International Society for Rock Mechanics and Rock Engineering
Norwegian Group for Rock Mechanics

ISRM International Symposium
Eurock 2025 – Expanding the Underground Space
Trondheim, Norway, 16-20 June
ISBN 978-82-8208-079-8

Unexposed rock joint identification based on point cloud analysis

C.C. Chiu
National Taipei University of Technology, Taipei, Taiwan
ccchiu@ntut.edu.tw

Abstract

This study determines unexposed rock joints based on point cloud analysis. Joint identification is crucial to automatic rock mass characterization. The conventional extraction way is based on three-dimensional point cloud facet fitting, which cannot identify unexposed joints, resulting in an underestimation. Therefore, this research proposes an extraction process to capture unexposed rock joints. The program is developed on the C++ platform with the CloudCompare and CCCoreLib libraries. It completes the fitting of three-dimensional joint traces through processes including color filtering, eigenvector computing, linearity computing, cylindrical DB scan clustering, lineation, and plane fitting. The fitted planes match the actual fracture locations, proving the effectiveness of the proposed fitting algorithm.

Keywords

Rock joint, fracture extraction, point cloud, automatic characterization, trace.

ISRM

NBG

1 Introduction

Joint identification is a crucial issue in rock mass characterization, and the identified results can further be utilized to build discrete fracture network (DFN) analysis or analyze kinematic stability. The conventional way is to manually sketch traces on a photo or perform window sampling, which is time-consuming. Fortunately, with the advance of survey methods, the maturity of unmanned aerial vehicles (UAV) and light detection and ranging (LiDAR) technology will allow engineers to quickly and conveniently collect large amounts of spatial information. Among them, the point cloud obtained through SFM technology or LiDAR scanning is valuable. A point cloud can be viewed as a collection of scanned points on the outcrop surface. By analyzing the point cloud, the outcrop joint can be extracted, and the spatial information of the rock joint can be further characterized. This kind of analysis method based on aerial survey technology or lidar scanning can not only obtain a large number of objective data at one time but also allow engineers to conduct outcrop surveys in a safe place. More importantly, these data are digitized, thus avoiding possible omissions caused by manual recording and reducing the burden on engineers. Therefore, combing joint identification with point cloud analysis is an important key to simplifying the survey methods in future geotechnical engineering site surveys.

In addition, with the improvement of computer performance and the maturity of analysis software, rock mechanics engineers can gradually construct more complex geological analysis models. The conventional analysis method mostly assumes that the mechanical behavior of the rock mass is homogeneous and isotropic, relies on drilling data to divide the site into layers with specific mechanical parameters, and then performs slope stability analysis and calculates the safety factor. This analysis mode is not sensitive to the spatial information of joint planes. Even if engineers know the importance of joint distribution to rock mass stability, practically, they will only consider the dominant orientation and analyze the kinematic stability further. Other spatial information, such as fracture intensity, fracture size, orientation variation, etc., are not quantitatively investigated for stability analysis.

Therefore, to better explore the stability of rock slopes, a more comprehensive and quantitative investigation and analysis method will inevitably be another crucial issue of future geotechnical analysis, and the extraction technology of rock joints is the core issue in terms of quantitative analysis. This research aims to develop a quantitative rock joint extraction method so that geotechnical engineers can establish the outcrop survey with minimal training and time costing and deliver the survey results to the senior engineers for further analysis.

The identification object of rock joints based on 3D point cloud can be distinguished by plane, edge, and trace. Plane means that the joint surface can be identified by a series of point clouds that share a similar normal vector. Edge means the joint surface can be determined by the turning point of planes. Trace is the unexposed rock joints that appear on the plane surface (Daghigh et al. 2022). The former two can be analyzed using a geometric-based method based on 3D coordinate analysis. The latter is hard to be identified by spatial analysis because its identification is usually based on color instead of geometry. The techniques of identifying traces based on 2D images are mature. However, most algorithms are developed based on image processing, such as Canny edge detection or Hough detection (Tang et al. 2021), which is hard to adopt in identifying 3D point cloud models.

