remote sensing

Article

Major Orientation Estimation-Based Rock Surface Extraction for 3D Rock-Mass Point Clouds

Lupeng Liu, Jun Xiao * and Ying Wang

School of Artificial Intelligence, University of Chinese Academy of Sciences, No. 19 Yuquan Road, Shijingshan District, Beijing 100049, China; liulupeng14@mails.ucas.ac.cn (L.L.); ywang@ucas.ac.cn (Y.W.)
* Correspondence: xiaojun@ucas.ac.cn; Tel.: +86-10-8825-6566

Received: 2 February 2019; Accepted: 6 March 2019; Published: 15 March 2019

Abstract: In the fields of 3D modeling, analysis of discontinuities and engineering calculation, surface extraction is of great importance. The rapid development of photogrammetry and Light Detection and Ranging (LiDAR) technology facilitates the study of surface extraction. Automatic extraction of rock surfaces from 3D rock-mass point clouds also becomes the basis of 3D modeling and engineering calculation of rock mass. This paper presents an automated and effective method for extracting rock surfaces from unorganized rock-mass point clouds. This method consists of three stages: (i) clustering based on voxels; (ii) estimating major orientations based on Gaussian Kernel and (iii) rock surface extraction. Firstly, the two-level spatial grid is used for fast voxelization and segmenting the point cloud into three types of voxels, including coplanar, non-coplanar and sparse voxels. Secondly, the coplanar voxels, rather than the scattered points, are employed to estimate major orientations by using a bivariate Gaussian Kernel. Finally, the seed voxels are selected on the basis of major orientations and the region growing method based on voxels is applied to extract rock surfaces, resulting in sets of surface clusters. The sub-surfaces of each cluster are coplanar or parallel. In this paper, artificial icosahedron point cloud and natural rock-mass point clouds are used for testing the proposed method, respectively. The experimental results show that, the proposed method can effectively and accurately extract rock surfaces in unorganized rock-mass point clouds.

Keywords: rock surface extraction; Gaussian Kernel; voxelization; unorganized point cloud; Rock Mass; major orientations

1. Introduction

Feature surface extraction in point clouds is a requisite part of many computer graphics, image processing and computer vision, including 3D reconstruction [1], object recognition [2,3], virtual reality [4], etc. Remote sensors, such as Light Detection and Ranging (LiDAR) and Differential SAR Interferometry (DInSAR), can capture high resolution and accurate 3D information of object surface from a considerable distance [5,6]. The rapid development and popularization of remote sensors promotes the research of surface extraction and 3D modeling in unorganized point clouds. Nevertheless, the existing surface-extraction methods can't deal with the growing size of the point clouds.

Currently, the three common methods of surface extraction in point clouds are Hough Transform (HT) [7], Random Sample Consensus (RANSAC) [8] and Region Growing (RG) [9,10]. The Hough Transform is a traditional approach of detecting parameterized shapes and objects in 2D images. It is typically used for lines and circles detection [11]. The 3D Hough Transform can also extract feature surfaces, cylinders and spheres in 3D point clouds [12]. However, the traditional 3D Hough Transform is complex, time consuming and the detected surfaces may be disconnected. To detect a surface with RANSAC, three points are chosen randomly to calculate the surface parameters defined by them. A score function is used to determine whether the model fits the remaining points [13]. The algorithm stops when it reaches stability. RANSAC may detect spurious feature surfaces or discontinuous surfaces. Region Growing performs a local search to identify and expand the regions with same characteristics [14]. Region Growing may lead to holes and over-segmentation issues.

In recent years, the prevailing methods of surface extraction have focused on urban scenes or indoor scenes. While the urban point clouds consist of regular geometric surfaces (such as walls and floors, etc.), the rock-mass point clouds are shaggy and mostly in the form of high and steep slope [15]. Stability analysis is the core of geotechnical engineering such as aboideau, tunnel and highwall [5,16,17]. Presently, the various methods of stability analysis in rock mass (such as block theory [18], discontinuous deformation analysis [19] and numerical manifold method [20], etc.) mostly be performed on the accurate numerical models of rock mass. The rock surface extraction and modeling in rock-mass point clouds becomes a requisite part of geotechnical engineering. The main driver of our methodology is to extract rock surfaces and reconstruct the watertight 3D model of rock mass itself.

In this paper, the proposed method is designed based on the rough surface and irregular shape of rock mass. By using the spatial grid for fast voxelization, the approximate coplanar points are classified. At the same time, the different coplanar criteria are compared and advice for adjusting the parameters of coplanar criteria are pointed out. To estimate major orientations and overcome the shortage of computational complexity, we cast votes for the normal vectors [21] of every approximate coplanar voxels by using a bivariate Gaussian Kernel [22]. Eventually, according to the results of major orientation estimation, the voxels of region growing units are selected [23]. The region growing based on voxels guarantees the connectivity of detected rock surfaces and improves efficiency, which results in sets of coplanar clusters and the largest connected sub-surfaces of each cluster. Notice that the discontinuity sets are now actual planes but not surfaces (such as folds, etc.). To its analysis, it is common to accept them as planes, but for that consideration, it must be assumed that the region of analysis has homogeneous characteristics.

The remaining part of this paper is structured as follows: after providing an overview on related work, the proposed method of rock surface extraction is thoroughly discussed in Section 3. Section 4 shows and analyzes the experimental results. In the final Section 5, we conclude this paper and point out the future work.

2. Related Work

2.1. Hough Transform

The 2D Hough Transform is a standard method proposed by Hough [7] which could detect geometric primitives in the field of digital image processing. The 3D Hough Transform is proposed on the basis of 2D Hough Transform, which is able to detect the feature surfaces in 3D point clouds. Every single point in point set $\{P\}$ is converted to a sinusoidal surface in Hough space and voting for the accumulator cells which intersect with the sinusoidal surface (i.e., Hough Voting). After voting for all points of $\{P\}$, the local maximum values are selected from the Hough space (i.e., peak detection). Then the peaks are converted to the surface equations and the sample points that satisfy these equations are found out. The time complexity of Hough Transform is $O(|P|N_\theta N_\phi)$, where $|P|$ is the number of sample points, N_θ is the number of accumulator cells in direction of θ and N_ϕ in direction of ϕ, respectively. The time complexity and space complexity of HT depend on the discretization degree of the accumulator. The higher the discretization degree, the higher computational complexity.

