Paper:

A Novel Recursive Non-Parametric DBSCAN Algorithm for 3D Data Analysis with an Application in Rockfall Detection

Pitisit Dillon[*1,*2,†], **Pakinee Aimmanee**[*1], **Akihiko Wakai**[*3], **Go Sato**[*4], **Hoang Viet Hung**[*5], **and Jessada Karnjana**[*2]

[*1] Sirindhorn International Institute of Technology, Thammasat University
Phahonyothin Road, Khlong Nueng, Khlong Luang, Pathum Thani 12120, Thailand
[†] Corresponding author, E-mail: m6222040278@g.siit.tu.ac.th
[*2] National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency, Pathum Thani, Thailand
[*3] Graduate School of Science and Technology, Gunma University, Gumma, Japan
[*4] Graduate School of Environmental Information, Teikyo Heisei University, Tokyo, Japan
[*5] Faculty of Civil Engineering, Thuyloi University of Vietnam, Hanoi, Vietnam
[Received November 30, 2020; accepted April 14, 2021]

The density-based spatial clustering of applications with noise (DBSCAN) algorithm is a well-known algorithm for spatial-clustering data point clouds. It can be applied to many applications, such as crack detection, rockfall detection, and glacier movement detection. Traditional DBSCAN requires two predefined parameters. Suitable values of these parameters depend upon the distribution of the input point cloud. Therefore, estimating these parameters is challenging. This paper proposed a new version of DBSCAN that can automatically customize the parameters. The proposed method consists of two processes: initial parameter estimation based on grid analysis and DBSCAN based on the divide-and-conquer (DC-DBSCAN) approach, which repeatedly performs DBSCAN on each cluster separately and recursively. To verify the proposed method, we applied it to a 3D point cloud dataset that was used to analyze rockfall events at the Puigcercos cliff, Spain. The total number of data points used in this study was 15,567. The experimental results show that the proposed method is better than the traditional DBSCAN in terms of purity and NMI scores. The purity scores of the proposed method and the traditional DBSCAN method were 96.22% and 91.09%, respectively. The NMI scores of the proposed method and the traditional DBSCAN method are 0.78 and 0.49, respectively. Also, it can detect events that traditional DBSCAN cannot detect.

Keywords: DBSCAN, divide and conquer, grid density, 3D point cloud, rockfall detection

1. Introduction

Density-based spatial clustering of applications with noise (DBSCAN), proposed in 1996, is a well-known clustering algorithm that labels a group of data based on density [1, 2]. This algorithm has been applied in many fields of study and has been improved for different objectives. For example, DBSCAN can be used to analyze 3D point clouds for rockfall detection [3, 4], glacier movement detection [5], and flight anomaly detection [6].

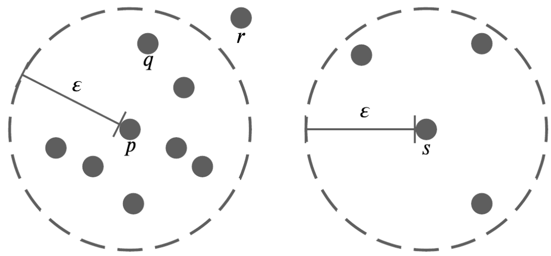


Fig. 1. Illustration of DBSCAN parameters.

The DBSCAN algorithm requires two parameters: ε and `minPts`, which substantially affect the clustering performance. The parameter ε is the largest Euclidean distance that allows two points to be in the same neighborhood.

The parameter `minPts` defines the minimum number of points in an ε-neighborhood circle centered at a point. A graphical representation of these two parameters is illustrated in **Fig. 1**. In this figure, points p and q are in the ε-neighborhood circle, while p and r are not. There are seven points in the same ε-neighborhood circle with p. If the number of points in the same ε-neighborhood with p is greater than or equal to `minPts`, we say that the points around p are dense. Otherwise, they are sparse. Suitable values of the parameters depend on the distribution of data. Thus, it is sometimes difficult to determine the appropriate parameters in advance.

In the literature, many parameter-estimation methods have been proposed. For example, Karami and Johansson used the differential evolution algorithm to find the optimal parameters of DBSCAN [7]. Even though the parameters obtained from the differential evolution work well, the cost function of the differential evolution needs knowledge about the ground truth. Thus, it might not be practical.

Darong and Peng proposed a grid-based DBSCAN algorithm, where the parameters were obtained from grid analysis [8]. Micheletti et al. used the *k*-nearest neighbors algorithm to estimate the parameters [5]. McInnes et al. proposed a new version of DBSCAN, called hierarchical DBSCAN or HDBSCAN [9]. HDBSCAN performs DBSCAN with many different ε values and then integrates the results to find the most stable clustering. However, it requires the smallest cluster size as one of its inputs. Lai et al. applied a new optimizer based on the multiverse optimization (MVO) theory to estimate the DBSCAN parameters [10]. The simulation results showed that the MVO could quickly find the parameters, and the clustering accuracy was high.

Once the parameters are obtained from the parameter estimation model, these methods apply the constant parameters to the input data. The constant-parameter-based DBSCAN might not cluster data points correctly if the cluster's density varies, as shown in Section 4. Therefore, this paper implements a new version of DBSCAN that can handle this case and customize the parameters automatically without a priori knowledge of the ground truth.

