



## Rockfall Detection from Terrestrial LiDAR Point Clouds by Using DBSCAN with ClutterRemoval Based on Grid Density

---

Pitisit Dillon, Jessada Karnjana and Pakinee Aimmanee

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

November 23, 2020

# Rockfall Detection from Terrestrial LiDAR Point Clouds by Using DBSCAN with Clutter Removal Based on Grid Density

Pitisit Dillon<sup>1,2</sup>, Jessada Karnjana<sup>1</sup>, and Pakinee Aimmanee<sup>2</sup>

<sup>1</sup> NECTEC, National Science and Technology Development Agency  
112 Thailand Science Park, Khlong Luang, Pathum Thani, 12120 Thailand

<sup>2</sup> Sirindhorn International Institute of Technology, Thammasat University  
131 Moo 5, Tiwanon Rd., Bangkadi, Muang, Pathum Thani, 12000 Thailand  
m6222040278@g.siit.tu.ac.th

**Abstract.** This paper proposes a simple method for rockfall detection from terrestrial LiDAR point clouds. The method consists of four steps: registration, subtraction, clutter removal, and spatial clustering. The paper contributes a straightforward method for clutter removal based on grid density, which is computational complexity inexpensive compared to the standard method based on nearest neighbor distance. Experimental results show that both are comparable in terms of identifying rockfall events. The proposed method can detect 21 events from 27 events from our simulations, and a conventional method can detect 23 events. The false-positive events of the proposed and conventional methods are 1 and 15, respectively. In contrast, for 52,000 points, the proposed method is about 16 times faster. Also, this paper suggests a simple means to estimate the parameters used in the spatial clustering algorithm.

**Keywords:** Rockfall detection, terrestrial LiDAR point cloud, Clutter removal, DBSCAN

## 1 Introduction

A rockfall is a free fall movement of a detached segment of bedrock from a cliff or a very steep slope [4]. It can be considered as the faster type of landslide. Rockfalls cause damages not only to properties but also to lives. The former is mostly by impeded transportation and commerce due to blocked highways and waterways. The latter is as direct casualties from falling rocks [8]. The rockfall phenomena can be studied in a number of approaches, such as historical inventories, susceptibility assessment, frequency estimation, hazard assessment, and risk assessment [12]. Recently, a terrestrial laser scanner (TLS) has been used to study geological phenomena due to its advantage in acquiring high-resolution data [3][5][12][10]. For example, the TLS with the density-based spatial clustering of applications with noise (DBSCAN) has been applied in studying geomorphology of a rock glacier [10]. The similar technique has also been applied for rockfall detection [3][5][12]. A typical framework used to detect rockfalls from

two point-cloud datasets, which are obtained from the TLS, consists of three parts: preprocessing, clutter removal, and spatial clustering [12], and a standard clutter removal algorithm is based on the nearest-neighbor-related approach, such as the nearest neighbor clutter removal (NNCR) [6]. However, the algorithm like NNCR has one problem concerning the computational time when the number of data points is large. Therefore, this paper aims to propose and discuss a comparable but more straightforward method for removing the clutter.

The rest of this paper is organized as follows. Section 2 provides background information used in the proposed method. Section 3 describes the detail of the proposed method. Section 4 reports experimental conditions and results. Discussion is made in Section 5, and Section 6 concludes this work.

## 2 Background

This section provides background information concerning an open-source software tool used to preprocess data points (i.e., CloudCompare) and a standard density-based spatial clustering method, called DBSCAN. Both are used together with our proposed clutter removal based on grid density.

### 2.1 CloudCompare

CloudCompare is an open-source 3D point cloud editing and processing software [2]. It was firstly designed to compare two point-cloud datasets directly. It can also perform many tasks, such as image registration, resampling, segmentation, and some statistical computation. One of its advantages is that CloudCompare can process more than 10 million data points on a standard laptop. In this work, CloudCompare is used to preprocess data in two steps: image registration and surface subtraction.

### 2.2 Density-based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN is one of the most common spatial clustering algorithms, which is based on data point density [7]. The algorithm collects points that are close to each other together (i.e., the points in high-density regions) and assigns the collection as a cluster. For points that lie in low-density areas or those not in the cluster, they are designated as noise. The DBSCAN concept and algorithm are briefly reviewed in this subsection for the completeness in itself.