The plane and edge identification has been studied frequently (Hackel et al. 2016; Dewez et al. 2016; Daghigh et al. 2022; Chiu and Liu 2024). Two powerful algorithms – RANSAC and Facet – have already been built using the open-source software CloudCompare. Several studies obtained joint distribution based on these two powerful tools. For example, Chen et al. (2017) uses an improved RANSAC method and an improved Floodfill algorithm to extract the shape of rock block, which can be treated as the identification method of plane surface. Menegoni et al.(2019) compare the difference in joint orientation distribution between manual measurement, DSE, qFacet FM, and qFacet Kd-tree method with different amounts of measurement data. Our group also uses these two algorithms to evaluate Q-system ratings based on the point clouds (Chiu and Liu 2024). Therefore, in this study, we only focus on developing a trace identification method instead of plane and edge detection. This study proposed several processes to extract rock joints from the point clouds. The methods include color filtering, eigenvector computing, linearity computing, cylindrical DB scan clustering, lineation, and plane fitting. The fitted planes match the actual fracture locations, proving the effectiveness of the proposed fitting algorithm. The research will continue to improve the fitting algorithm and attempt to enhance the accuracy of trace identification.

2 Research methods and results

2.1 CloudCompare and CCCoreLib

The CloudCompare software (CloudCompare 2002) is utilized as a major tool for point cloud analysis. It is a 3D point cloud processing software that can analyze point cloud data produced by UAV photogrammetry or lidar scanning. At the same time, it can also process triangle meshes, polygons, and polylines. The purpose of CloudCompare is to quickly detect changes in three-dimensional high-density point clouds acquired by lidar scanning in industrial facilities or construction sites. Later, it added more point cloud analysis algorithms to improve its effectiveness. CloudCompare is an open-source software so academic institutions can use abundant functions with CloudCompare without expensive cost. In addition, CloudCompare also provides its rich function library for public use, allowing users to mobilize analysis functions under the C++ interface. In this study, most point cloud processing is done in C++.

CloudCompare provides a set of basic tools for manually editing and analyzing 3D point clouds, and also includes many advanced processing algorithms, such as projection, comparison, distance calculation, statistical distribution, point cloud segmentation, point cloud thinning, geometric feature estimation, etc., and can present point cloud data in the software in a visual way to help users quickly grasp the point cloud situation. CloudCompare can handle most mainstream point cloud data formats, including obj, stl, las, ASCII, ply, etc. In addition, many high-end analysis algorithms such as M3C2, point cloud feature extraction, etc. can also be introduced through the plug-in function.

CCCoreLib is part of the core calculation library of CloudCompare and is responsible for most of the data structure and algorithm work. This study uses both CCoreLib and CloudCompare function libraries. The difference between the two is that CloudCompare is publicly licensed under GPL 3.0, therefore the derived programs that use its related functions must be open sourced in accordance with the GPL 3.0 public license; while CCCoreLib is under the LGPL 2.0 public license. If developers do not have the ability to change the functions Library source code can be used commercially.

2.2 Site and point cloud data

The study site is located in the Longdong, northeast region of Taiwan. The photogrammetry was implemented by UAV DJI Phantom 4 Pro, and Structure-from-Motion technology was used to build outcrop point clouds. To improve the analysis efficiency, only a part of the point clouds is utilized for analysis (Fig. 1). The number of points is 10,866,409, and the horizontal projection area is 8.476×8.914 m^2. Therefore, the vertical point resolution is 143,821 points per m^2.

