

Three-dimensional point cloud segmentation using a combination of RANSAC and clustering methods

Puyam S. Singh^{1,*}, Iainehborlang M. Nongsiej², Valarie Marboh²,
Dibyajyoti Chutia¹, Victor Saikhom¹ and S. P. Aggarwal¹

¹North Eastern Space Applications Centre, Department of Space, Government of India, Umiam 793 103, India

²Department of Computer Science, St Anthony's College, Shillong 793 001, India

There are challenges in performing 3D scene understanding on point clouds derived from drone images as these data are highly unstructured with no neighbouring information, highly redundant making the processing difficult and time-consuming and have variable density making it difficult to group and segment them. For proper scene understanding, these point clouds need to be segmented and classified into different groups representing similar characteristics. The approaches for segmentation differ based on the distinctiveness of each data product. Although newer machine learning-based approaches work well, they need large amounts of standardized labelled data which in turn require extensive resources and human intervention to obtain good results. Considering these, we have proposed a hybrid clustering-based hierarchical model for effective segmentation of dense 3D point cloud. We have applied the model to local data having a mix of man-made and natural vegetation with variable topography. The combination of RANSAC, DBSCAN and Euclidean method of cluster extraction proved to be useful for precise segmentation and classification of point clouds. The performance of the model has been assessed using Davies–Bouldin dbIndex-based intrinsic measures. The hybrid approach is able to segment 91% of the point clouds precisely compared to the conventional one-step clustering approach.

Keywords: Clustering, drone images, hierarchical model, three-dimensional point cloud, segmentation.

POINT clouds represent world objects in a three-dimensional space. Each of these points has X , Y and Z coordinates. Point clouds derived using image-based models will have three additional colour attributes for R, G and B for every point. The drone imaging system can be deployed to rapidly collect the overlapping images and use them to generate a sparse point cloud initially. This is done through a well-defined process of feature detection and feature matching from closely overlapping images, followed by triangulation and bundle adjustment. It is called structure from motion, which means reconstructing a structure or scene from mov-

ing cameras capturing multiple overlapped photographs. The sparse point cloud is further densified using multi-view-stereo that uses a point densification algorithm and generates a high-quality, dense three-dimensional point cloud. These 3D point clouds are highly unstructured and unordered and do not have neighbouring information or scan position and direction. Based on the surface features and topography, the points will also show diverse variations in density across the objects in the scene. Further, the 3D points have limited attributes with no classification information. All these characteristics make the segmentation task challenging. The ability to accurately segment and classify these point clouds should benefit many real-world applications from general geospatial analysis to sophisticated vision-based applications such as robotics.

Considering the nature of 3D point attributes and their structure, different methods can be used for segmentation. Edge-based methods may be used for segmenting regions based on the object boundaries. These methods fail when the scene contains objects with arbitrary shapes. The region-based methods can be used to form different point clusters with similar properties. Here, the threshold parameter is important as it can affect the segmented regions. Segmentation based on attributes will depend on the computed value of the attributes and spatial relations of the points. The model-based segmentation methods rely on primitive geometric shapes such as sphere, cone, plane or cylinder to cluster and group. Model-based segmentation may work well for scenes comprising planar structures. The graph-based methods are used after converting the point cloud into a 3D graph. They are effective for application in areas such as robotic navigation. Despite having many techniques for segmenting point clouds, there are always challenges in using them for robust real-time applications. Further, techniques such as newer machine learning and deep learning require large amounts of labelled point data and high computing resources, and thus are expensive and time-consuming. The present study focuses on a combination of algorithms for clustering and the construction of a hierarchical model for clustering and segmentation on medium-sized point clouds consisting of non-uniform and complex features representing various land and natural resource features with good

*For correspondence. (e-mail: ss.puyam@nesac.gov.in)

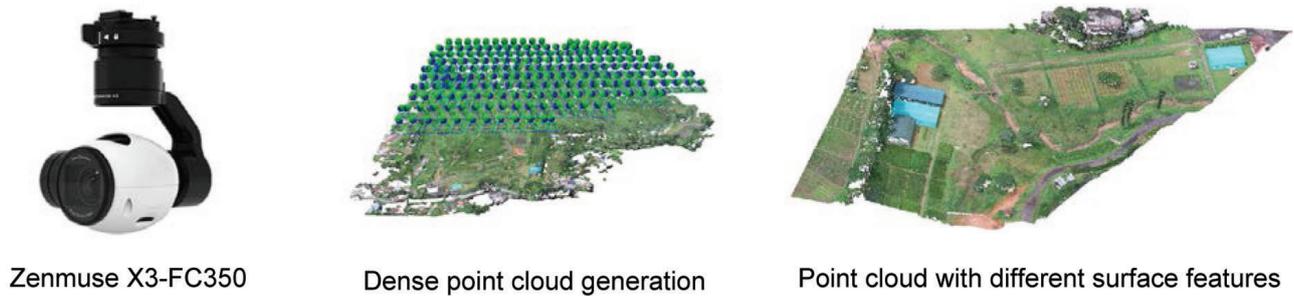


Figure 1. Data acquisition and generation of dense 3D point clouds.