To verify the proposed method, we applied the proposed algorithm to a 3D point cloud dataset that was used to analyze and identify rockfall events at the Puiggcercos cliff, Catalonia, Spain. The result of this analysis can be used to construct a rockfall frequency map, which is crucial for hazard or risk assessment [11, 12].

The rest of this paper is organized as follows. Section 2 reviews the traditional DBSCAN algorithm. Section 3 proposes a new version of the DBSCAN algorithm, which is based on the divide-and-conquer approach. Section 4 provides the details of our experiments and results. The discussion is in Section 5, and Section 6 concludes this work.

2. DBSCAN

DBSCAN is a spatial clustering algorithm based on data point density [1]. The algorithm assigns a group of points that are close to each other to a cluster. In other words, the points in a high-density region are grouped and called a cluster. However, a point that is not a member of any cluster is assigned as noise.

Let p and q denote points in a point cloud $\boldsymbol{P}$, and let $d(p,q)$ be the Euclidean distance between points p and q. The set of points that have distances to point p that are less than or equal to a predefined distance ε is denoted by $N_\varepsilon(p)$, which is called the *ε-neighborhood* of point p. That is,

$$N_\varepsilon(p) = \{q \in \boldsymbol{P} \mid d(p,q) \leq \varepsilon\}. \quad \ldots\ldots \quad (1)$$

Let $|N_\varepsilon(p)|$ denote the number of points in the ε-neighborhood of point p, and let `minPts` be a predefined positive integer. If $|N_\varepsilon(p)| \geq$ `minPts`, point p is a *core point*. If $p \in N_\varepsilon(q)$ and q is a core point, point p is *directly density-reachable* for point q.

Point p is *density-reachable* from point q if there is a sequence of points (i.e., $q, p_1, p_2, \ldots, p_n, p$) that satisfies the following three conditions:

- p_{i+1} is *directly density-reachable* from p_i,
- p_1 is *directly density-reachable* from q, and
- p is *directly density-reachable* from p_n.

If p_i is between p and q and both p and q are density-reachable from p_i, point p is *density-connected* to point q.

Let $\boldsymbol{C}$ denote a cluster of $\boldsymbol{P}$. DBSCAN uses the following rules to cluster points in $\boldsymbol{P}$.

- If $p \in \boldsymbol{C}$ and q is *density-reachable* from p, then $q \in C$.
- For any p and q in $\boldsymbol{C}$, p is *density-connected* to q.

The DBSCAN algorithm discovers a cluster by choosing a *core point* and retrieving all points that are *density-reachable* from that core point [1]. Any point that is outside of all clusters is assigned as noise. The DBSCAN algorithm can be summarized by the following steps [2].

1 Find $N_\varepsilon(p)$ for all p in $\boldsymbol{P}$.

2 Identify core points.

3 Connect neighboring core points to form clusters.

4 Add non-core points to a nearby cluster if the cluster is an *ε-neighborhood*.

3. Proposed Method

The general idea behind our proposed method is that we first roughly estimate the DBSCAN's parameters. Then, the DBSCAN algorithm is repeatedly performed on each cluster separately until the number of points labeled as noise outnumbers the other non-noise groups. To understand the proposed method clearly, we first define some symbols unambiguously as follows. Let $\boldsymbol{P}^n$ be an $n \times 4$ matrix representing a point cloud of n data points obtained from a terrestrial laser scanner. Each data point is represented by a quadruple (x,y,z,I), where (x,y,z) is the coordinate of the data point in the Cartesian system, and I is the reflected laser intensity.

Let $\boldsymbol{P}^{n*}$ denote a point cloud with labeled clusters, which is obtained by applying the DBSCAN algorithm with the parameters `minPts` and ε to $\boldsymbol{P}^n$, i.e., $\boldsymbol{P}^{n*} =$ DBSCAN($\boldsymbol{P}^n$, `minPts`, ε). Thus, $\boldsymbol{P}^{n*}$ can be represented by an $n \times 5$ matrix, where each data point has an additional element that indicates a cluster ID. That is, one data point is represented by a quintuple (x,y,z,I,c), where c is the cluster ID. In general, $c \in \{0,1,2,\ldots\}$, and a data point that is labeled with $c = 0$ is recognized as noise in

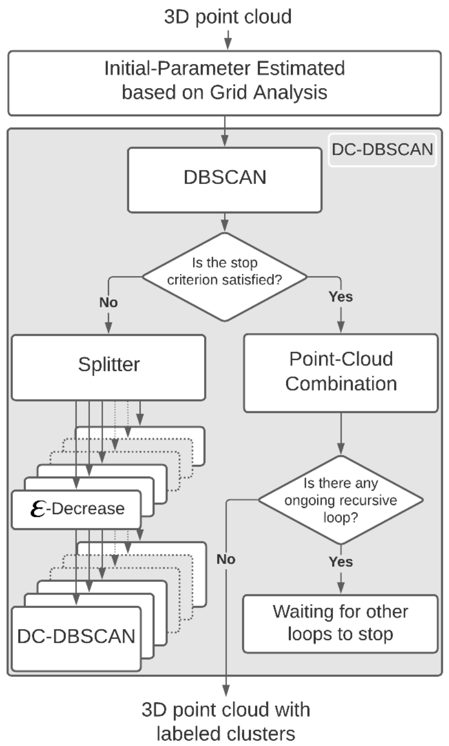


Fig. 2. Proposed method.

the DBSCAN algorithm [1]. We also denote the greatest cluster ID number of $\boldsymbol{P}^{n*}$ by $c_{\max}(\boldsymbol{P}^{n*})$.