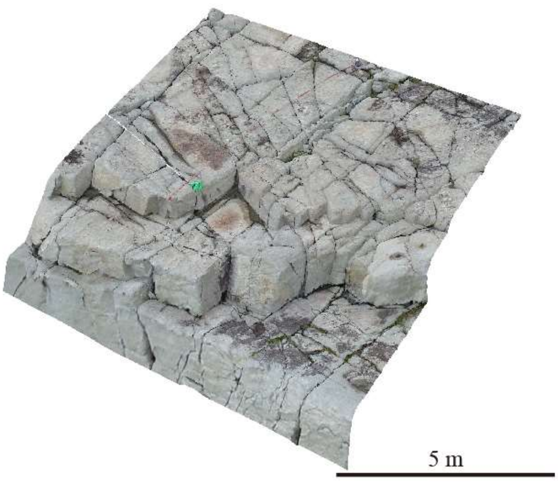

Fig. 1 Study site and point clouds of analysis regions

2.3 Process

Fig. 2 shows the fracture extraction process, which can be divided into the following steps: (1) Import the original point cloud into the C++ platform based on the CloudCompare library; (2) Perform point cloud filtering based on the RGB color information to obtain the point cloud belonging to the trace; (3) Perform eigenvalue and eigenvector calculations; (4) Calculate the linearity of the point cloud based on eigenvalues; (5) Perform the cylindrical DB Scan algorithm to clustering the trace point cloud; (6) Linearize the classified trace point cloud; (7) Perform plane fitting based on identified traces pair; (8) Finally obtain traces and the 3D joint surface.

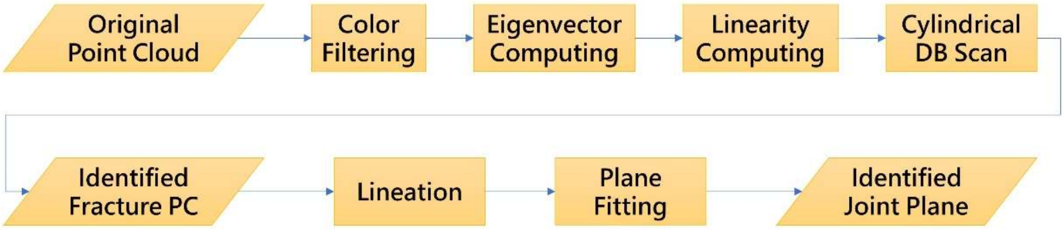

Fig. 2 Analysis process.

2.3.1 Color Filtering

Color Filtering is the first step to interpret point cloud cracks. The principle of trace analysis of point cloud identification is that the program identification logic must be based on the human's subjective judgment. The human eye determines the location of traces based on color information, so the first step in the identification process is to analyze the color content (RGB) of the point cloud (Fig. 3). Since most of the traces are black, this study summed up the red, green and blue contents of the point cloud:

$$C = (R + G + B)/3 \quad (1)$$

Where C Average of color composition
R Ratio of red color content
G Ratio of green color content
B Ratio of blue color content

The range of the above values is from 0 to 255. In this research, $C < 80$ is set as the threshold of trace identification (Fig. 4). This filtering process can be established both by CloudCompare software with the filtering of the composition of the RGB color scale or by using C++ processing. It is worth noting that this process does not convert RGB point clouds into grayscale. It still keeps RGB information. Also, $C < 80$ is not a strict regulation. The appropriate value of filtering depends on the environment of point clouds. The user must adjust the threshold based on their judgment that the value can extract the most traces. This trial process can be repeated with the assistance of CloudCompare.

After color filtering, traces are clearly exposed, which helps the further identification and also improve the efficiency due to the reduce of the number of point clouds.