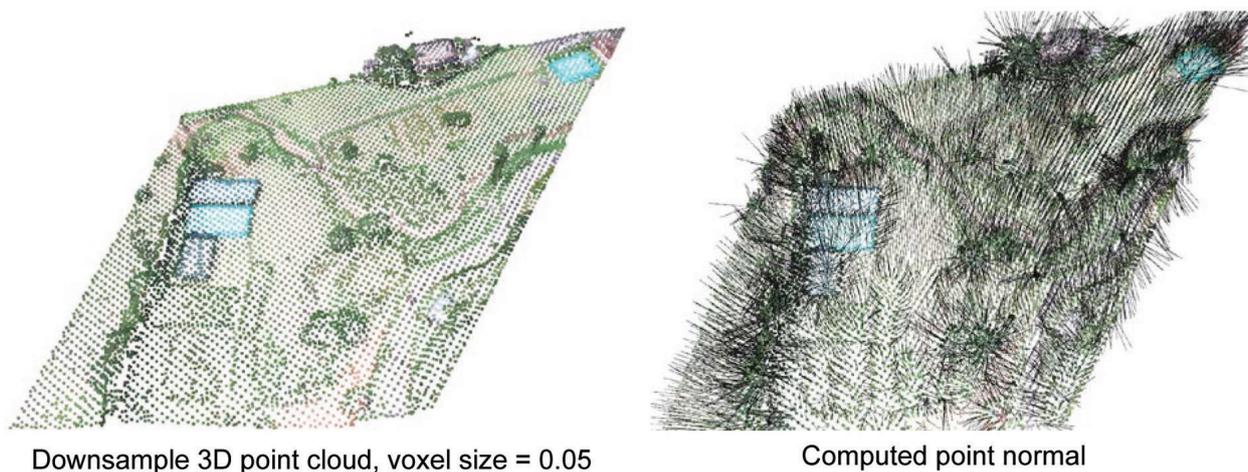


Figure 2. Estimation of surface normals in 3D point cloud.

performance. We propose a hybrid clustering process with model-fitting methods such as RANSAC in combination with a hierarchical-based clustering algorithm, which works well to detect shapes and segment the point clouds accurately.

Materials and methods

The camera of focal length of 4 mm (Zenmuse X3-FC350) having an effective resolution of 12.4 megapixels fitted to a drone (T600 DJI Inspire Series) was used to capture about 120 images, giving a maximum image size of 4000×3000 . The drone was flown at 100 m altitude and ensured 75% forward and slide overlapping. This helped us obtain repeated, robust feature points in each image pair and match them to generate a denser point cloud. The structure from motion in combination with multi-view stereo algorithm was used to generate highly dense 3D point clouds^{1,2}.

For the study, we considered a subset of data stored in a LAS file format which contained 738,583 points (Figure 1). The derived point cloud data were noisy, sparse and unorganized, and stored limited point attributes such as X , Y , Z and R , G , B . The sampling density of the points was also typically uneven due to varying linear and angular rates of the scanner. In addition, the surface shape was arbitrary

with sharp features and there was no statistical distribution pattern in the data.

To understand the structure of a point cloud and its orientation, a pre-processing step was carried out where point normals were estimated for each point after uniform down-sampling of the input point cloud with a voxel size of 5 cm (ref. 3). In the voxel grid geometry, 3D data are represented on a regular 3D grid. The colour and voxel value are calculated by averaging all the points within a voxel (Figure 2). Voxel normal estimation can be used to understand how each point cloud surface is oriented to obtain cues on the surface characteristics of objects. Planar surfaces will have their normal perpendicular to the surface, as opposed to other surfaces with different normal orientations. Such information can be used as an additional attribute to segment the point clouds more effectively⁴.

We can observe and differentiate different objects such as ground features with uniform orientation, linear features representing stream networks, artificial building structures with curvy roofs and natural vegetation with diverse point normal orientation, etc. Further, the varying density of point clouds representing different surface features can also be observed. Such preliminary information can assist while choosing different parameters for clustering algorithms.