Let $N_c(\boldsymbol{P}^{n*})$ denote the number of clusters of $\boldsymbol{P}^{n*}$, and let $\boldsymbol{P}_i^{n*}$ be an $n \times 4$ matrix representing a point cloud of all data points labeled with i, where $i \leq N_c(\boldsymbol{P}^{n*})$, and each point in $\boldsymbol{P}_i^{n*}$ is represented by a quadruple (x, y, z, I).

Let $N_p(\boldsymbol{P}_i^{n*})$ be the number of data points of $\boldsymbol{P}_i^{n*}$, i.e., $N_p(\boldsymbol{P}_i^{n*})$ is equal to the number of rows of $\boldsymbol{P}_i^{n*}$. Thus, it is straightforward to state that $\sum_{\forall i} N_p(\boldsymbol{P}_i^{n*}) = n$.

The proposed method consists of two processes, as illustrated in **Fig. 2**. First, 3D point cloud $\boldsymbol{P}^n$ is analyzed by grid analysis to estimate the appropriate initial parameters (`minPts` and ε) of the DBSCAN algorithm. Second, the divide-and-conquer-based DBSCAN (DC-DBSCAN) process takes $\boldsymbol{P}^n$, `minPts`, and ε as the inputs and returns the point cloud with labeled clusters as the output. This DC-DBSCAN process is based on the DBSCAN algorithm and a recursive step. The DBSCAN algorithm takes $\boldsymbol{P}^n$, `minPts`, and ε as the inputs and produces $\boldsymbol{P}^{n*}$ as the output. The recursive step takes $\boldsymbol{P}_{\boldsymbol{i}}^{n*}$, `minPts`, and ε as its inputs and returns the point cloud with labeled clusters as the output.

The DC-DBSCAN process operates as follows. Let $\boldsymbol{Q}^m$, where m is the total number of data points, $\texttt{minPts}_Q$, and ε_Q be the inputs of DC-DBSCAN. First, it performs DBSCAN($\boldsymbol{Q}^m$, $\texttt{minPts}_Q, \varepsilon_Q$) and obtains $\boldsymbol{Q}^{m*}$. Then, it checks that the stop criterion for $\boldsymbol{Q}^{m*}$, i.e., $N_p(\boldsymbol{Q}_0^{m*})$ is not the smallest among $N_p(\boldsymbol{Q}_i^{m*})$ for $i \neq 0$. In other words, the stop criterion is satisfied when the noise cluster is not the smallest cluster of $\boldsymbol{Q}^{m*}$. If it is (in this case), the point cloud obtained from DBSCAN($\boldsymbol{Q}^m$, $\texttt{minPts}_Q, \varepsilon_Q - \Delta$), where Δ is an ε-decrement step (which is normally set to 1), is a part of the resulting point cloud with labeled clusters.

In contrast, if $N_p(\boldsymbol{Q}_0^{m*})$ is the smallest, the point cloud $\boldsymbol{Q}^{m*}$ is split into $c_{\max}(\boldsymbol{Q}^{m*})$ point clouds: $\boldsymbol{Q}_1^{m*}$, $\boldsymbol{Q}_2^{m*}$, ..., and $\boldsymbol{Q}_{c_{\max}(\boldsymbol{Q}^{m*})}^{m*}$. Then, each $\boldsymbol{Q}_i^{m*}$ for $i = 1$ to $c_{\max}(\boldsymbol{Q}^{m*})$ is separately fed to DC-DBSCAN with the parameters $\texttt{minPts}_Q$ and ε_{Q_i}, where $\varepsilon_{Q_i} = \varepsilon_Q - \Delta$. These recursive loops are terminated because as the parameter ε gets smaller, the number of points assigned as noise increases. Consequently, the stop criterion is satisfied for all loops.

When all recursive loops stop repeating themselves, the combined point cloud with labeled clusters is returned by the proposed method.

Besides DC-DBSCAN, another crucial part of the proposed method is the initial-parameter estimation based on grid analysis. A flowchart of this process is sketched in **Fig. 3**, and it works as follows.

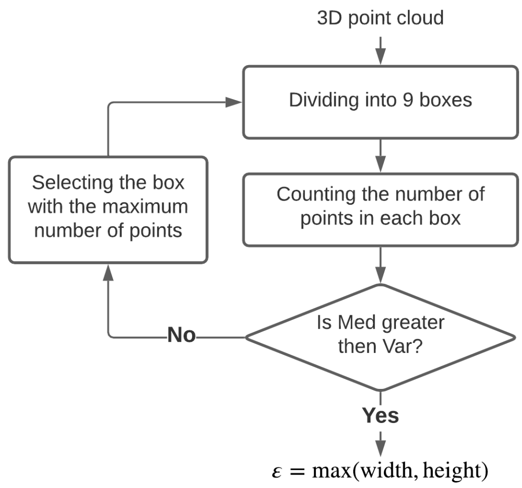


Fig. 3. Flowchart of the grid analysis.

First, the 3D point cloud $\boldsymbol{P}^n$ is spatially divided into nine equal rectangular boxes, as shown in **Fig. 4** (top). Second, if the variance of the number of points in those boxes is greater than the median of the number of points, the box with the maximum number of points is selected. These two steps are repeatedly performed on the designated box, as depicted in **Fig. 4** (bottom). When the loop is broken, ε is set to the maximum between the width and the height (in cm) of the last selected box.