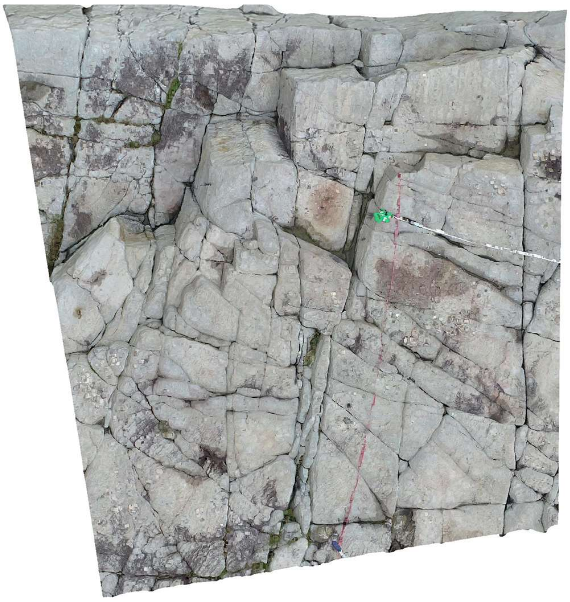

Fig. 3 Raw point cloud before color filtering.

Fig. 4 Point cloud after color filtering $C < 80$

2.3.2 Eigenvector Computing

The linearization degree of traces is the basis for identifying traces in this study, which can be determined by analyzing the eigenvalues and eigenvectors of the point cloud. In this study, a radius of 10 cm around a single point cloud is selected as the analysis region, and the included point clouds are called a point cloud group. Then, calculate the geometric center of the point cloud group, find the vector ($\overrightarrow{x_n}$) from each point in the group to the center, and organize all vectors into matrix form:

$$X = \begin{bmatrix} \overrightarrow{x_1} \\ \overrightarrow{x_2} \\ \cdots \end{bmatrix} \tag{2}$$

The next step is to find the covariance matrix A of the point cloud group:

$$A = X^T X \tag{3}$$

In this way, by calculating the eigenvalues and eigenvectors of matrix A, the spatial arrangement characteristics of the point cloud group can be obtained. If the point cloud group exhibits a highly linear relationship, its maximum eigenvalue will be much larger than the second and third eigenvalues, and the maximum eigenvector will be in the direction of the trace.

2.3.3 Linearity Computing

The calculation of linearity uses the eigenvalues of point clouds to calculate the spatial distribution characteristics of point clouds. Linearity is calculated as follows(Weinmann et al. 2013):

$$L = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{4}$$

Where λ_1 Maximum eigenvalue
λ_2 Intermediate eigenvalue

After completing the calculation of eigenvalues, the calculation of linearity is then carried out. Fig. 5 shows the calculated linearity distribution. It can be found that in the trace area, linearity is above 0.9 (red area), and only at the crack junction or at the wider crack point cloud, the linearity is lower than 0.9. Therefore, this study uses 0.9 as the cut-off for trace interpretation and only performs subsequent extraction on point clouds with linearity greater than 0.9 (Fig. 6).

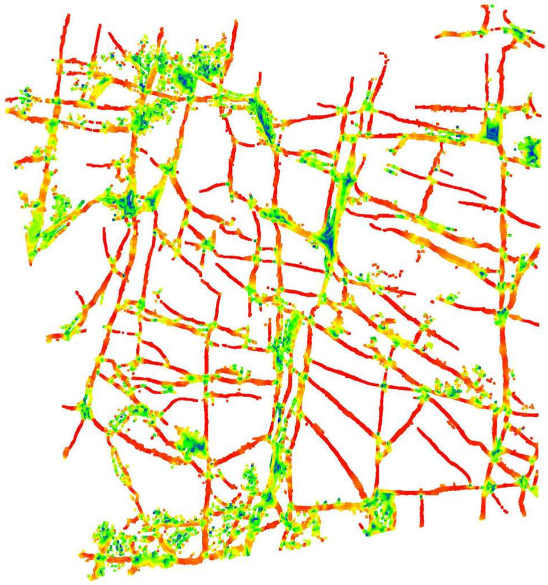

Fig. 5 Distribution of linearity.

Fig. 6 Range of linearity > 0.9.