Borrmann et al. [24] summarized the traditional Hough Transform and designed a new spherical accumulator. Different variants of the Hough Transform were compared and analyzed through experiments, including the standard Hough Transform (SHT), probability Hough Transform (PHT), random Hough Transform (RHT) and so on. Vosselman et al. [21] proposed a two-step procedure for Hough Transform, which used normal vectors to confirm the spatial position of the parameter. The two-step procedure reduces the complexity of mapping from the measurement space to the parameter space. Huang and Brenner [25] proposed a joint multi-plane detection strategy to improve the performance of the Hough Transform, which had good performance on the roof with ridge lines. Hulik et al. [26] proposed an optimization method for continuous surface extraction in point clouds. The integral image normal computation of the Point Cloud Library (PCL) [27] was used for estimating the local normal vectors of each point. When accumulating to the Hough space, a hierarchical structure was used to ensure high-resolution would not consume additional memory. A buffer strategy was proposed for Gaussian smoothing and a sliding-window technique was applied in Peak detection. Fernandes and Oliveira [28] proposed the Kernel-based Hough Transform (KHT) for real-time detection of straight lines in two-dimensional images. Due to the lack of clear information of the adjacent samples, it was not applicable to unorganized point clouds. In this regard, Limberger and Oliveira [22] proposed an improved Octree-based clustering strategy for 3D unorganized point clouds (i.e., 3D-KHT). Octree and Principal Component Analysis (PCA) were used to segment approximate coplanar samples. Hough voting for each sample point was carried out on the basis of Gaussian Kernel, which significantly increased the efficiency.

2.2. Region Growing

The use of region growing in point clouds to extract structural feature surfaces is a 3D simulation of region growing in the field of Image Processing [23,29]. By selecting a seed, it searches for local to identify and expand regions with the same characteristics.

Poppinga et al. [30] used priority queues and depth images to quickly extract the adjacent neighbors. The calculation methods of incremental covariance matrices and mean square error (MSE) through mathematical formulas were introduced. When a new point was added, there was no need to construct a new covariance matrix, which improved the efficiency of region growing. Xiao et al. [23] proposed the strategy of using sub-windows as seeds for region growing. A sub-window-based region growing (SBRG) for structured environment and a hybrid region growing (HRG) which combined sub-windows and spatial points for unstructured environment were introduced respectively. Vo et al. [31] proposed an octree-based region growing method for urban point cloud segmentation. The algorithm was divided into two stages based on a coarse-to-fine concept. In the first stage, the octree-based region growing was performed for major segmentation. The second stage was a refinement process. This method was at least an order of magnitude faster than conventional region growing algorithms.

2.3. Random Sample Consensus

Another important parameterization method for surface extraction in point clouds is Random Sample Consensus (RANSAC) [8]. It iteratively selects three points to calculate the surface equation and count the number of points located in this surface. The time complexity of RANSAC is $O(I|P|)$, where I is the number of iterations. RANSAC is robust to noise. However, due to its randomness, the time complexity is uncertain.

Yang and Förstner [32] presented an approach of surface extraction by integrating RANSAC and Minimum Description Length (MDL). This method could avoid detecting false feature surfaces caused by the complex geometry of point cloud. To avoid generating spurious-surfaces (i.e., consisting of points from several coplanar surfaces), Xu et al. [33] proposed the weighted RANSAC method according to the difference in error distribution between proper and improper surface hypotheses. Li et al. [34] proposed an improved RANSAC method based on Normal Distribution Transformation (NDT) cells. In each iteration, a surface NDT cell was selected as the minimum sample to ensure the correctness of the sampling on the same surface.

2.4. Other Methods

Holz et al. [35] presented a method to cluster the sample points in normal space and spherical coordinates simultaneously. By using 3D semantic awareness, this method could conduct obstacle detection and collision avoidance. Yamazaki et al. [36] utilized topology method and graph-cut method to extract surfaces. Without constructing a triangular mesh or others, the surfaces were segmented into different feature representations.

In the field of rock-mass engineering, some automatic or semi-automatic tools for surface extraction and analysis of discontinuities are publicly available to engineers and researchers (i.e., COLTOP 3D [37], Split-FX [38], DiAna [39], PlaneDetect [15], DSE [6] and so on). Riquelme et al. [6] presented a method for automatic characterization of rock mass discontinuities. This method analyzed the coplanarity of adjacent points to identify and define the algebraic equations of different rock surfaces. The Kernel Density Estimation (KDE) was used to find the main directions and the density-based scanning principle was used to identify the clusters. The authors published the Discontinuity Set Extractor (DSE) open-source software and this software also calculated the normal spacing and the discontinuity persistence [40,41]. By considering the Hough Transform and region growing simultaneously, Leng et al. [42] proposed a multi-scale surface-detection algorithm. Hough Transform was used to extract the seeds and feature surfaces of rock mass were extracted by region growing. The rock surfaces of different spatial scales were detected separately by introducing the concept of multi-scale.

3. Methodology

On the basis of summarizing the conventional surface extraction methods in point clouds, this paper proposes an effective rock surface extraction method in rock-mass point clouds. The summary of this method is shown in Algorithm 1. In this method, the spatial grid is used for fast voxelization and clustering approximate coplanar points. After voting for each coplanar voxel in normal space, the major orientations are obtained and the region growing based on voxels is performed. Finally, multiple sets of rock surfaces with the largest connected sub-surfaces are obtained. The proposed method avoids the phenomenon of "riband" [42] and facilitates subsequent operations (i.e., surface merging, concave calculation and 3D reconstruction).

Algorithm 1: Proposed rock surface extraction method.

```
Input: 3D point cloud $\{P\}$
Output: Rock surfaces
1  $\{clusters\} \leftarrow Clusterting\{P\}$ {clustering approximate coplanar points};
2  foreach $\vec{n}$ of $\{clusters\}$ do
3      accumulator← accumulator+voting($\vec{n}$){major orientation estimation based on bivariate
        Gaussian Kernel} ;
4  end
5  $\{\vec{n}\} \leftarrow$ peak detection {major orientations};
6  foreach $\vec{n_i}$ in $\{\vec{n}\}$ do
7      while find a seed accroding to $\vec{n_i}$ do
8          region growing based on voxels;
9      end
10 end
```

3.1. Clustering Based On Voxels

In the preprocessing stage, we use the kd-tree module and the k-Nearest Neighbor (knn) algorithm of the Point Cloud Library [27] to create indexes and estimate the normal vector of each point. It takes a bit of time to load and pre-process raw datasets. In recent years, researchers have proposed a newer version of neighborhood algorithms with better performance [43,44].