4. Experiment and Result

To evaluate the performance of the proposed method, we conducted experiments with an open dataset obtained from a terrestrial laser scanner (Optech's Intelligent Laser Ranging and Imaging System or ILRIS3D), provided by Abellan et al. [3, 4, 13]. The dataset contains data points representing the 3D surface of the Puiggcercos cliff, Catalonia, Spain, as shown in **Fig. 5**. The maxi-

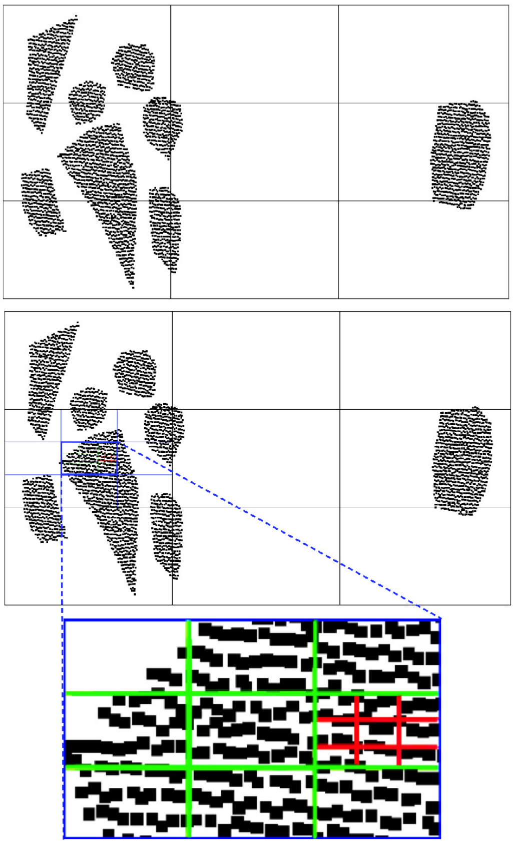


Fig. 4. Grid analysis: dividing the point cloud spatially into nine boxes (top) and dividing repeatedly for three times (bottom).

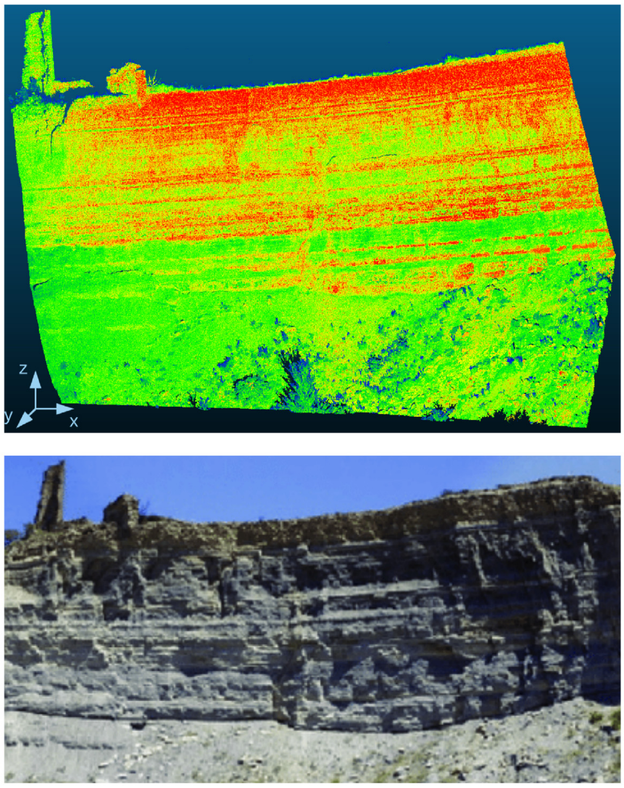


Fig. 5. 3D point cloud (top) of the surface of Puiggcercos cliff (bottom), Catalonia, Spain.

mum width and height of the cliff are 73.77 and 48.42 m, respectively. Analyzing two or more 3D point clouds recorded at different times shows the surface displacement. Thus, these point clouds can be applied for rockfall detection, rock-volume estimation, crack detection [4], and mass movement of a rock glacier [5]. In our experiments, we modified the dataset by using open-source 3D point cloud editing and processing software, namely CloudCompare [14], to generate four point-clouds that can be analyzed further for rockfall detection. The modified point-clouds used in our experiments are shown in **Fig. 6**, and each of them consists of eight rockfall events.

The dimensions of rockfall events from Event 1 to Event 8 in Scenario (a) are $(80,103)$, $(126,145)$, $(63,54)$, $(58,37)$, $(68,40)$, $(70,59)$, $(96,95)$, and $(50,75)$, respectively. The dimension of the rockfall is measured by the width and the height of the smallest rectangle that can fit it in. We use $(\text{width}, \text{height})$ (in cm) to denote the dimension of such a rectangle. An illustration example of a rockfall event $(85,40)$ is shown in **Fig. 7**. The point densities of these events are approximately equal. The events are easily clustered since they are not close to each other.

The dimensions of rockfall events from Event 1 to Event 8 in Scenario (b) are $(48,53)$, $(131,135)$, $(65,86)$, $(153,79)$, $(125,82)$, $(45,43)$, $(57,51)$, and $(88,32)$, respectively. The point densities of these events are approximately equal. The events are more difficult to cluster since they are close to each other.