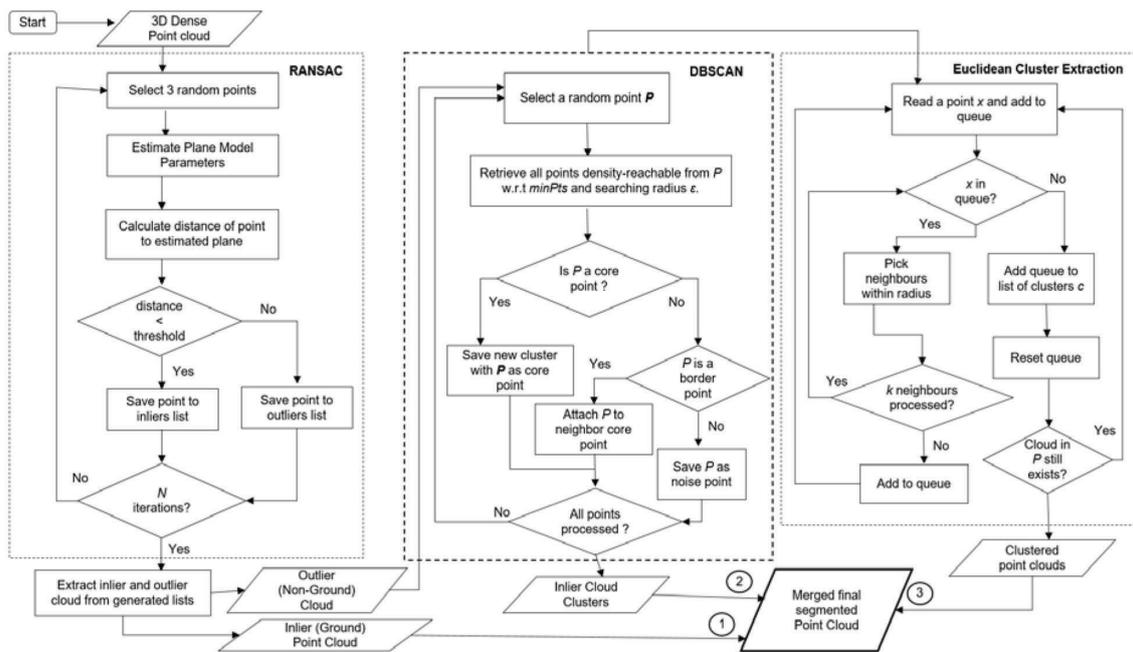


Figure 3. Proposed hierarchical-based 3D point cloud segmentation.

Considering the nature of point clouds having a mix of both planar man-made structures and vegetation structure, as well as non-uniform topography, a combination of model-based fitting algorithms such as RANSAC and clustering was adopted for grouping and labelling these points. This technique works effectively to separate outliers and noise present in the data⁵. RANSAC is good for detecting points having geometric primitives such as lines and circles present in the scene, but fails to group other unstructured point clouds representing trees and irregular structures⁶. Therefore, combining these two approaches can effectively segment all kinds of surface structures present within a point cloud.

The segmentation process began by first applying the RANSAC algorithm with a planar model and segmenting the points brought on the same plane in a global context of the point cloud. The rest of the point cloud, after subtracting the first output of RANSAC, was then fed to the DBSCAN clustering algorithm. This algorithm is good for clustering and grouping points with varied densities. The method is more effective for points with a greater altitude relative to all other points in the cloud and with each cluster of similar density. All such segmented point clouds were further fine-tuned by applying Euclidean cluster extraction algorithm. The segmented point clouds were separately merged to form the final segmented point cloud (Figure 3).

Results and discussion

Segmentation using RANSAC

RANSAC, a plane-fitting method, was chosen for its robust detection of planes in 3D point clouds and segmenting them

into inliers and outliers⁷. The inliers were labelled as ‘ground’ and they collected all the points with lower Z value under some threshold. The outliers were the ‘non-ground’ points fed into the next step in the segmentation process. The RANSAC method finds largest set of points that fits a plane under a given threshold. First, it randomly selects three points from the data to form a plane and calculates the parameters of the corresponding plane according to the plane equation $ax + by + cz + d = 0$, where x, y, z are the 3D coordinates. Given the known points, the constants a, b, c and d can be calculated. Using the final plane equation, the inlier 3D points are calculated at a threshold⁸. After a number of iterations, a threshold is selected to produce the maximum number of inlier ground points. This plane is then saved and segmented from the rest of the point cloud. The number of iterations n required to run the algorithm for finding the inliers can be calculated assuming the probability of success, p , where $n = \log(1 - p) / \log(1 - (1 - \epsilon)^s)$, for outlier ratio ϵ and sample size s (ref. 9). Based on the data, the plane equation for plane segmentation of inlier ground points was $-0.02x + 0.00y + 1.00z + -816.28 = 0$ with the distance threshold at 1.1.