Similar to octree, the two-scale spatial grid is a data structure for describing the three-dimensional space. As shown in Figure 1, each node of the spatial grid represents a cube voxel and could be divided into 8 sub-cubes (i.e., sub-voxels). The main reason we choose two-scale spatial grid instead of octree is that we could rapidly and easily searching for adjacent voxels by using three-dimensional array.

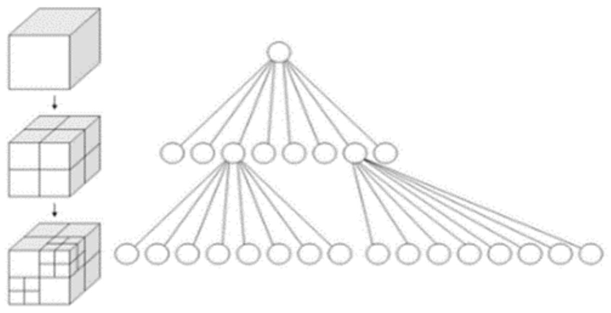

Figure 1. Spatial grid structure

The pseudo-code of voxelization and clustering is described in Algorithm 2. The coplanar clustering is done by analysing the eigenvalues of covariance matrix. For a voxel A, the covariance matrix $\sum$ is computed and coplanar constraints are used to determine whether the sample points contained in A have the coplanar characteristics. We sort the eigenvalues of matrix $\sum$ in ascending order (i.e., $\lambda_1 \leq \lambda_2 \leq \lambda_3$). With respect to the setting of the approximate coplanar constraints, Limberger and Oliveira [22] proposed a criterion for comparing λ (see Equation (1)) and Riquelme et al. [6] used the curvature to determine the coplanarity (see Equation (2)), where S_α, S_β and η are thresholds.

$$\lambda_2 > S_\alpha \lambda_1 \&\& S_\beta \lambda_2 > \lambda_3 \tag{1}$$

$$\frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \leq \eta \tag{2}$$

In this paper, the coplanar constraints shown in Equation (3) are used, among which S_α and ε are thresholds. Mean Squared Error (MSE), as shown in Equation (4), is a statistical method to measure the average error and the variation of data, which is commonly used for region growing.

$$\lambda_2 > S_\alpha \lambda_1 \&\& MSE < \varepsilon \tag{3}$$

$$MSE = \frac{1}{k} \sum_{i=1}^{k} \left(\overrightarrow{n} \cdot p_i - d \right)^2 \tag{4}$$

After coplanar clustering, we have obtained hundreds (thousands) of approximately coplanar voxels, and the non-coplanar phenomenon still exists in coplanar voxels. On the one hand, since the coplanar voxels contain non-coplanar sample points, the normal vectors (i.e., "principal directions") obtained by PCA for each "coplanar voxel" is not accurate. On the other hand, it is difficult to find the "major orientations" from the hundreds (thousands) of "principal directions" for selecting growing seeds. Taking these two aspects into consideration, we use a bivariate Gaussian Kernel to vote for the "principal direction" of each coplanar voxels (see Section 3.2). In the end, we selected a few more representative "major orientations" from the hundreds of "principal directions".

3.2. Major Orientation Estimation Based On Gaussian Kernel

In the above, the point cloud of rock mass is segmented by coplanar clustering, and the approximate coplanar voxels are obtained. In this section, major orientation estimation based on Gaussian Kernel is mainly performed on the normal vectors of these coplanar voxels.

3.2.1. Hemispherical Surface Accumulator

Many geologists and engineers have proposed various plane-detection methods through using stereonets. These methods mainly concentrate on the analysis of rock-mass discontinuities. Since the main driver of our methodology is to reconstruct the watertight 3D model of rock mass itself, we do not use stereonets here.

Algorithm 2: Coplanar clustering based on voxels.

```
Input :
$\{P\}$: 3D point cloud;
$min_{bigvoxel}$: the minimum number of points in big voxel;
$min_{samllvoxel}$: the minimum number of points in sub_voxel;
$voxel_{size}$: the length of big voxel;
Output:
$\{Voxel_{cop}\}$: the coplanar voxels;
$\{Voxel_{non}\}$: the non-coplanar voxels;
$\{Voxel_{spa}\}$: the sparse voxels;

Finding the bounding box of $\{P\}$;
$\{big\_voxels\} \leftarrow$ Segmenting $\{P\}$ according to $voxel_{size}$;
foreach $voxel_i$ of $\{big\_voxels\}$ do
    if $|voxel_i| > min_{bigvoxel}$ then
        if $voxel_i$ is coplanar{defined in Equation (3)} then
            $\{Voxel_{cop}\} \leftarrow \{Voxel_{cop}\} + voxel_i$;
        else
            $\{sub\_voxels\} \leftarrow$Segmenting $voxel_i$ into 8 sub_voxels;
            foreach $sub\_voxel_i$ of $\{sub\_voxels\}$ do
                if $|sub\_voxel_i| > min_{smallvoxel}$ then
                    if $sub\_voxel_i$ is coplanar{defined in Equation (3)} then
                        $\{Voxel_{cop}\} \leftarrow \{Voxel_{cop}\} + sub\_voxel_i$;
                    else
                        $\{Voxel_{non}\} \leftarrow \{Voxel_{non}\} + sub\_voxel_i$;
                    end
                else
                    $\{Voxel_{spa}\} \leftarrow \{Voxel_{spa}\} + sub\_voxel_i$;
                end
            end
        end
    else
        $\{Voxel_{spa}\} \leftarrow \{Voxel_{spa}\} + voxel_i$;
    end
end
```

If we map all unit normal vectors of coplanar voxels to a sphere, the end point of each unit normal vector must fall on the spherical surface. So we consider using the accumulator ball as the accumulator design. Notice that only the surface of the ball is divided into accumulator cells and all of these accumulator cells are of the same size [24] (as shown in Figure 2a). As we all known, the direction of normal vector can be positive or negative. If we reverse the direction of these normal vectors that fall in the lower hemisphere (i.e., $z < 0$ in Figure 2a), the end point of each normal vector must fall in the upper hemisphere(i.e., $z \geq 0$ in Figure 2a). According to these features of normal vectors, we eventually decide to use the hemispherical surface accumulator.