The dimensions of rockfall events from Event 1 to Event 8 in Scenario (c) are $(68,23)$, $(49,67)$, $(73,79)$, $(85,40)$, $(100,56)$, $(91,42)$, $(131,135)$, and $(134,69)$, respectively. The point densities of these events are different. The events are easy to cluster since they are not close to each other.

The dimensions of rockfall events from Event 1 to Event 8 in Scenario (d) are $(87,55)$, $(34,70)$, $(127,59)$, $(190,138)$, $(71,67)$, $(85,66)$, $(128,62)$, and $(45,43)$, respectively. The point densities of these events are different. The events are more difficult to be clustered since they are close to each other.

As mentioned earlier, DBSCAN is used to recognize rockfall events [4]. However, the effectiveness of DBSCAN depends on two parameter values (i.e., `minPts` and ε) that should be appropriate for the input dataset. Our primary aim for the following experiments was to confirm that the proposed method can adaptively and automatically adjust the parameters according to the input dataset. Note that, in our experiments, noise (or clutter) was removed from the input point cloud during a preprocessing process using CloudCompare.

We evaluated the effectiveness of the proposed method by using two evaluation measures: *purity* and *normalized mutual information* (NMI). The purity is the ratio of the

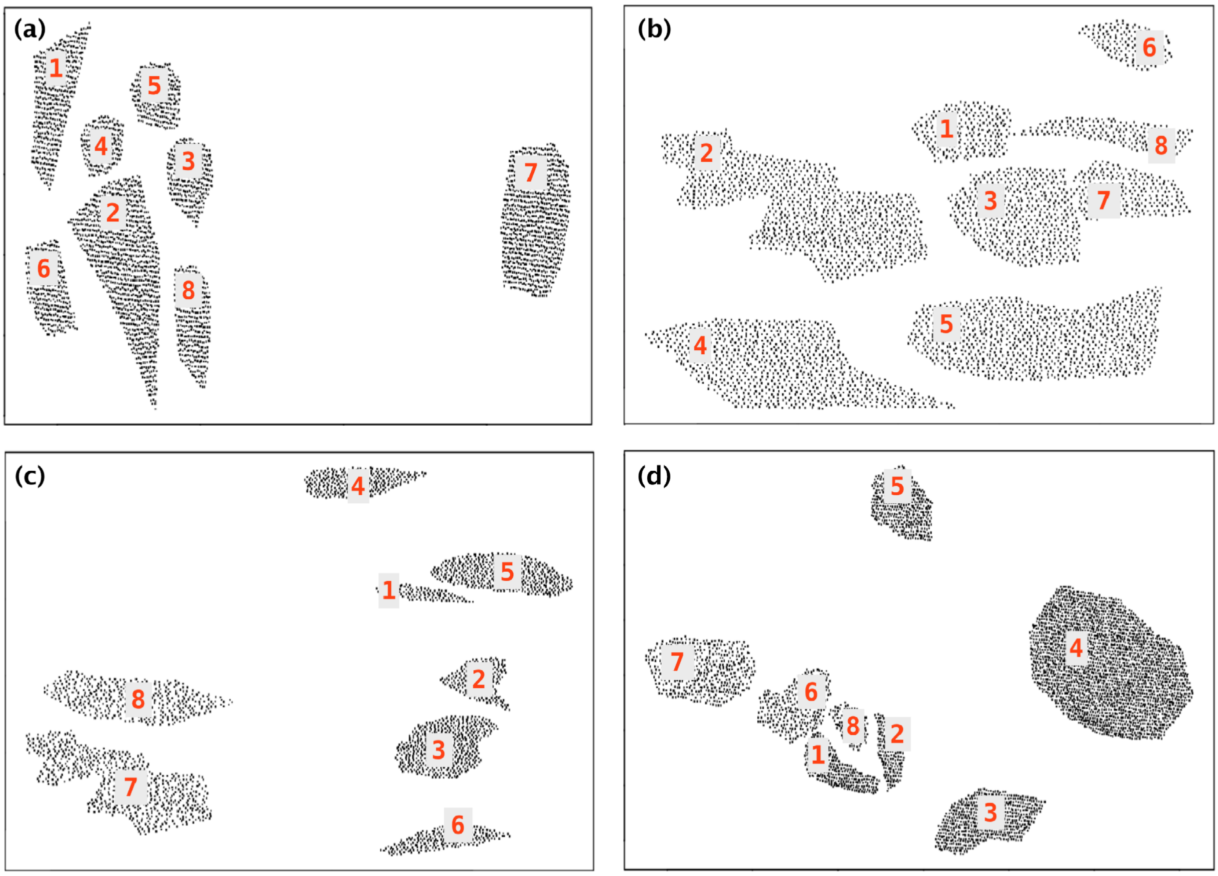


Fig. 6. Four point clouds from the larger point cloud in **Fig. 5**.