The resultant segmentation produced a point cloud of 402,563 points containing the inliers, i.e. points lying on the same ground plane. We observed the efficacy of RANSAC, where all the planar points present in the point cloud lying on the same plane were segmented. This process was effective when segmenting almost all the ground points. The algorithm treated non-ground points as outliers containing the rest of the above-ground features such as buildings, high-slope areas, trees, etc. As shown in Figure 4, the output depicts the efficiency of RANSAC, as it can effectively

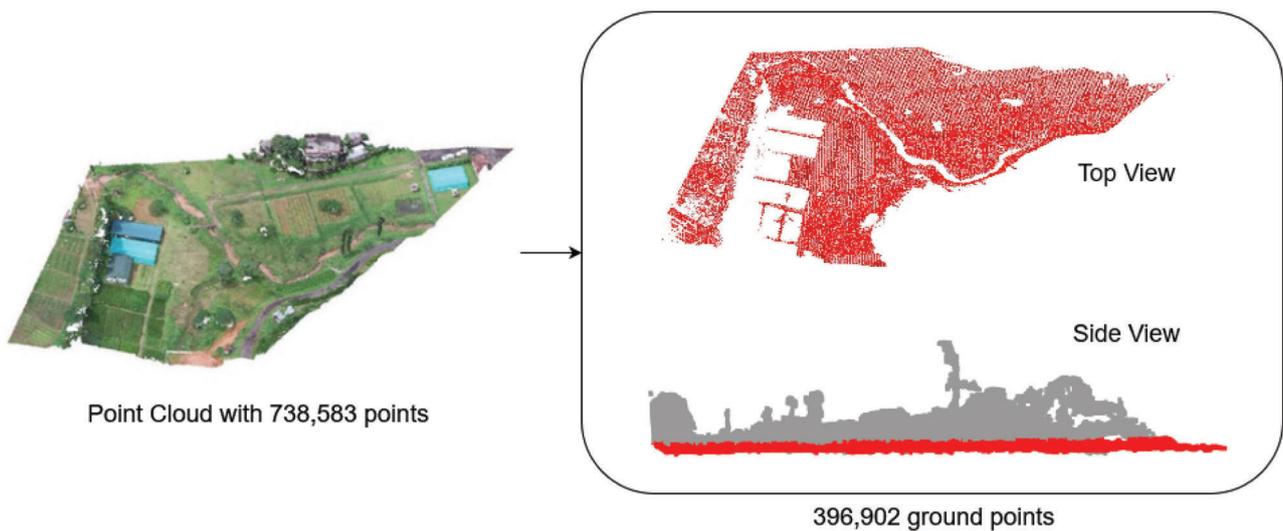


Figure 4. Segmentation using RANSAC.