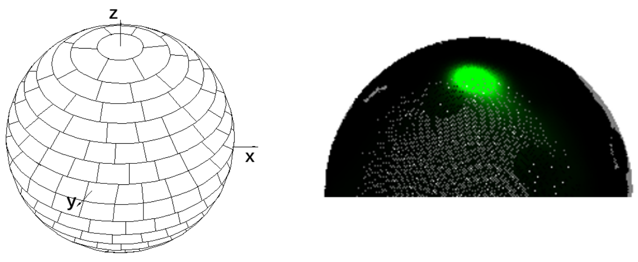

Figure 2. The accumulator design. (**a**) Accumulator ball. (**b**) Result of voting.

The example of voting result is shown in Figure 2b. When it is close to the poles (i.e., ϕ is close to 0), the voting value is smaller than other regions, which is similar to the spherical accumulator. This ultimately results in the vertical surfaces being detected earlier than the horizontal surfaces, which does not affect the correctness of the results [22,24].

3.2.2. Computing Bivariate Gaussian Kernels

Assuming that K is an approximate coplanar voxel (see Section 3.1), whose covariance matrix is $\sum_{x,y,z}$ and the center of gravity is $\mu = (\mu_x, \mu_y, \mu_z)^T$. $V = \{\overrightarrow{v_1}, \overrightarrow{v_2}, \overrightarrow{v_3}\}$ is the eigenvector of $\sum_{x,y,z}$ and $\Lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ $(\lambda_i \leq \lambda_{i+1})$ is the corresponding eigenvalue. The surface passes through μ and it's normal vector is $\overrightarrow{n} = \overrightarrow{v_1} = (n_x, n_y, n_z)^T$, so that the Equation (5) can be rewritten as Equation (6).

$$Ax + By + Cz + D = 0 \tag{5}$$

$$n_x x + n_y y + n_z z - (n_x\mu_x + n_y\mu_y + n_z\mu_z) = 0 \tag{6}$$

Using spherical coordinates, Equation (6) can be rewritten as Equation (7),

$$\begin{aligned} &\rho = -D{=}n_x\mu_x + n_y\mu_y + n_z\mu_z = \sqrt{p_x^2 + p_y^2 + p_z^2}, \\ &\theta = \arctan\left(\frac{p_y}{p_x}\right), \phi = \arccos\left(\frac{p_z}{\rho}\right), \end{aligned} \tag{7}$$

where $\rho \in \mathbb{R}_{\geq 0}$, $\theta \in [0°, 360°)$, $\phi \in [0°, 180°]$, $\overrightarrow{p} = (p_x, p_y, p_z) = \rho\,\overrightarrow{n}$.

The detail of computing $\sum_{\phi,\theta}$ and voting thresholds of coplanar voxels is described in Algorithm 3. With the use of first-order uncertainty propagation analysis [45], $\sum_{\phi,\theta}$ could be derived from $\sum_{x,y,z}$ as

$$\Sigma(\theta,\phi) = \begin{bmatrix} \sigma_\phi^2 & \sigma_{\phi\theta} \\ \sigma_{\phi\theta} & \sigma_\theta^2 \end{bmatrix} = J\Sigma(x,y,z)J^T = J \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 \end{bmatrix} J^T, \tag{8}$$

where J is the Jacobian matrix:

$$J = \begin{bmatrix} \frac{\partial\phi}{\partial p_x} & \frac{\partial\phi}{\partial p_y} & \frac{\partial\phi}{\partial p_z} \\ \frac{\partial\theta}{\partial p_x} & \frac{\partial\theta}{\partial p_y} & \frac{\partial\theta}{\partial p_z} \end{bmatrix} = \begin{bmatrix} \frac{p_x p_z}{\sqrt{w}\rho^2} & \frac{p_y p_z}{\sqrt{w}\rho^2} & -\frac{\sqrt{w}}{\rho^2} \\ -\frac{p_y}{w} & \frac{p_x}{w} & 0 \end{bmatrix}, \tag{9}$$

where $w = p_x^2 + p_y^2$.

Algorithm 3: Computing $\sum_{\phi,\theta}$ and voting thresholds of coplanar voxels.

```
Input :
{$Voxel_{cop}$}: the coplanar voxels;
Output:
{$\sum_{\phi,\theta}$}: covariance matrix in $(\phi,\theta)$ space;
{Th_min}: voting thresholds;
1  foreach $voxel_i$ in $\{Voxel_{cop}\}$ do
2      $\sum_{x,y,z} \leftarrow$ cov($voxel_i$) {computing covariance matrix in (x,y,z) space};
3      $\sum_{\phi,\theta} \leftarrow J\sum_{x,y,z} J^T$ {$\sum_{\phi,\theta}$ and J are defined in Equations (8) and (9)};
4      $(V_{\phi,\theta}, \Lambda_{\phi,\theta}) \leftarrow eigen(\sum_{\phi,\theta})$ {eigen-decomposition};
5      $\lambda_{min} \leftarrow smallestEigenvalue\ in\ \Lambda_{\phi,\theta}$;
6      $V_{min} \leftarrow Eigenvector(\lambda_{min})$;
7      $std\_dev \leftarrow sqrt(\lambda_{min})$ {standard devation};
8      $Th\_min \leftarrow Gaussian(2 \times std\_dev \times V_{min})$ {Using Equation (11) };
9      $\{\sum_{\phi,\theta}\} \leftarrow \{\sum_{\phi,\theta}\} + \sum_{\phi,\theta}$;
10     $\{Th\_min\} \leftarrow \{Th\_min\} + Th\_min$;
11 end
```

3.2.3. Estimating Major Orientations

For a coplanar voxel K, its normal vector $\overrightarrow{\boldsymbol{n_k}}$ can be converted into polar coordinates (ϕ_k, θ_k) according to Equation (7). Voting for $\overrightarrow{\boldsymbol{n_k}}$ is to vote for accumulator cell of (ϕ_k, θ_k) and its surrounding cells. The voting value can be calculated from a multi-dimensional Gaussian Kernel (see Equations (10) and (11)). Since the covariance matrix is symmetric and positive definite, the two-dimensional Gaussian distribution probability density equation [46] is shown as

$$p\left(x|\mu,\sum\right) = \frac{1}{(2\pi)^{k/2}|\sum|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^t \sum^{-1}(x-\mu)\right), \tag{10}$$

let $k = 2$, $\overrightarrow{\delta} = x - \mu$, this equation can be rewritten as

$$\mathrm{p}\left(\overrightarrow{\delta},\sum\right) = \frac{1}{2\pi|\sum|^{1/2}} \exp\left(-\frac{1}{2}\overrightarrow{\delta}^t \sum^{-1} \overrightarrow{\delta}\right), \tag{11}$$

where μ denotes (ϕ_k, θ_k), $\overrightarrow{\delta}$ denotes the distance with respect to μ, and $\sum$ denotes the covariance matrix about (ϕ, θ). It is worth noting that, when voting for a voxel with a small variance, the voting area of this voxel is small and vice versa.