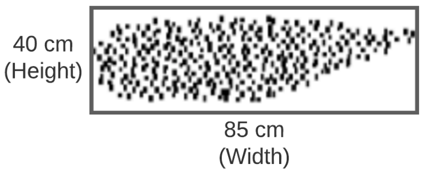


Fig. 7. Example of a rockfall event $(85, 40)$.

sum of all true positives to the total number of data points. The purity λ is defined as

$$\lambda = \frac{1}{n}\sum_{h=1}^{k} \max_{1\le j\le l} \{n_h^1, n_h^2, n_h^3, \ldots, n_h^j\}, \quad \ldots \ldots \quad (2)$$

where n is the total number of points, l is the total number of ground-truth clusters, k is the total number of clusters labeled by the algorithm, and n_h^j is the number of points that are labeled by the algorithm as cluster h and belong ground-truth cluster j. The higher the value of λ, the better the algorithm's effectiveness.

The NMI score is defined as follows [15, 16]. Let X be a discrete random variable representing a ground-truth label (or ground-truth cluster ID) of points in the point cloud. Given that the ground truth consists of $u+1$ clusters, the possible outcomes of X are $0, 1, \ldots, u$, where each outcome is a ground-truth cluster ID. Let Y be a discrete random variable representing a cluster ID that is labeled by the clustering algorithm. Given that the algorithm clusters the point cloud into $v+1$ clusters, the possible outcomes of Y are $0, 1, \ldots, v$, where each outcome of Y is a cluster ID. The NMI score $I^*(X;Y)$ is defined by

$$I^*(X;Y) = \frac{2 \cdot I(X;Y)}{H(X)+H(Y)}, \quad \ldots \ldots \ldots \quad (3)$$

where $I(X;Y)$ is the mutual information between X and Y, $H(X)$ is the entropy of X, and $H(Y)$ is the entropy of Y. These terms are defined as follows.

$$H(X) = -\sum_{i=0}^{u} p(X=i) \cdot \log(p(X=i)), \quad \ldots \quad (4)$$

and

$$p(X=i) = \frac{x_i}{n}, \quad \ldots \ldots \ldots \ldots \ldots \quad (5)$$

where x_i, for $i=0$ to u, is the total number of points in the i-th ground-truth cluster, and n is the total number of points in the point cloud.

$$H(Y) = -\sum_{i=0}^{v} p(Y=i) \cdot \log(p(Y=i)), \quad \ldots \quad (6)$$

Table 1. Evaluation comparison between the proposed method and DBSCAN [4] for Scenario (a).

Event number	ε [cm]	The number of points		
		Proposed method	DBSCAN [4] ($\varepsilon = 10$ cm)	Ground truth
1	14	488	488	488
2	6	1028	1228	1028
3	9	299	568	299
4	6	200	0	200
5	9	269	0	269
6	14	329	329	329
7	14	887	887	887
8	14	342	342	342
λ		100%	87.79%	100%
NMI score		1	0.44	1

Table 2. Evaluation comparison between the proposed method and DBSCAN [4] for Scenario (b).

Event number	ε [cm]	The number of points		
		Proposed method	DBSCAN [4] ($\varepsilon = 10$ cm)	Ground truth
1	3	248	449	247
2	6	1025	1025	1025
3	6	758	758	513
4	6	855	855	855
5	6	931	931	931
6	6	159	159	159
7	6	0	0	245
8	3	201	0	202
λ		94.11%	89.30%	100%
NMI score		0.67	0.31	1

and

$$p(Y=i) = \frac{y_i}{n}, \quad (7)$$

where y_i, for $i = 0$ to v, is the total number of points in the i-th cluster that is labeled by the algorithm.

$$I(X;Y) = H(X) - H(X|Y), \quad (8)$$

$$H(X|Y) = \sum_{j=0}^{v} -p(Y=j) \cdot H(X|Y=j), \quad (9)$$

and

$$H(X|Y=j) = \sum_{i=0}^{u} p(X=i|Y=j) \cdot \log\left(p(X=i|Y=j)\right), \quad (10)$$

where $p(X=i|Y=j)$ is a probability that $X=i$ given that $Y=j$, for $i=0$ to u and $j=0$ to v. The value of $I^*(X;Y)$ is in the interval $[0,1]$, and it equals 1 if and only if all ground-truth clusters and clusters labeled by the algorithm exactly coincide.

In this work, we compare the proposed method, where `minPts` is a constant and is set to 5, to traditional DBSCAN with the parameters `minPts` $= 5$ and $\varepsilon = 10$, as recommended by Tonini and Abellan [4]. The experimental evaluations for all four point clouds are shown in **Table 1** to **Table 4**. The first part of each table shows the numbers of clustered points for events in each Scenario. Its second column (ε) shows the ε values obtained from the proposed method. The second part of each table shows the purity score λ and the NMI score. It can be seen that, in most cases, the proposed method outperformed the traditional DBSCAN method [4].

The proposed method's average purity score was 96.22%, whereas that of traditional DBSCAN was 91.09%. The proposed method has higher purity scores in Scenarios (a), (b), and (d), but it has a slightly lower score in Scenario (c). We discuss these results in the next section.

An example of results is shown in **Fig. 8**, which is from Scenario (a) of **Fig. 6**. It can be seen that DBSCAN could not detect two rockfall events (i.e., Events 4 and 5), whereas the proposed method could detect all of them. This is because traditional DBSCAN uses a constant ε for one point cloud while the proposed method benefits from the adaptive ε, as shown in **Table 1**.

Table 3. Evaluation comparison between the proposed method and DBSCAN [4] for Scenario (c).