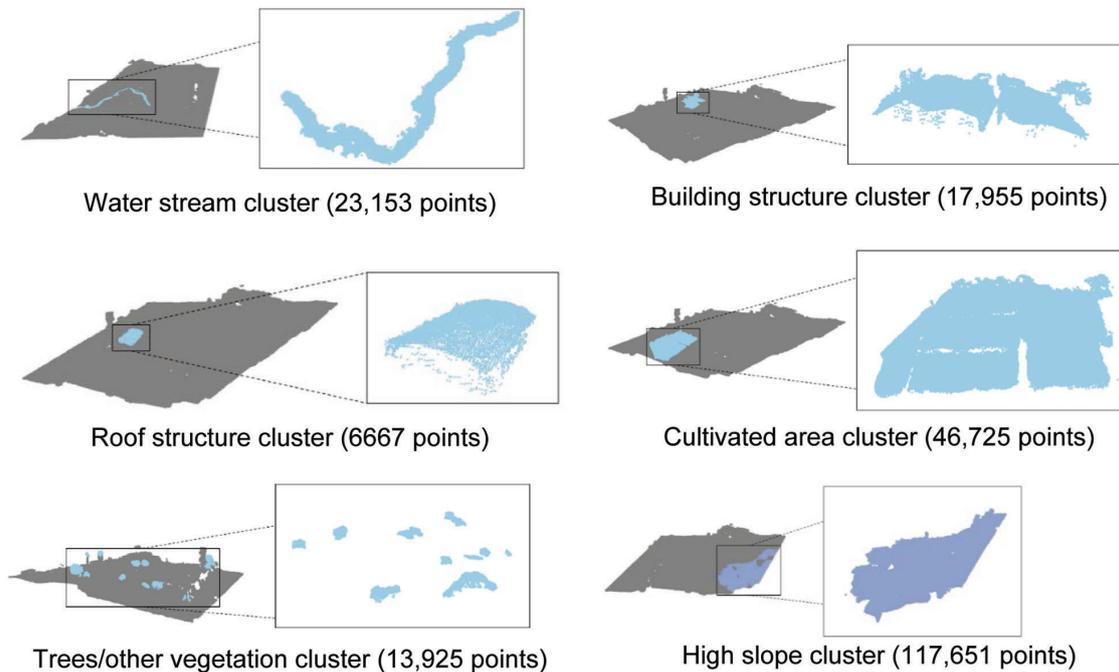


Figure 5. Detected clusters.

segment above-ground features. An advantage of RANSAC is its robust estimation of the model parameters, i.e. it can estimate the parameters with a high degree of accuracy even when a significant number of outliers is present in the point cloud¹⁰.

Segmentation using DBSCAN

Next, we considered the remaining 3D points with relatively higher Z values. The DBSCAN algorithm is used to create

multiple clusters of varying density^{11–13}. It needs at least two parameters: the minimum number of points $minPts$ and the searching radius ϵ . As our input 3D point cloud was relatively large with a natural scene of arbitrary shapes, $minPts$ was set to 500 to capture all the major clusters. ϵ defines the maximum searching radius between two points in a cluster. For calculating ϵ , mean distance of all points to all k (number of $minPts$) nearest neighbours was calculated. All the k th distances were then sorted in descending order and plotted on a k -dist graph^{14,15}. The desired value for ϵ will be at the first point in the valley of curvature. For our point

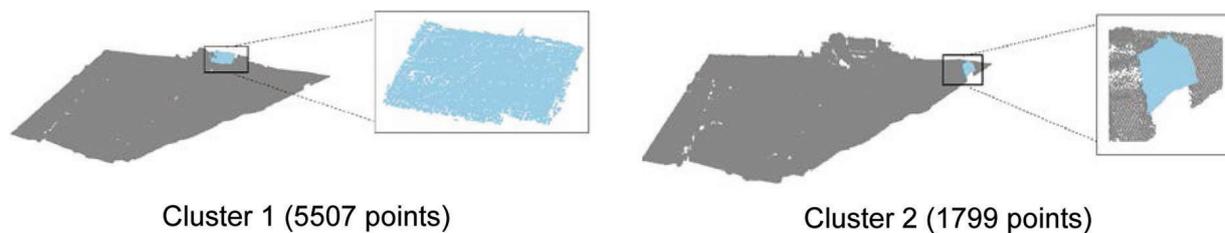


Figure 6. Roof structure clusters.