For a coplanar voxel K, voting is carried out on the accumulator's cells where the corresponding (ϕ_k, θ_k) is located. Iteratively voting is performed around (ϕ_k, θ_k) until the calculated value of Equation (11) is smaller than the voting threshold of K (see Section 3.2.2). In this paper, we set two standard deviation to calculate the voting thresholds (see line 8 in Algorithm 3), which can guarantee a correct rate of 95.4%. However, in our experimental tests, the rock surfaces could be extracted efficiently by using one standard deviation to calculate the voting thresholds, and the efficiency of major orientation estimation could be improved greatly.

Besides calculating the voting value by Equation (11), the spatial length of voxel and the number of points contained in voxel are also considered [22]. A weighting function is defined as

$$w_k = w_a\left(\frac{voxel_{size}}{mesh_{size}}\right) + w_d\left(\frac{|K|}{|P|}\right), \tag{12}$$

where $w_a + w_d = 1$, $voxel_{size}$ is the edge length of the voxel and $mesh_{size}$ is the edge length of the bounding box. $|K|$ is the number of points contained in voxel and $|P|$ is the total number of the point cloud. The voting value received by the accumulator cell is the product of Equations (11) and (12).

Since we use the hemispherical surface accumulator, the number of accumulator cells is not very large and the Sliding-Window method is used for peak detection. A window ($N \times N$) is defined to slide around the accumulator. In this window, only the maximum value is retained. After sliding the window around the hemispherical surface, the window has passed through each accumulator cell and the major orientations has been obtained.

3.3. Rock Surface Extraction

After major orientation estimation, a set of major orientations is obtained. Voxel-based region growing is then performed according to these major orientations. The details of region growing based on voxels are shown in Algorithm 4, where $\{\overrightarrow{N}\}$ refers to the major orientations and $\{Voxel_{cop}\}$ refers to the coplanar voxels (see Section 3.1). "t_1" represents the maximum angle between normal vector of coplanar voxel and major orientation. The value of "t_1" should be as small as possible. "t_2" is the maximum angle threshold for region growing.

Algorithm 4: Region growing based on voxels.

```
Input :
$\{\overrightarrow{N}\}$: major orientations;
$\{Voxel_{cop}\}$: the coplanar voxels;
$t_1$ : maximum angle threshold for searching for seed voxels;
$t_2$: maximum angle threshold for region growing;
$Max_{dis}$: maximum distance between a point to a surface or a surface to another surface;
Output:
$\{Surfaces\}$: result of rock surface extraction;

1  foreach $\overrightarrow{n_i}$ in $\{\overrightarrow{N}\}$ do
2      foreach $voxel_j$ in $\{Voxel_{cop}\}$ do
3          if $\overrightarrow{n_i} \cdot normal(voxel_j) \geq \cos(t_1)$ then
4              $seed \leftarrow voxel_j$ ;
5              $Q.push(seed)$;
6              while $Q$ is not empty do
7                  $p \leftarrow Q.front,\ Q.pop(),\ Surface_{ij} \leftarrow Surface_{ij} \cup p$;
8                  $\{Neighbor\} \leftarrow neighbor\ voxels\ of\ p$;
9                  Determining whether the $\{Neighbor\}$ belongs to $Surface_{ij}$ (defined in
                     Algorithm 5);
10             end
11             $\{Surfaces\} \leftarrow \{Surfaces\} + Surface_{ij}$;
12         end
13         update $\{Voxel_{cop}\}$;
14     end
15 end
```

As shown in Algorithm 4, for a normal vector $\overrightarrow{n}$, we select the seed voxel in the set of coplanar voxels, carry on the voxel-based region growing and update the set of coplanar voxels in turn. These operations are done iteratively until no seed voxels are found within the t_1 threshold. In this case, a surface cluster with similar normal vector $\overrightarrow{n}$ and its giant component can be obtained simultaneously.

Then the above operations for next major orientation are repeated. In Algorithm 4, the symbol "i" of $Surface_{ij}$ refers to the ID of surface cluster and the symbol "j" refers to the ID of parallel sub-surfaces.

When looking for the neighbor voxels of q, the information of voxelization in Section 3.1 is used. As shown in Figure 3a, voxel q is a big-voxel, voxel A and voxel B are q's surrounding sub-voxels. They are coplanar but not adjacent. Therefore, voxel A and voxel B are not the neighbor voxels of q. In Figure 3b, A, B, C and D are not the neighbor voxels of q as well. The real neighbor voxels of voxel q need to be discussed in a variety of cases.

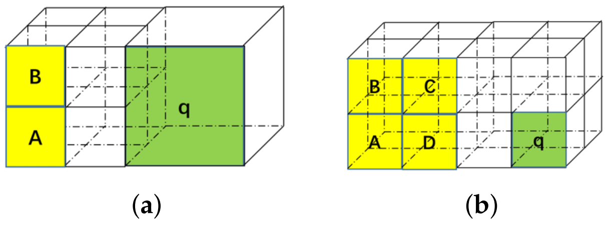

Figure 3. The neighbor voxels of q, (**a**) q is a big-voxel, A and B are sub-voxels. The relationship between A (or B) and q is coplanar but not adjacent; (**b**) None of $\{A, B, C, D\}$ is the neighbor voxel of q.

Algorithm 5 shows the pseudo code of whether the neighbor belongs to $Surface_{ij}$. In Algorithm 4, $\{Neighbor\}$ could be coplanar voxels, sparse voxels or non-coplanar voxels. For sparse voxels and non-coplanar voxels, each point in the voxel is judged by coplanar constraints (i.e., point-based region growing, see lines 2–8 in Algorithm 5).