Let  $p$  and  $q$  are points in a database  $D$ , and  $d(p, q)$  denotes a distance between points  $p$  and  $q$  with regard to a distance function  $d$ , e.g., Euclidean distance or Manhattan distance. The  $\varepsilon$ -neighborhood of a point  $p$ , denoted by  $N_\varepsilon(p)$ , is defined as a set of points of which their distances to the point  $p$  is not greater than  $\varepsilon$ . That is,  $N_\varepsilon(p) = \{q \in D \mid d(p, q) \leq \varepsilon\}$ . A point  $p$  is said to be *directly density-reachable* for a point  $q$  if it satisfies two conditions. First,  $p \in N_\varepsilon(q)$ . Second,  $N_\varepsilon(q)$  has at least `minPts` points, i.e.,  $|N_\varepsilon(q)| \geq \text{minPts}$ , where `minPts`

is a positive integer. Note that the point  $q$  that satisfies the second condition is called a *core point* [7].

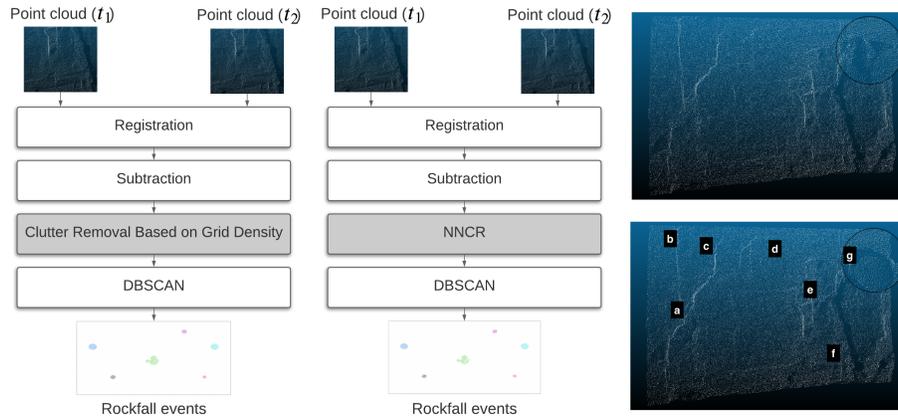
A point  $p$  is said to be *density-reachable* from a point  $q$  if there is a sequence of points  $p_1, p_2, \dots, p_n$ , where  $p_1 = q$ ,  $p_n = p$ , and  $p_{i+1}$  is *directly density-reachable* from  $p_i$ . A point  $p$  is said to be *density-connected* to a point  $q$  if there exists a point from which the point  $p$  and  $q$  are *density-reachable*.

Based on the concepts of density-reachability and density-connectivity, the DBSCAN algorithm groups together points in  $D$  and assigns them to members of a cluster  $C$ , which is a non-empty subset of  $D$ , if the following conditions are satisfied. First, for any points  $p$  and  $q$ , if  $p \in C$  and  $q$  is *density-reachable* from  $p$ , then  $q \in C$ . Second, for all points  $p$  and all points  $q$  in  $C$ ,  $p$  is *density-connected* to  $q$ . According to the definition of the cluster, the DBSCAN algorithm can discover a cluster by choosing a *core point* first and then retrieving all points that are *density-reachable* from that core point [7]. All other points that do not belong to any cluster are considered as noise. The abstract DBSCAN algorithm can be summarized in three steps [11]. First, find  $N_\varepsilon(p)$  for all points  $p$  in  $D$  and identify core points. Second, join neighbor core points into clusters, ignoring all non-core points. Last, for each non-core point, add to a nearby cluster if the cluster is an  $\varepsilon$ -*neighborhood*; otherwise, assign it to noise.

### 3 Proposed Method

The proposed method is similar to the conventional rockfall detection method [12] based on DBSCAN with the nearest neighbor clutter removal (NNCR) [6] in the sense that it consists of the same four processes: surface registration, subtraction, clutter removal, and DBSCAN, as shown in Fig. 1 (left and middle). However, instead of deploying NNCR, which is computational time expensive, we propose a simple grid-density-based process to do the same task in the clutter removal.

