineering from the Universitas Gadjah Mada, Yogyakarta, in 2022, and now continuing the master's degree in Geomatic Engineering at the same university. His current research interests include LiDAR, 3D modeling, and geospatial data sciences. He can be contacted at email: calvin.wijaya@mail.ugm.ac.id.