It is worth noting that, due to the rough and uneven surfaces of rock mass, the coplanar voxels are often not "really" coplanar (as shown in Figure 4a,b). Figure 4c shows the finally result without considering non-coplanar points in coplanar voxels and it is inaccurate. The parameters of $MinScale$ and $MaxScale$ are introduced to overcome this issue, where $MinScale$ and $MaxScale$ are the minimum and maximum percentage of t_2, respectively. For coplanar voxels, the voxel-based region growing is performed when the angle between the normal vectors is less than $t_2 \times MinScale$, and the point-based region growing is performed when the angle between the normal vectors is greater than $t_2 \times MinScale$ and less than $t_2 \times MaxScale$ (see lines 9–21 in Algorithm 5). Figure 4d illustrates the result of considering the non-coplanar points in coplanar voxels.

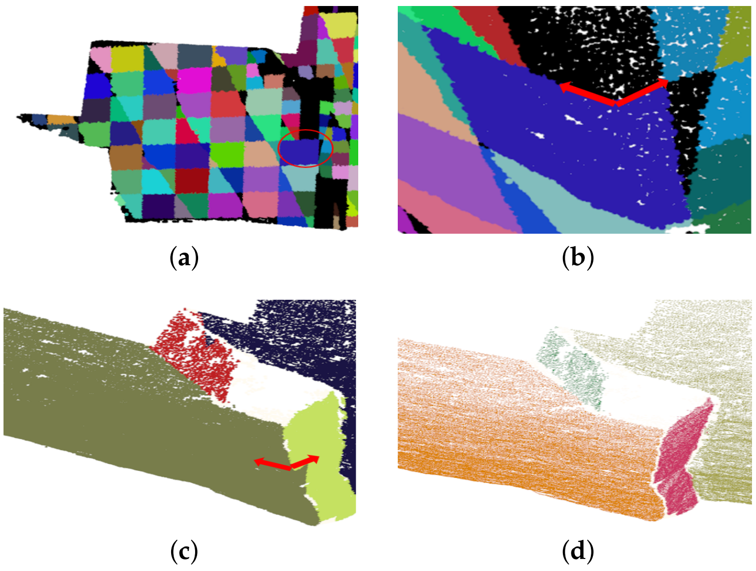

Figure 4. The coplanar voxels may be not "really coplanar". (**a**) The result of clustering; (**b**) The partial enlarged view of (**a**); (**c**) The finally result of surface extraction without considering the non-coplanar points in coplanar voxels; (**d**) The finally result of surface extraction with Algorithm 5.

Algorithm 5: Determining whether the neighbor belongs to $Surface_{ij}$.

```
Input  : $Surface_{ij}$, $Q$, $\{Neighbor\}$, $t_2$, $Max_{dis}$ ;
$\{P_{normal}\}$: the 3D point cloud with normals;
$MinScale$: minimum percentage of $t_2$;
$MaxScale$ : maximum percentage of $t_2$;
$t_3$: maximum angle threshold between the normal vector of point and the normal vector of surface;
Output: $Surface_{ij}$, $Q$;
foreach $v_m$ in $\{Neighbor\}$ do
    if $v_m$ is sparse or non-coplanar voxel then
        foreach point $p_n$ in $v_m$ do
            if $dist \leq Max_{dis}$ && $normal(Surface_{ij}) \cdot normal(p_n) \geq \cos(t_3)$ then
                $Surface_{ij} \leftarrow Surface_{ij} \cup p_n$;
            end
        end
    end
    if $v_m$ is coplanar voxel then
        if $normal(Surface_{ij}) \cdot normal(v_m) \geq \cos(t_2 \times MaxScale)$ && $dist \leq Max_{dis}$ then
            if $normal(Surface_{ij}) \cdot normal(v_m) \geq \cos(t_2 \times MinScale)$ then
                $Surface_{ij} \leftarrow Surface_{ij} \cup v_m$, $Q.push(v_m)$;
            else
                foreach point $p_n$ in $v_m$ do
                    if $normal(Surface_{ij}) \cdot normal(p_n) \geq \cos(t_3)$ && $dist \leq Max_{dis}$ then
                        $Surface_{ij} \leftarrow Surface_{ij} \cup p_n$;
                    end
                end
            end
        end
    end
end
```

4. Experiments and Results

The experimental platform of our method is Microsoft Visual Studio 2013. The PCL [27], a C++ point cloud library is used to do point cloud processing and the Eigen library is used to do matrix operation. The code is performed on an Intel (R) Core i7-4790 3.60 GHz CPU with 4.00 GB RAM. All code is compiled to 32 bit without any parallel operations (such as CUDA based on GPU or OpenMP based on CPU, etc.).

4.1. Datasets

In our study, one synthetic icosahedron point cloud and four rock-mass point clouds were used to conduct the experiments. All of the point clouds used in the experiments are unorganized. The icosahedron point cloud is generated by PCL [27] and all rock-mass point clouds are derived from Rockbench repository [47]. Dataset Rock1–Rock3 come from the first set of data in Rockbench, which are captured by a Leica HDS6000 scanner in Kingston, Canada. Dataset Rock 4 comes from the tenth set of data in Rockbench, which consists of a 3D point cloud on a quartzitic roadcut and be captured by an Optech scanner in Ouray (Colorado). Dataset Rock4 has been used to apply to Rock-mass discontinuity analysis by Riquelme et al. [6] and conduct plane detection to reconstruct the watertight 3D model by Leng et al. [42]. The point density of all point cloud is uneven. The basic information of all point clouds is shown in Table 1. The display of all point clouds is presented in Figure 5. Notice that, the joint faces of Dataset Rock4 are not clear as other rock-mass point clouds.

Table 1. The information of 5 point clouds.