2.3.4 Cylindrical DB Scan

DB Scan (Density-based spatial clustering of applications with noise) is a cluster analysis algorithm (Ester et al. 1996), which is one of the most commonly used cluster analysis algorithms in data science. This algorithm can group nearby points and mark noise points located in low-density areas and can find clusters of any shape. Noise points can also be marked simultaneously. The parameters required for its analysis are also quite few. It only needs to determine the minimum required point threshold and the search radius, which is quite simple and convenient.

Traditional DB Scan searches based on a set search radius. Fig. 7 is a schematic diagram of the operation of DB Scan. Point A is a randomly selected search starting point. DB Scan will count the number of data points within the search radius. If the number is greater than the required point threshold, point A is the core point, and the data points within the search radius are the data groups to be searched. Then, for Repeat the above steps for the data group to be searched. If the point threshold is met, the point can be added to the category of point A. Repeat the above steps until the data group to be searched is retrieved, and then the next round of search can be performed for unmarked points. If the selected point does not meet the required point threshold, it is set as noise (blue point N in Fig. 7).

Cylindrical DB Scan is an algorithm obtained by modifying DB Scan in this study. Conventional DB Scan searches for data points within a specified radius, so it is a spherical search. However, the filtered point cloud in this study is a highly linear 3D data. If the point cloud is searched in a spherical manner, it is difficult to meet the point requirements and may cause misjudgment. Therefore, we change the search range to a cylindrical shape. This search method also needs to determine the required point threshold and search radius. Then the maximum eigenvector of the search point is used as the cylinder axial direction, the cylinder radius is the search radius, and the cylinder height is four times the search radius (that is, toward the positive and negative maximum feature direction twice the search radius). In this way, cluster interpretation of linear cracks can be efficiently performed. Fig. 8 and Fig. 9 are the results of Cylindrical DB Scan clustering. Different colors represent different clusters. Fig. 8 seems to have a gradient because the search of point clouds has an order; the clustering index will be related to the search order. Therefore, a wrong impression may occur that it only has a few clusters. Fig. 9 is clearer because each cluster is wrapped in an outer frame through plane fitting to facilitate observation (Fig. 9).