The proposed method's inputs are two 3D point-cloud surfaces, and the proposed method can be summarized as follows. First, two point-cloud surfaces are registered before comparison so that both are geometrically aligned. Figure 1 (right) shows an example of two input point-cloud surfaces. Second, the difference between the two surfaces is determined. Note that the Cloud Compare software performs these first two steps. In order to determine the difference, a threshold value is set. The result of this subtraction is a set of data points of which their differences are above the threshold. Third, considered only in 2D, the subtraction result from the previous step, which is called a *surface-difference image*, is divided into equal square boxes of size  $w$ , as shown in Fig. 2 (b). Then, the number of data points in each box is counted. Let  $c_{\max}$  and  $c_{\min}$  are maximum and minimum numbers of data points in a box, respectively. According to our observation, data points in any box with less than  $(c_{\max} - c_{\min})/2$  data points can be considered clutters. All clutters are then removed from the surface-difference image. Fourth, the clutter-removed surface-difference image obtained from the previous step is clustered by DBSCAN to identify rockfall events.



**Fig. 1.** Proposed method (left) and the conventional method (middle). Example of two input point-cloud surfaces: before rockfall events (right-top) and after the events (right-bottom). The rockfall events are marked by characters from *a* to *g*.

## 4 Experiments and Results

This section provides details of the datasets used in our simulation, experimental conditions, experiments, and results.

### 4.1 Data

Our experiments' point cloud dataset is taken from an open dataset provided by Abellan *et al.* [3][12] and available online [1]. The data points were acquired by a terrestrial laser scanner, Optech Intelligent Laser Ranging and Imaging System (ILRIS3D), from the Puigcerros cliff, Catalonia, Spain, as shown in Fig. 3. The wavelength of the infrared laser pulse of ILRIS3D is 1,535 nm. The distance  $d$  between the cliff's surface and the instrument is determined by the flight time  $\Delta t$  of the pulse, i.e.,  $d = c\Delta t/2$ , where  $c$  is the speed of light [9]. The points' positions are initially recorded in the spherical coordinate system, i.e., (radial distance  $r$ , azimuthal angle  $\theta$ , polar angle  $\phi$ ). Later, they are converted to coordinates in the Cartesian system  $(x, y, z)$  by the equation  $(x, y, z) = (r \sin \theta \cos \phi, r \sin \theta \sin \phi, r \cos \theta)$ . Not only the coordinate, but the instrument also records the reflective-light intensity for each data point as well. Thus, the standard format of each data point record can be represented by a quadruple  $(x, y, z, I)$ , where  $I$  is the intensity. The maximum range of ILRIS3D is 700 m, with an error of 0.7 cm at 100 m.

In order to evaluate the performance of the proposed method, two datasets are required: one representing the surface before rockfall events and another representing the surface after the events. In this work, we generated the latter from the former by the following steps so that we can know the number and

locations of the rockfall events exactly. First,  $P$  percent of the data points are randomly selected. Second, all selected points are randomly shifted in the  $+x$ -axis direction for a certain value (in cm) in an interval  $[a, b]$ . These two steps can represent natural noise in the measurement process and an error due to the instrument’s installation, not at exactly the same locations, at two times. Third, a few simulated hemispherical holes with various diameters are added to the surface by modifying some  $y$  values of the data points. Some holes are generated by two to five partially-overlapped hemispherical holes. These holes can represent the deformation of the surface after the rockfall events. An example of the holes is shown in Fig. 1 (right). Note that in order to reduce the computational time in our simulation, 3D point-cloud inputs used in experiments are subareas or regions of the image in Fig. 3 (middle), as shown in Fig. 3 (bottom). The numbers of data points of all five regions are 15, 718, 41, 466, 33, 645, 13, 898, and 5, 385 points, for region numbers running from 1 to 5.