Name	Number of Points	Maximum Point Spacing (m)	Minimum Point Spacing (m)	Average Point Spacing (m)	Bounding Box Size (m)
Icosahedron	372,140	0.1308	2.2×10^{-16}	0.0634	$40 \times 40 \times 40$
Rock1	387,610	0.5167	0.0299	0.0396	$48.86 \times 36.04 \times 14.19$
Rock2	264,309	0.4663	0.0499	0.0639	$51.49 \times 23.94 \times 12.29$
Rock3	1,178,578	0.8016	0.0244	0.0311	$33.75 \times 38.43 \times 27.96$
Rock4	1,024,521	0.1728	0.0009	0.0191	$29.29 \times 25.85 \times 22.88$

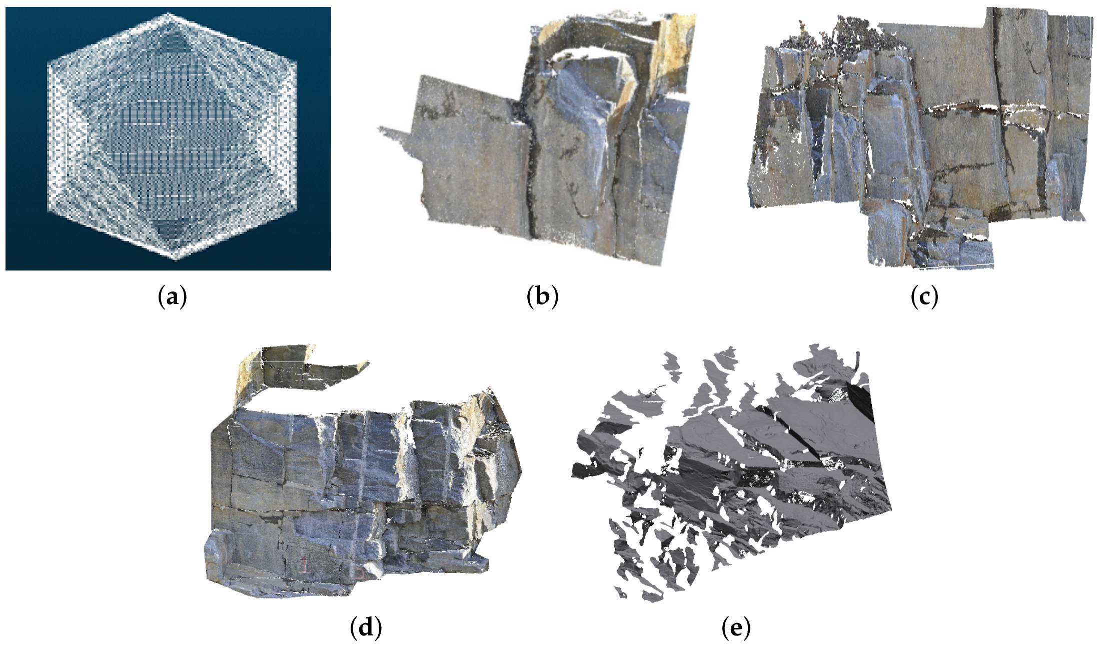

Figure 5. The display of 5 point clouds. (**a**) Icosahedron; (**b**) Rock1; (**c**) Rock2; (**d**) Rock3 and (**e**) Rock4.

4.2. Evaluation Metrics

To evaluate the performance of the proposed method, the evaluation metrics of precision, recall, F1 and execution times [31,48–51] are used to quantitatively evaluate the results of rock surface extraction at the point level. Precision represents the percentage of correctly detected elements and recall indicates the percentage of reference ground truth data that are correctly detected. The F1 score balances precision and recall. If a is an extracted segment and m is a reference segment, the true positive (TP) = $a \cap m$, the false positive (FP) = $a - m$ and the false negative (FN) = $m - a$. The evaluation metrics of precision, recall and F1 are computed as Equations (13)–(15).

$$precision = \frac{|TP|}{|TP| + |FP|} \tag{13}$$

$$recall = \frac{|TP|}{|TP| + |FN|} \tag{14}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{15}$$

4.3. Parameter Turing

The excellent performance of the proposed method depends on the appropriate configuration of all parameters. Table 2 lists the main parameters used in the proposed method. As shown in Table 2, these parameters are divided into three groups.

Table 2. Summary of the parameters.

Stage	ID	Parameters	Meaning	Configuration method
Clustering	1	knn	Neighbors for estimating the normal	Related to the spatial extent of feature surfaces. Set manually.
	2	$voxel_{size}$	The length of big voxel	Related to the spatial extend of feature surfaces. Set manually.
	3	S_α	Allowable maximum ratio about λ_2 and λ_1	20–40, related to the roughness of the surface. Set manually.
	4	ε	Allowable maximum MSE value of coplanar voxels	0.05–0.15, related to the roughness of the surface. Set manually.
	5	$min_{bigvoxel}$	Minimum number of points in a big voxel	Related to the sample densities of the point cloud and $voxel_{size}$. Set manually.
	6	$min_{samllvoxel}$	Minimum number of points in a small voxel	Related to the sample densities of the point cloud and $voxel_{size}$. Set manually.
Major orientation estimation	1	ϕ_{maxnum}	The discretization of ϕ	Default is 180.
	2	θ_{maxnum}	The discretization of θ	Default is 360.
	3	W_a	The weight coefficient of the edge length of voxel	Default is 0.75.
	4	W_d	The weight coefficient of the points contained in voxel	Default is 0.25 $(W_a + W_d = 1)$.
	5	$Size_{win}$	The size of sliding-window	Default is 8.
Rock surface extraction	1	Max_{dis}	Allowable maximum distance between a point to a surface or a surface to another surface	Related to the roughness of the surface. Set manually.
	2	Min_{pla}	Allowable minimum number of points in a surface	Related to the spatial extent of feature surfaces. Set manually.
	3	t_1	Maximum angle between major orientation and the normal vector of voxel	$5°$–$15°$
	4	t_2	Maximum angle between the normal vector of surface and the normal vector of neighbor voxel	$> t_1, 10°$–$20°$
	5	t_3	Maximum angle between the normal vector of point and the normal vector of surface	$> t_2$. Default is $30°$

In clustering stage, the parameters of knn, $voxel_{size}$, S_α, ε, $min_{bigvoxel}$ and $min_{smallvoxel}$ are used to segment the original point cloud into coplanar, non-coplanar and sparse voxels. In addition, the value of $knn, min_{bigvoxel}$ and $min_{smallvoxel}$ depends on the densities of rock-mass point cloud.

Due to the point cloud of rock mass is rough and uneven, the constraints of coplanar clustering are relatively strict or loose depending on the actual situation of the point cloud. In this paper, several guidelines for coplanar constraints are as follows:

- Constraint 1: each "target area" contains at least one coplanar voxel.
- Constraint 2: coplanar constraints should be as strict as possible.
- Constraint 3: the resolution of voxel should be as large as possible.