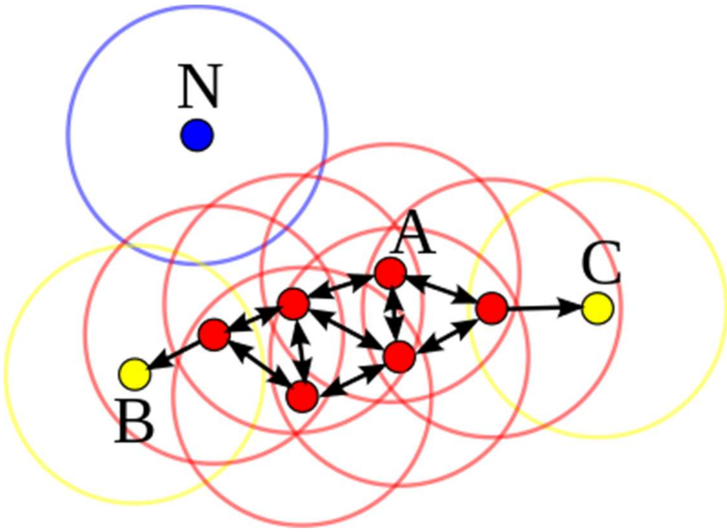

Fig. 7 Schematic of DB scan clustering

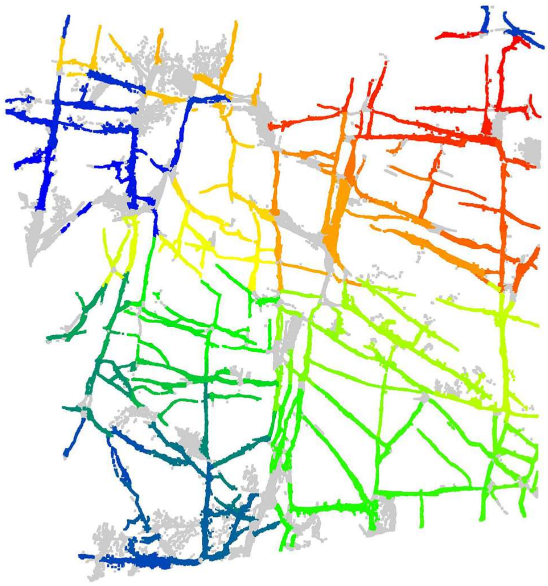

Fig. 8 Clustering after cylindrical DB scan.

Fig. 9 Frame after clustering.

2.3.5 Lineation

After completing the clustering of trace point clouds, each cluster point cloud can be linearly simplified. Since the linearity of the point cloud after clustering is quite high, we can start from the center point of the cluster, take the maximum eigenvector of the cluster point cloud as the linear direction, and the diagonal of the cluster's bounding box is the length of the crack line. A simplified trace line can be obtained by extending half of the crack line length from the cluster center to both sides in the linear direction. Fig. 10 is the identified simplification traces. It is found that traces highly overlap with the traces on point clouds, indicating the extraction works appropriately.

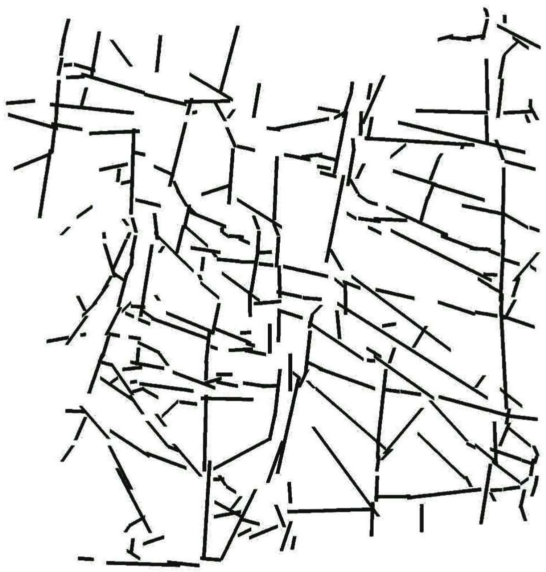

Fig. 10 Result of lineation.

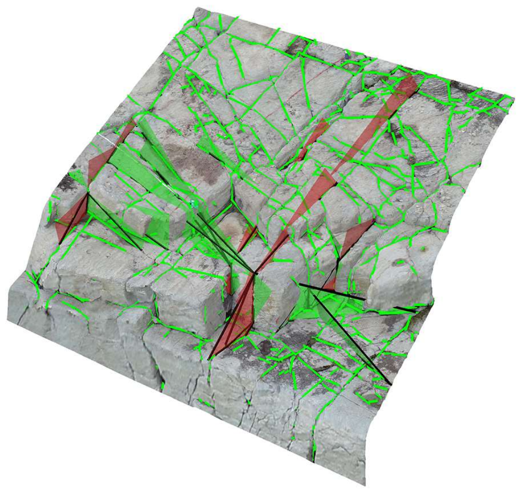

Fig. 11 Identified joint planes.

2.3.6 Plane Fitting

After completing the identification of traces, the next step is to fit the 3D joint surface. The goal of this step is to combine the co-plane 3D traces into a 3D joint plane; otherwise, the position of the crack cannot be determined correctly. To avoid fitting unconnected cracks, such as a set of traces that are parallel to each other or cracks that are too far apart. This study combines crack line types according to the following procedures:

(1) Decide the threshold distance and calculate the shortest distance between the two fitting traces. If the shortest distance between the two fitting traces is greater than the threshold distance, it means that the two fitting traces do not belong to the same joint plane or are parallel lines. In this study, the threshold distance was set to 10 cm.