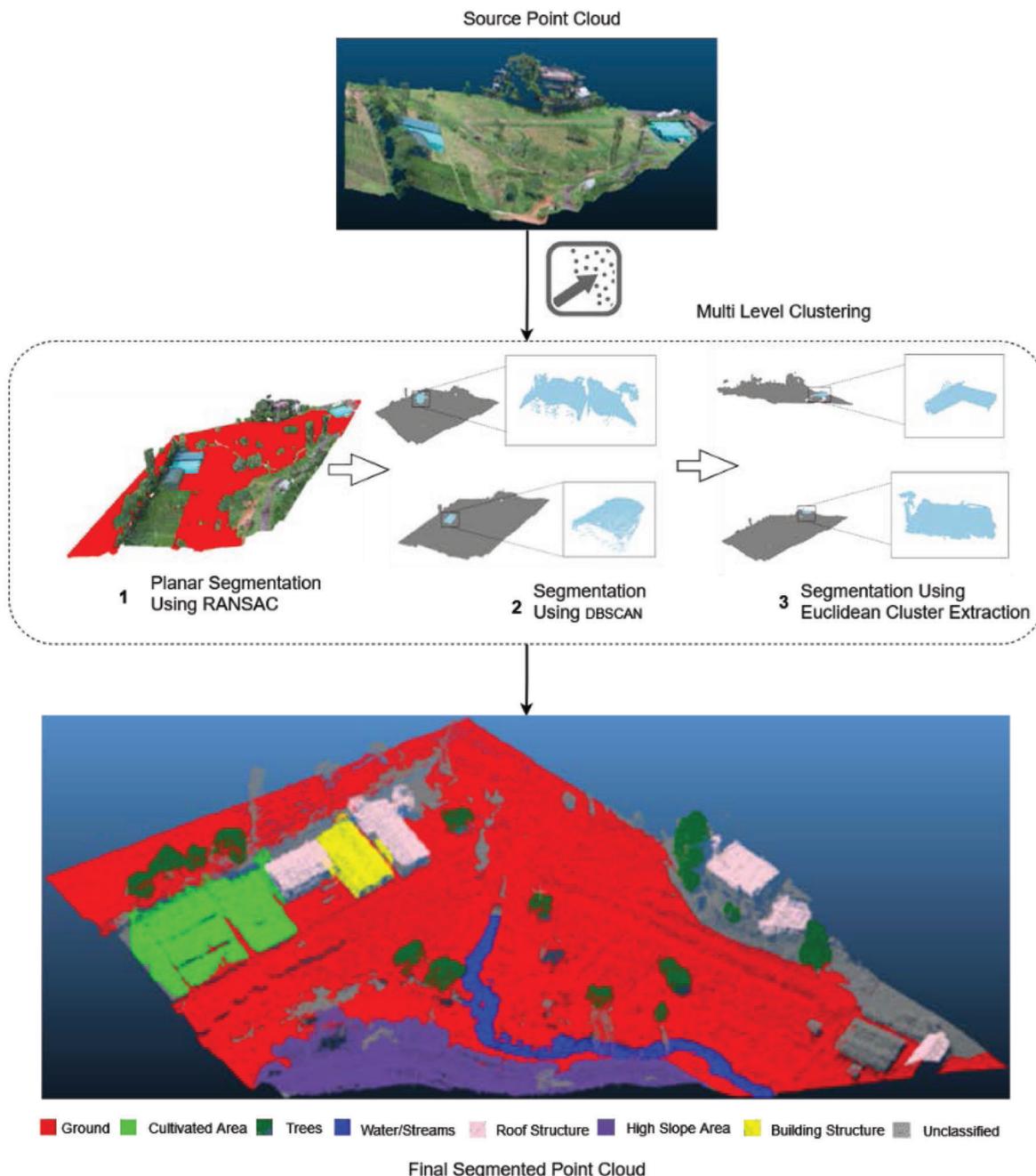


Figure 7. Methodology for point cloud segmentation.

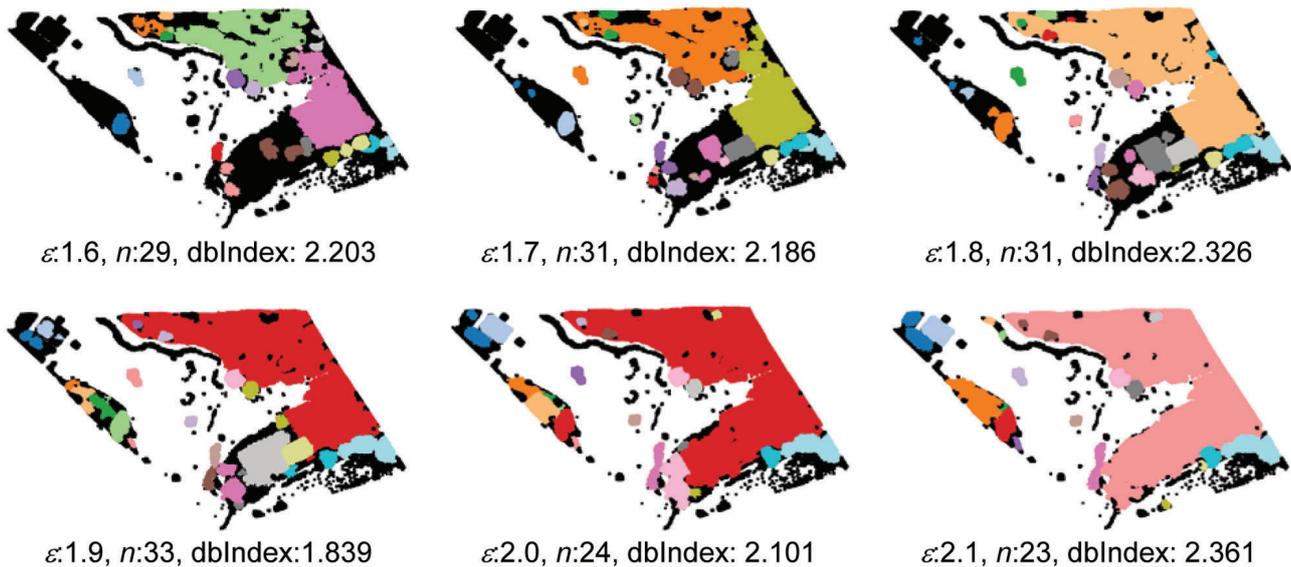


Figure 8. Relationship between dbIndex, searching radius ε and cluster number n .