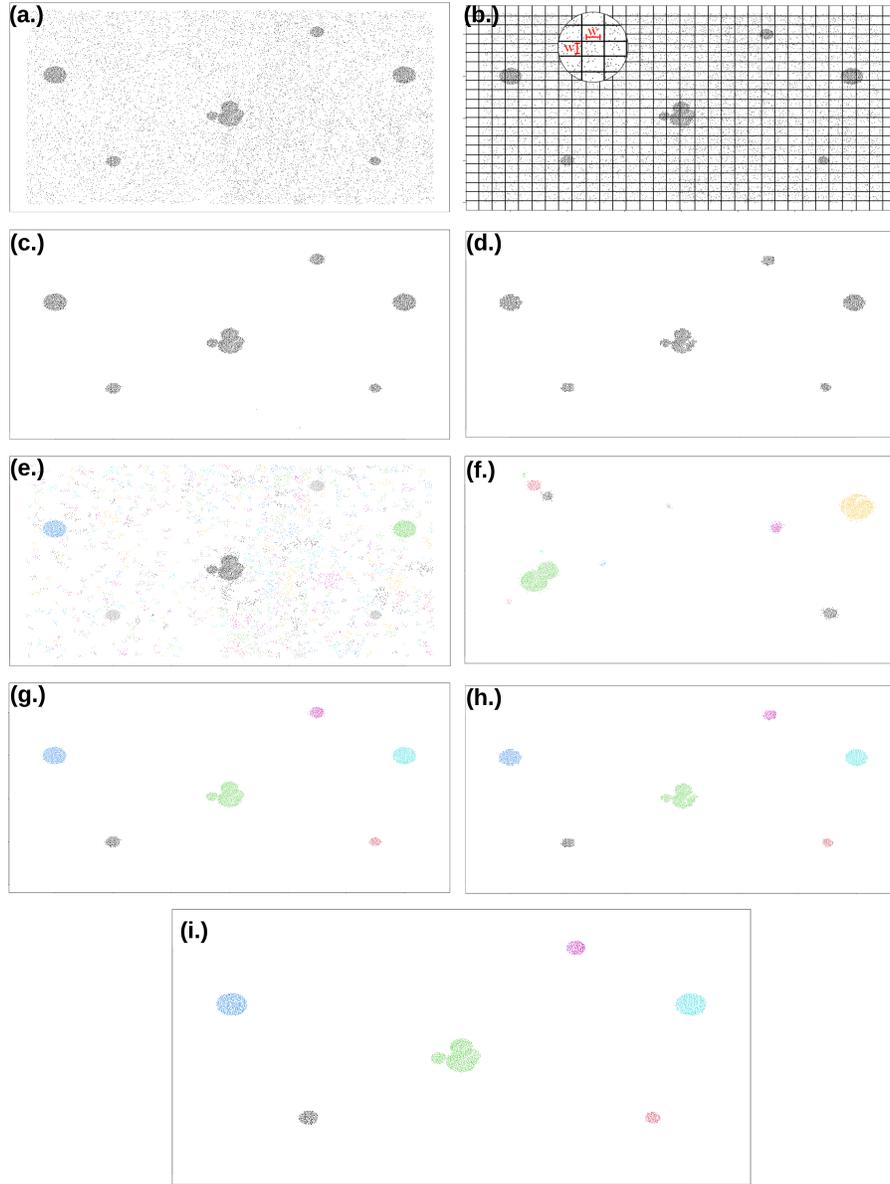
## 4.2 Experimental Conditions

The numbers of rockfall events in all five regions were as follows. Note that some events were generated from two to five hemispheres. These events are marked with the asterisk, as shown in Table 1. In region 1, there are 7 events from **a** to **g** with the largest radii of 25, 20, 7, 20, 60, 50, and 25 cm, respectively, and the smallest radius of **f** is 20 cm. In region 2, there are 9 events from **a** to **i** with the largest radii of 60, 40, 15, 40, 25, 40, 20, 30, and 60 cm, respectively, and the smallest radii of **a** and **d** are 25 and 20 cm, respectively. In region 3, there are 7 events from **a** to **g** with the largest radii of 7, 18, 10, 20, 50, 25, and 40 cm, respectively, and the smallest radius of **e** is 35 cm. In region 4, there are 7 events from **a** to **g** with the largest radii of 3, 25, 40, 40, 40, 25, and 20 cm, respectively, and the smallest radius of **e** is 10 cm. In region 5, there are 5 events from **a** to **d** with the largest radii of 4, 5, 14, and 25 cm, respectively, and the smallest radius of **d** is 18 cm.