(2) Set the threshold angle, so that the angle between the two traces must be greater than the threshold angle. The purpose of this setting is to prevent two parallel traces being fitted into joint planes. In this study, the threshold angle is set to 30 degrees (that is, angle between the two traces larger than 30 degrees is possible to be fitted as a joint plane).

(3) The length of the two fitting lines to be combined must be greater than 50 cm to avoid misjudgment caused by small traces.

Fig. 11 is a 3D joint plane obtained through the proposed fitting process; the elevation of the point cloud has been reduced a little bit to better present the identified traces on the outcrop. It can be found that most of the traces cut through the actual location of the traces, so the identification results are pretty reasonable. Some co-plane traces have been identified as joint planes, but not all traces are identified because those traces may not have enough 3D exposure on the point cloud.

3 Conclusions

This study proposes a preliminary trace identify processes based on 3D point cloud analysis. The program uses C++ as the development platform, uses CloudCompare and CCCoreLib function libraries to perform point cloud analysis calculations, and extracts the 3D joint plane through color filtering, eigenvector computing, linearity computing, cylindrical DB scan clustering, lineation and plane fitting processes. The fitting joint plane matches the actual joint position, which proves that the proposed identify algorithm has certain effectiveness. This research still has lots of future work, such as advancing the fitting algorithm to improve the accuracy of trace identification based on point cloud analysis, adopting the proposed algorithm to other cases to examine its universality, and finding a method to assess the accuracy of the proposed algorithm.

Acknowledgement

The author would like to thank the financial support of this research under Contracts NSTC 112-2221-E-027-055-MY3. I also thank Mr. Chun-Yuan Liu from Sinotech Engineering Consultants, Inc. for his kind assistance with UAV photogrammetry.

References

Chen N, Kemeny J, Jiang Q, Pan Z (2017) Automatic extraction of blocks from 3D point clouds of fractured rock. Comput Geosci 109:149–161. https://doi.org/10.1016/j.cageo.2017.08.013

Chiu CC, Liu CY (2024) Case Study of Program-Assisted Q-System Rating Based on Point Cloud Analysis. p D042S056R003

CloudCompare (2002) CloudCompare official site. https://www.cloudcompare.org/doc/wiki/index.php/Introduction. Accessed 15 Aug 2022

Daghigh H, Tannant DD, Daghigh V, et al (2022) A critical review of discontinuity plane extraction from 3D point cloud data of rock mass surfaces. Computers & Geosciences 169:105241. https://doi.org/10.1016/j.cageo.2022.105241

Dewez TJB, Girardeau-Montaut D, Allanic C, Rohmer J (2016) FACETS : A cloudcompare plugin to extract geological planes from unstructured 3d point clouds. Int Arch Photogramm Remote Sens Spatial Inf Sci XLI-B5:799–804. https://doi.org/10.5194/isprs-archives-XLI-B5-799-2016

Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. AAAI Press, Portland, Oregon, pp 226–231

Hackel T, Wegner JD, Schindler K (2016) Contour Detection in Unstructured 3D Point Clouds. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 1610–1618

Menegoni N, Giordan D, Perotti C, Tannant DD (2019) Detection and geometric characterization of rock mass discontinuities using a 3D high-resolution digital outcrop model generated from RPAS imagery – Ormea rock slope, Italy. Eng Geol 252:145–163. https://doi.org/10.1016/j.enggeo.2019.02.028

Tang Y, He L, Lu W, et al (2021) A novel approach for fracture skeleton extraction from rock surface images. International Journal of Rock Mechanics and Mining Sciences 142:104732. https://doi.org/10.1016/j.ijrmms.2021.104732

Weinmann M, Jutzi B, Mallet C (2013) Feature relevance assessment for the semantic interpretation of 3D point cloud data. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-5-W2:313–318. https://doi.org/10.5194/isprsannals-II-5-W2-313-2013