Event number	ε [cm]	The number of points		
		Proposed method	DBSCAN [4] ($\varepsilon = 10$ cm)	Ground truth
1	10	113	113	113
2	17	225	225	225
3	17	502	502	502
4	28	306	306	306
5	10	469	469	469
6	4	219	225	225
7	5	654	654	654
8	5	339	414	414
λ		97.21%	100%	100%
NMI score		0.87	1	1

Table 4. Evaluation comparison between the proposed method and DBSCAN [4] for Scenario (d).

Event number	ε [cm]	The number of points		
		Proposed method	DBSCAN [4] ($\varepsilon = 10$ cm)	Ground truth
1	6	603	894	304
2	6	176	0	176
3	32	589	589	589
4	32	2283	2283	2283
5	32	419	419	419
6	6	0	0	299
7	15	455	455	455
8	6	115	0	115
λ		93.56%	87.28%	100%
NMI score		0.59	0.21	1

Fig. 8. Point clouds representing the rockfall events (**Fig. 6(a)**): clusters obtained from the proposed method (top), clusters obtained from the traditional DBSCAN (middle), and the ground-truth clusters (bottom). In these figures, the same color indicates the same cluster. It can be seen that the traditional DBSCAN cannot separate Event 3 from Event 5 and Event 2 from Event 4. In contrast, the proposed method could separate these events.

5. Discussion

This section discusses two issues concerning the performance and the potential of the proposed method. First, in Scenario (c), the proposed method's purity score and NMI score are slightly lower than those of DBSCAN in two rockfall events (i.e., Event 6 and Event 8), as shown in **Table 3**. In Event 6, the proposed method missed six out of 225 data points, and in Event 8, it missed five out of 414 data points. Even though these misses are not of significance, it is important to investigate the effects.

The proposed method uses the adaptive epsilon strategy, i.e., the parameter ε can vary from cluster to cluster by applying the recursive loop with a stop criterion, as detailed in Section 3. This missing-point case occurs when the algorithm terminates the loop too slow. For Event 6 of Scenario (c), it stopped after ε went down to four. Thus, some points at the edge were missing, as shown in **Fig. 9**. Therefore, the stop criterion plays a crucial role. If the algorithm stops too fast (i.e., it stops while the value of ε is large), it may not be able to separate two supposedly different clusters, as shown in **Fig. 8** (middle). Thus, this kind of trade-off should be investigated further, to formulate a better stop criterion.

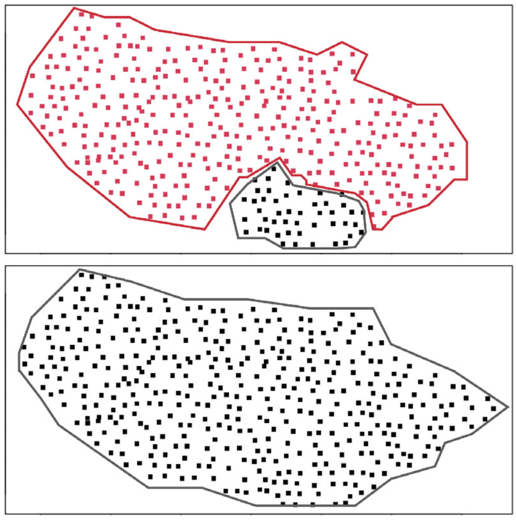


Fig. 9. Missing points in a cluster due to the smaller ε of the proposed method (top) and the ground-truth cluster (bottom). Note that the missing points in the top panel are bounded by the black line.

Second, when we look at Event 3 and Event 7 of Scenario (b), both methods could not separate Event 7 from Event 3. These two events are not easy to separate, even by the human eye. However, if we try to apply the same strategy that we used with ε to `minPts` after we obtain the customized ε (i.e., performing DC-DBSCAN recursively with an increase in `minPts`), we found that it is possible to separate Event 7 from Event 3, as shown in **Fig. 10**. This extension will be studied further in the future.

6. Conclusion

This paper proposed a version of DBSCAN that can automatically customize parameter ε for each cluster in a point cloud. The proposed method consists of two processes: initial parameter estimation based on grid analysis and DBSCAN based on the divide-and-conquer approach (DC-DBSCAN). DC-DBSCAN repeatedly performs DBSCAN on each cluster separately and recursively. Therefore, the suitable parameter ε for each group of the point cloud might not be a constant. The experimental results showed that the purity score and the NMI score of the proposed method were higher than those of traditional DBSCAN. This paper also discussed the potential of the proposed method in clustering two rockfall events that were close to each other and the challenge in separating them.

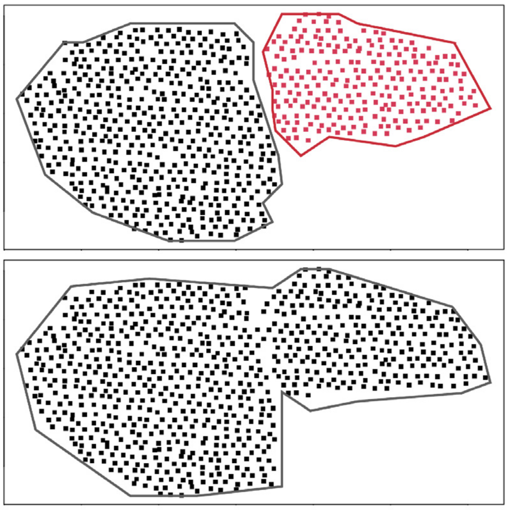


Fig. 10. The proposed method and the traditional DBSCAN cannot separate two events (top). However, when `minPts` is increased and go to 8, two clusters can be recognized (bottom).