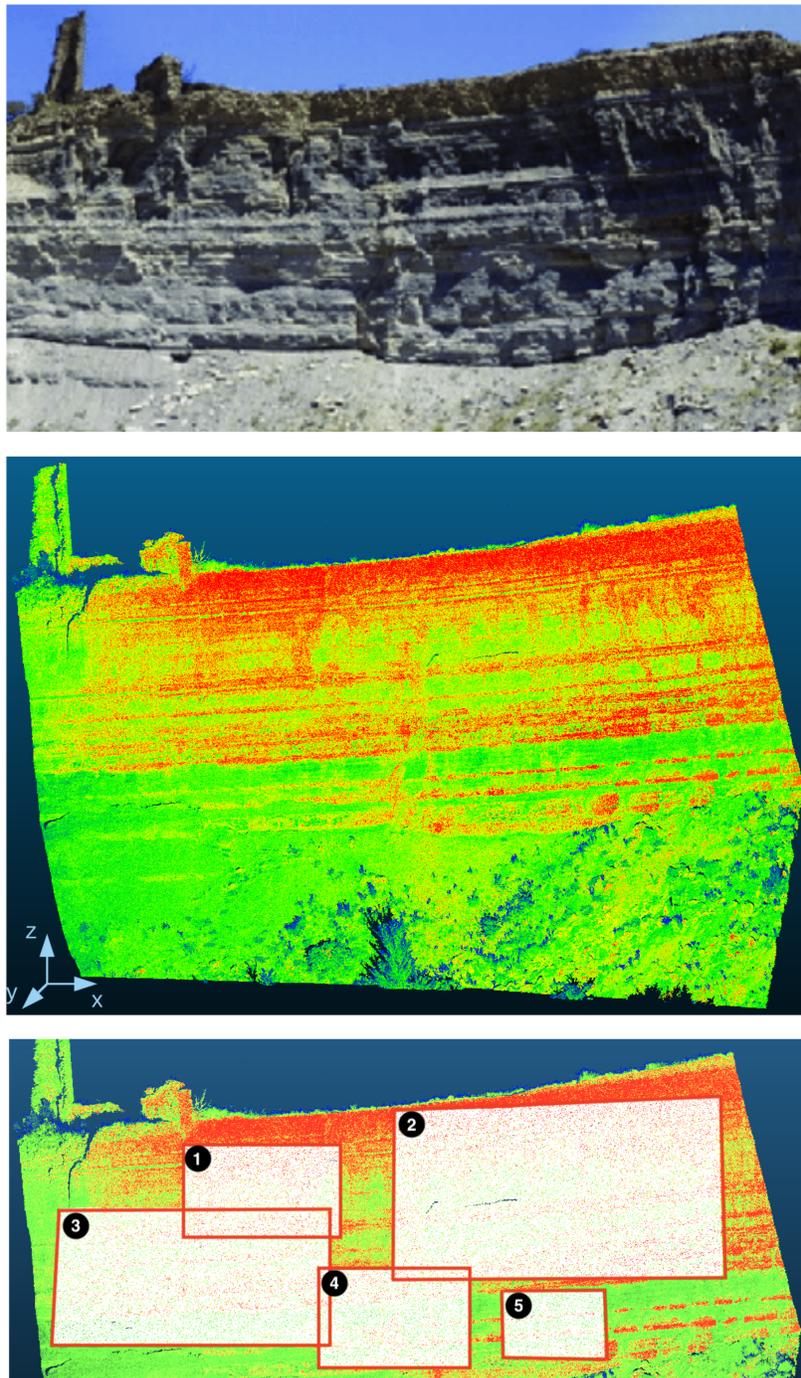
The conditions used in our simulations are as follows. The parameter  $P$ , which is the percent of data points that are randomly selected for shifting, was set to 90%. The shifting interval  $[a, b]$  was set to  $[1, 2]$  because the resolution of ILRIS3D is around 2 cm. The box’s size  $w$  was set to 10 cm because we assume that a rockfall event with a diameter less than 10 cm might not of importance in this work. The DBSCAN parameters used here are 10 and 5 for  $\epsilon$  and `minPts`, respectively. Note that these parameters’ values are suggested by the conventional method [12].