Table 1. Segmented features

Segment	Number of 3D points
Ground	402,563
High-slope area	117,651
Noise/unclassified	85,178
Cultivated area	46,725
Water streams	23,153
Building structure	17,955
Roof structure	13,973
Trees	13,925

cloud, we used searching for a radius of 190 cm ($\varepsilon = 1.9$). Most patches of vegetation and trees were accurately clustered in our results, including buildings and other roof structures (Figure 5).

However, DBSCAN failed to cluster some object characteristics of trees, shrubs and patches of vegetation. The points representing roof-building structures and cultivation areas in the point cloud were also treated as noise. To identify and cluster these points, the third approach of segmenting using Euclidean cluster extraction was used.

Euclidean cluster extraction

The set of clusters which could not be detected by DBSCAN and was treated as noise can be connected in a given radius to form clusters using Euclidean cluster extraction. It works by finding the nearest neighbours of a point in the data. We set the distance threshold at 20 cm to find two more clusters (Figure 6).

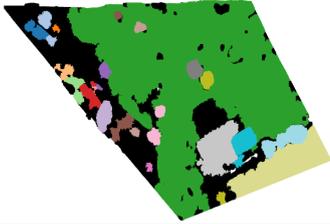
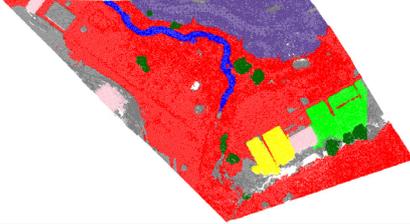
The combination of three different techniques in a hierarchical manner was able to segment the majority of the fea-

tures, although 11.8% of the point cloud could not be clustered or identified due to the non-uniform variation in point cloud density and highly unstructured points (Figure 7 and Table 1).

Performance measures of clustering algorithms

We can assess and analyse the performance of clustering algorithms. For any clustering-based algorithm, the validity of the clusters must be assessed so that they have sufficient inter-cluster distance and each point in the cluster is within the individual cluster radius and well localized. In case of DBSCAN, we adopted an intrinsic index-based measure of performance, where additional ground truths are not required. The Davies Bouldin index (dbIndex) is a popular clustering performance measuring technique. It works by dividing the clusters and finding their similarity by comparing the cluster distances and their sizes¹⁶. It finds the ratio of inter-clusters and intra-clusters. The dbIndex can be used to find the optimal cluster, where the lower value shows better clustering mechanism. Therefore, it can be used to cross-verify the cluster parameters used in the proposed approach¹⁷.

Table 2. Comparison table of DBSCAN and the proposed approach

Details	DBSCAN cluster extraction	The proposed Hierarchical-based clustering
dbIndex	1.839	1.76
Execution time (s)	189.210074	34.245472
Number of clusters	33	8
Output		

The dbIndex finds the average similarity between each cluster C_i for $i = 1, \dots, k$ and its most similar cluster C_j . Cluster similarity is defined as a measure R_{ij} .

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}, \quad (1)$$

where s_i is the average distance between each point of cluster i and the centroid of that cluster, and d_{ij} is the distance between cluster centroids i and j . dbIndex is then defined as

$$\text{dbIndex} = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} R_{ij}. \quad (2)$$

Figure 8 shows the influence of dbIndex on ε and n . Variable searching values were used, which gave different cluster outputs for the DBSCAN clustering algorithm and the dbIndex calculated. The outlier non-ground point cloud was considered to determine cluster density in the data. The lower index value gives the optimal number of clusters and searching radius. We observed and verified that 33 optimum clusters were formed with $\varepsilon = 1.9$, which gave a lower dbIndex among all configurations. The output with $\varepsilon = 1.9$ showed clusters of different objects with fewer noise data, such as building structures, water streams, tree patches, high-slope areas, etc. Colours were assigned arbitrarily to differentiate the individual clusters and do not necessarily represent unique object classes across these images (Figure 8).

Finally, the performance of the entire process of RANSAC, DBSCAN and Euclidean cluster evaluation was evaluated in terms of dbIndex, number of clusters and overall execution time, and compared with DBSCAN (only) used on the entire point cloud. The combination of RANSAC and DBSCAN algorithms in hierarchical approach has helped in separation of grounds and efficiently cluster the non-ground points as separate features in less time. This approach offers better performance in terms of unsupervised clustering and processing of point cloud data than DBSCAN alone (Table 2).