Acknowledgements

This research is supported by Thailand Advanced Institute of Science and Technology (TAIST), National Science and Technology Development Agency (NSTDA), and Tokyo Institute of Technology under the TAIST-Tokyo Tech program. This work has also been partially supported by the ASEAN Committee on Science, Technology and Innovation (COSTI) under the ASEAN Plan of Action on Science, Technology and Innovation (APASTI) funding scheme and by the e-Asia JRP funding scheme.

References:

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," KDD'96: Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining, pp. 226-231, 1996.

[2] E. Schubert, J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN," ACM Trans. on Database Systems (TODS), Vol.42, No.3, Article 19, pp. 1-21, 2017.

[3] A. Abellán, J. Calvet, J. M. Vilaplana, and J. Blanchard, "Detection and spatial prediction of rockfalls by means of terrestrial laser scanner monitoring," Geomorphology, Vol.119, No.3-4, pp. 162-171, 2010.

[4] M. Tonini and A. Abellan, "Rockfall detection from terrestrial LiDAR point clouds: A clustering approach using R," J. of Spatial Information Science, Vol.2014, No.8, pp. 95-110, 2014.

[5] N. Micheletti, M. Tonini, and S. N. Lane, "Geomorphological activity at a rock glacier front detected with a 3D density-based clustering algorithm," Geomorphology, Vol.278, pp. 287-297, 2017.

[6] K. Sheridan, T. G. Puranik, E. Mangortey, O. J. Pinon-Fischer, M. Kirby, and D. N. Mavris, "An Application of DBSCAN Clustering For Flight Anomaly Detection During The Approach Phase," American Institute of Aeronautics and Astronautics (AIAA) Scitech 2020 Forum, 2020.

[7] A. Karami and R. Johansson, "Choosing DBSCAN Parameters Automatically Using Differential Evolution," Int. J. of Computer Applications, Vol.91, No.7, pp. 1-11, 2014.

[8] H. Darong and W. Peng, "Grid-based DBSCAN algorithm with referential parameters," Physics Procedia, Vol.24, Part B, pp. 1166-1170, 2012.

[9] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," J. of Open Source Software, Vol.2, No.11, Article 205, 2017.

[10] W. Lai, M. Zhou, F. Hu, K. Bian, and Q. Song, "A new DBSCAN parameters determination method based on improved MVO," IEEE Access, Vol.7, pp. 104085-104095, 2019.

[11] A. Abellán, D. Carrea, A. Loye, M. Tonini, M. Jaboyedoff, M. Royan, and A. Pedrazzini, "Understanding precursory rockfalls along cracks," Geophysical Research Abstracts, Vol.14, EGU2012-6037-1, 2012.

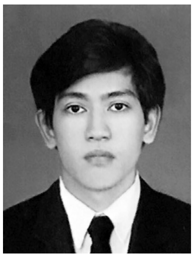
[12] K. T. Chau, R. H. C. Wong, J. Liu, and C. F. Lee, "Rockfall hazard analysis for Hong Kong based on rockfall inventory," Rock Mechanics and Rock Engineering, Vol.36, No.5, pp. 383-408, 2003.

[13] "3D Landslide Website," http://3d-landslide.com/ [accessed May 17, 2021]

[14] "CloudCompare Website," https://www.danielgm.net/cc/ [accessed May 17, 2021]

[15] T. O. Kvålseth, "On normalized mutual information: measure derivations and properties," Entropy, Vol.19, No.11, Article 631, 2017.

[16] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," New J. of Physics, Vol.11, No.3, Article 033015, 2009.



Name:
Pitisit Dillon

Affiliation:
Master's degree Student Under TAIST-Tokyo Tech program, Sirindhorn International Institute of Technology, Thammasat University

Address:
Phahonyothin Road, Khlong Nueng, Khlong Luang, Pathum Thani 12120, Thailand

Selected Publications:
- "Rockfall Detection from Terrestrial LiDAR Point Clouds by Using DBSCAN with Clutter Removal Based on Grid Density," Int. Conf. on Computational Intelligence in Information System, pp. 129-139, 2021.
- "Improving Accuracy of Dissolved Oxygen Measurement in an Automatic Aerator-Control System for Shrimp Farming by Kalman Filtering," Int. Conf. on Computational Intelligence in Information System, pp. 145-156, 2018.

Name:
Pakinee Aimmanee

Affiliation:
Sirindhorn International Institute of Technology, Thammasat University

Address:
131 Moo 5 Tivanont Road, Bangkadi, Meung, Patum Thani 12000, Thailand

Name:
Akihiko Wakai

Affiliation:
Professor, Gunma University

Address:
1-5-1 Tenjin, Kiryu, Gunma 376-8515, Japan

Name:
Go Sato

Affiliation:
Professor, Graduate school of Environmental Informations, Teikyo Heisei University

Address:
4-21-2 Nakano, Nakano, Tokyo 164-8530, Japan

Name:
Hoang Viet Hung

Affiliation:
Associate Professor, Faculty of Civil Engineering, Thuyloi University of Vietnam

Address:
175 Tayson Street, Dongda, Hanoi, Vietnam

Name:
Jessada Karnjana

Affiliation:
National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency

Address:
112 Thailand Science Park, Khlong Luang, Pathum Thani 12120, Thailand