## 4.3 Experimental Results

Comparisons of data points detected in rockfall events between the proposed method and the conventional method [12] are shown in Table 1. It can be seen that, in a total of 34 events, the proposed method could identify 30 events, and the conventional method could detect 27 events. The conventional method is slightly better than the proposed method in this aspect. The proposed method



**Fig. 2.** Comparison of the results of the proposed method and the conventional method [12]: (a) surface-difference image from the subtraction process, (b) grid of equal square boxes of size  $w$ , (c) clutter-removed surface-difference image obtained from NNCR, (d) clutter-removed surface-difference image obtained from the proposed grid-density-based method, (e) detected rockfall events when the surface-difference image is put as the input of DBSCAN (with parameters suggested by the conventional method) directly, (f) detected rockfall events when applying only DBSCAN with parameters estimated by the proposed grid-density-based method (without the clutter removal), (g) detected rockfall events by the conventional method, (h) detected rockfall events by the proposed method, and (i) ground truth.



**Fig. 3.** 3D point cloud dataset (middle) scanned from the Puigcercos cliff (top), Catalonia, Spain [3][12][1], and five regions used in our simulations (bottom).

could not detect some events, except the event *c* in region 3, because the sizes of the simulated hemispheres are smaller than the box's size  $w$ . In other words, the smallest event the proposed method can detect has a size of  $w$ . These results are not beyond the expectation and can be solved by reducing the box's size. However, in terms of false-positive detection, the proposed method is better, i.e., the proposed method infrequently identified rockfall when there is no rockfall compared to the conventional method. Example of detected rockfall events is shown in Fig. 2.

Figure 4 shows the comparison of the computational time complexity between the proposed method and the conventional method when the number of data points increases. It can be seen that the proposed method is faster.

**Table 1.** Comparison of data points detected in rockfall events between the proposed and the conventional methods [12]. Note that **nFP** is a number of false-positive points. Note also that the numbers in parentheses are radii.

Region 1								
	a(25)	b(20)	c(7)	d(20)	e(60)	f*(50)	g(25)	nFP
Ground truth	153	97	14	111	768	1060	160	0
Conventional	161	100	13	129	774	1084	174	15
Proposed	97	65	0	95	501	948	121	7

Region 2										
	a*(60)	b(40)	c(15)	d*(40)	e(25)	f(40)	g(20)	h(30)	i(60)	nFP
Ground truth	1087	548	61	471	198	568	144	308	1546	0
Conventional	1099	554	60	476	203	576	145	313	1568	16
Proposed	173	245	11	0	77	284	94	155	1389	0

Region 3								
	a(7)	b(18)	c(10)	d(20)	e*(50)	f(25)	g(40)	nFP
Ground truth	8	81	27	100	769	163	482	0
Conventional	0	80	25	102	776	170	487	56
Proposed	0	62	0	42	420	85	294	0

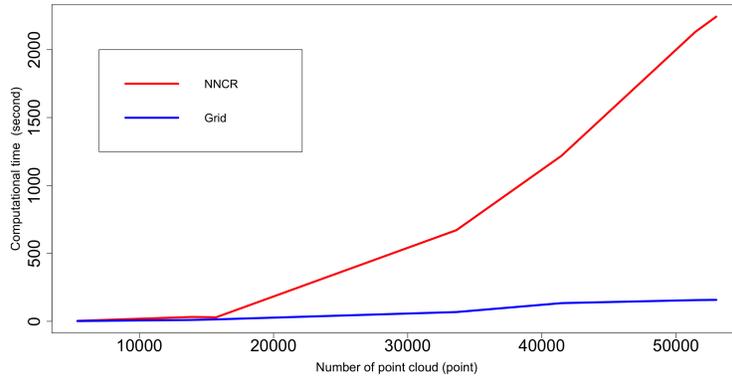
Region 4								
	a(3)	b(25)	c(40)	d(40)	e*(40)	f(25)	g(20)	nFP
Ground truth	3	197	501	520	859	194	135	0
Conventional	0	204	510	530	877	199	137	4
Proposed	0	165	462	497	775	161	126	0