Conclusion

The combination of plane fitting and clustering algorithms in a hierarchical setting can effectively segment 3D point clouds with limited point attributes, and yield good results. RANSAC and DBSCAN work well for grouping and identifying objects of both planar as well arbitrary shape features from dense point clouds. Euclidean cluster extraction supplements the clustering results by its ability to cluster 3D points and locate additional objects. The performance of the clustering algorithm on the data was assessed using dbIndex. We found an effective searching radius of 1.9 for performing density-based clustering to find optimal clusters in order to capture all non-ground surface objects with low dbIndex. The proposed hierarchical-based approach resulted in the precise segmentation of 91% of point clouds giving eight well-defined clusters and reducing the overall execution time. Thus, this study highlights a simple yet effective hierarchical-based model architecture for better segmentation of highly unstructured, dense point clouds derived from drone imagery.

- Jiang, S., Jiang, C. and Jiang, W., Efficient structure from motion for large-scale UAV images: a review and a comparison of SfM tools. *ISPRS J. Photogramm. Remote Sensing*, 2020, **167**, 230–251; ISSN 0924-2716, <https://doi.org/10.1016/j.isprs.2020.04.016>.
- Leal-Alves, D. C. *et al.*, Digital elevation model generation using UAV-SfM photogrammetry techniques to map sea-level rise scenarios at Cassino Beach, Brazil. *SN Appl. Sci.*, 2020, **2**, 2181; <https://doi.org/10.1007/s42452-020-03936-z>.
- Dey, T. K., Li, G. and Sun, J., Normal estimation for point clouds: a comparison study for a Voronoi based method. In *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics*, Stony Brook, New York, USA, 2005, pp. 39–46; doi:10.1109/PBG.2005.194062.
- Zhao, R., Pang, M., Liu, C. and Zhang, Y., Robust normal estimation for 3D LiDAR point clouds in urban environments. *Sensors*, 2019, **19**, 1248; <https://doi.org/10.3390/s19051248>.
- Tarsha-Kurdi, F., Landes, T. and Grussenmeyer, P., Extended RANSAC algorithm for automatic detection of building roof planes from lidar data. *Photogramm. J. Finland*, 2008, **21**, 97–109.
- Fischler, M. A. and Bolles, R. C., Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981, **24**, 381–395.

7. Kurban, R., Skuka, F. and Bozpolat, H., Plane segmentation of Kinect point clouds using RANSAC. In 7th International Conference on Information Technology, Amman, Jordan, 2015.
8. Schnabel, R., Wahl, R. and Klein, R., Efficient RANSAC for point-cloud shape detection. *Comput. Graph. Forum*, 2007, **26**, 214–226.
9. Ruzgiene, B. and Förstner, W., RANSAC for outlier detection. *Geod. Kartogr.*, 2005, **31**(3), 83–87; doi:10.1080/13921541.2005.9636670.
10. Li, L., Yang, F., Zhu, H., Li, D., Li, Y. and Tang, L., An improved RANSAC for 3D point cloud plane segmentation based on normal distribution transformation cells. *Remote Sensing*, 2017, **9**, 433; <https://doi.org/10.3390/rs9050433>.
11. Wang, P., Gu, T., Sun, B., Huang, D. and Sun, K., Research on 3D point cloud data preprocessing and clustering algorithm of obstacles for intelligent vehicle. *World Electr. Veh. J.*, 2022, **13**, 130; <https://doi.org/10.3390/wevj13070130>.
12. Ahmed, S. M. and Chew, C. M., Density-based clustering for 3D object detection in point clouds. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 2020, pp. 10605–10614; doi:10.1109/CVPR42600.2020.01062.
13. Louhichi, S., Gzara, M. and Ben Abdallah, H., A density based algorithm for discovering clusters with varied density. In World Congress on Computer Applications and Information Systems, Hammamet, Tunisia, 2014, pp. 1–6; doi:10.1109/WCCAIS.2014.6916622.
14. Ester, M., Kriegel, H.-P., Sander, J. and Xu, X., A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, AAAI Press, Portland, Oregon, USA, 1996.
15. Davies, D. L. and Bouldin, D. W., A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, Pattern Analysis and Machine Intelligence-1, 1979, **2**, 224–227.
16. Maulik, U. and Bandyopadhyay, S., Performance evaluation of some clustering algorithms and validity indices. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002, **24**(12), 1650–1654; doi:10.1109/TPAMI.2002.1114856.
17. Wiroonsri, N., Clustering performance analysis using new correlation based cluster validity indices, 2021; arXiv preprint arXiv:2109.11172.

Received 12 July 2022; revised accepted 14 October 2022

doi: 10.18520/cs/v124/i4/434-441