Region 5					
	a(4)	b(5)	c(14)	d*(25)	nFP
Ground truth	7	10	74	234	0
Conventional	0	0	76	243	0
Proposed	0	0	48	202	0

## 5 Discussion

In this section, we discuss the potential of using grid density to estimate the parameters of DBSCAN. As mentioned in Section 2.2, the DBSCAN algorithm requires two reasonable parameters to cluster data points effectively. Those parameters are  $\varepsilon$  and **minPts**. The parameter  $\varepsilon$  is a distance to a point  $p$  that



**Fig. 4.** Comparison of the computational time between the proposed method and the conventional method [12].

defines a set  $N_\varepsilon(p)$ , and the parameter `minPts` is an integer that defines the core point, the point that lies in high-density areas. Hence, if the smallest distance between two points on a laser line can be approximated (e.g., by the instrument’s resolution), and the size  $w$  of the box of the grid is set to the laser line scanner resolution, the maximum number of points in the box (i.e.,  $c_{\max}$ ) can be reasonably used to approximate `minPts`. That is, `minPts` =  $c_{\max}$ , and  $\varepsilon = w$ . Figure 2(f) shows an example of the result of applying DBSCAN with  $c_{\max}$  and  $w$  to a surface-difference image. It can be seen that the resulting image is clearly better than the image in Fig. 2(d). This issue will be investigated further. Note that this result was obtained under the condition that  $[a, b]$  is  $[1, 2]$ .

Another critical issue worth discussing in this section is a relationship between the threshold used in the subtraction process by CloudCompare and the shifting interval  $[a, b]$ . It is clear that the threshold affects the number of points in the surface-difference image directly. In our simulation, we fixed  $[a, b]$  to  $[1, 2]$  to correspond to a possible situation for the laser line resolution of approximately 2 cm. We, therefore, fixed the threshold value to 1.8 cm. This threshold was set to this value because it resulted in a good performance for the conventional method [12], to which our proposed method is compared. The relationship between the threshold and the shifting interval should be studied further as well.

## 6 Conclusion

This paper proposed a method for rockfall detection based on DBSCAN with grid-density-based clutter removal from two data point clouds. The method consists of four steps: registration, subtraction, clutter removal, and DBSCAN. The paper’s contribution is in providing a simple method for clutter removal in comparison with the conventional NNCR. The experimental results showed that the proposed grid-density-based clutter removal worked comparably to the NNCR, while the computational time complexity reduced reasonably. In addition, this

paper suggested a straightforward means to estimate two parameters of the DB-SCAN algorithm based on the laser line resolution and grid density.

## Acknowledgement

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.

Also, this work has 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 e-Asia JRP funding scheme.

## References

1. 3D Landslide Website, <http://3d-landslide.com/>
2. CloudCompare Website, <https://www.danielgm.net/cc/>
3. Abellán, A., Calvet, J., Vilaplana, J.M., Blanchard, J.: Detection and spatial prediction of rockfalls by means of terrestrial laser scanner monitoring. *Geomorphology* **119**(3-4), 162–171 (2010)
4. Bates, R.L., Jackson, J.A.: *Glossary of geology* (1987)
5. Bitelli, G., Dubbini, M., Zanutta, A.: Terrestrial laser scanning and digital photogrammetry techniques to monitor landslide bodies. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **35**(B5), 246–251 (2004)
6. Byers, S., Raftery, A.E.: Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association* **93**(442), 577–584 (1998)
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd*. vol. 96, pp. 226–231 (1996)
8. Geertsema, M., Highland, L.: *Landslides: human health effects* (2011)
9. Jaboyedoff, M., Oppikofer, T., Abellán, A., Derron, M.H., Loye, A., Metzger, R., Pedrazzini, A.: Use of lidar in landslide investigations: a review. *Natural hazards* **61**(1), 5–28 (2012)
10. Micheletti, N., Tonini, M., Lane, S.N.: Geomorphological activity at a rock glacier front detected with a 3d density-based clustering algorithm. *Geomorphology* **278**, 287–297 (2017)
11. Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X.: Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)* **42**(3), 1–21 (2017)
12. Tonini, M., Abellan, A.: Rockfall detection from terrestrial lidar point clouds: A clustering approach using r. *Journal of Spatial Information Science* **2014**(8), 95–110 (2014)