Because only the sample points within the coplanar voxels could participate in the major orientation estimation (see Section 3.2.3) and the selection of seed voxels (see Section 3.3), the Constraint 1 is established. If no coplanar voxel is detected in coplanar region of the point cloud (i.e., the "target" area in Figure 6a), the final result of rock surface extraction will certainly not be able to extract this "target" area (as shown in Figure 6b). In our tests, the parameters of S_α, ε and $voxel_{size}$ can all affect the finally results in this case, but here, we only adjust the parameters of S_α and ε to explain the efficacy of Constraint 1 (as shown in Figure 6c,d). Constraint 2 is to ensure the coplanarity of coplanar voxels. Constraint 3, which related to the parameter of $voxel_{size}$, is designed to reduce the number of voxels and improve efficiency.

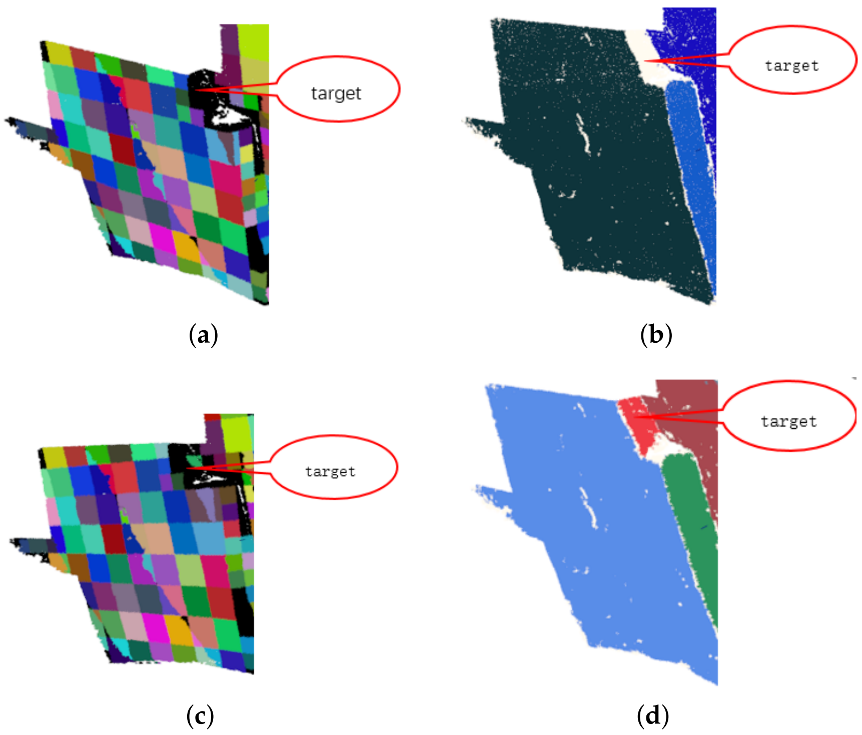

Figure 6. The results of clustering and rock surface extraction with different parameters. (**a**,**b**) Results of clustering and rock surface extraction respectively ($S_\alpha = 25$ and $\varepsilon = 0.05$). (**c**,**d**) Results of clustering and rock surface extraction respectively ($S_\alpha = 22.5$ and $\varepsilon = 0.05$).

Due to the researchers have been focusing on Hough Transform for decades [24,52,53], we just use the default value of parameters in the stage of major orientations estimation.

In the process of region growing, the parameter of Max_{dis} is the distance between a sample point to a surface or a surface to another surface, which related to the roughness of the point cloud. The parameter of Min_{pla} denotes the minimum number of sample points contained in a spatial surface. It related to the sample densities of the point cloud and the spatial extent of feature surfaces at the same time. The parameter of t_1 is the maximum angle between major orientation and the normal vector of voxel. It related to the roughness of the surfaces.

Table 3 shows the configuration of main parameters used in our experiments. Since the surface of the icosahedron point cloud have a large spatial extension, the Min_{pla} value for icosahedron point cloud is greater than others in Table 3. On the other hand, due to the surfaces of the icosahedron point cloud are much flatter than other 4 rock-mass point clouds, its t_1 value is much smaller than others (as shown in Table 3).

Table 3. The parameter setting of our method.

Parameter	Icosahedron	Rock1	Rock2	Rock3	Rock4
knn	50	80	80	80	80
$voxel_{size}$	6	1.73	1.73	1.70	1.50
S_α	40	30	22.5	30	30
ε	0.05	0.05	0.05	0.05	0.05
$min_{samllvoxel}$	150	150	150	300	300
$min_{bigvoxel}$	300	300	300	600	600
Max_{dis}	0.1	0.3	0.3	0.3	0.3
Min_{pla}	2000	900	900	1000	1000
t_1	5°	14°	14°	10°	10°
t_2	17°	15°	15°	15°	15°

4.4. Results for Synthetic Icosahedron Point Cloud

The icosahedron point cloud used here is a synthetic 3D point cloud and would not be affected by real LiDAR problems (i.e., registration, uneven density, etc.). It composes of 20 identical equilateral triangular faces (see Figure 7a). The parameter settings for icosahedron point cloud are shown in Table 3 and the result of our method is shown in Figure 7h.

We compare our method with 4 conventional surface extraction methods and 2 specialized rock surface extraction methods for rock-mass point clouds to validate our method's performance. RHT is the random Hough Transform with a new accumulator design proposed by Borrmann et al. [24]. RG is the typical region growing method of Poppinga et al. [30]. The RHT method and RG method are implemented in C++ code with PCL library. RAN_PCL is the typical RANSAC method [27] and RAN_CGAL is the optimized RANSAC method of Schnabel et al. [54]. The implementation of RAN_PCL and RAN_CGAL is provided by PCL library and CGAL library respectively. HT_RG is a method of combining Hough Transform and Region Growing to detect rock surfaces in rock-mass point clouds [42] and the implementation code is provided by the authors. DSE is an open-source MATLAB software for semi-automatic rock mass joints recognition published by Riquelme et al. [6].

The time consumption of seven methods for icosahedron point cloud is shown in Table 4 and the time unit is second. Obviously, our method is the most efficient. In Figure 7, The results of (a)–(h) illustrate that each method can extract the surfaces correctly.

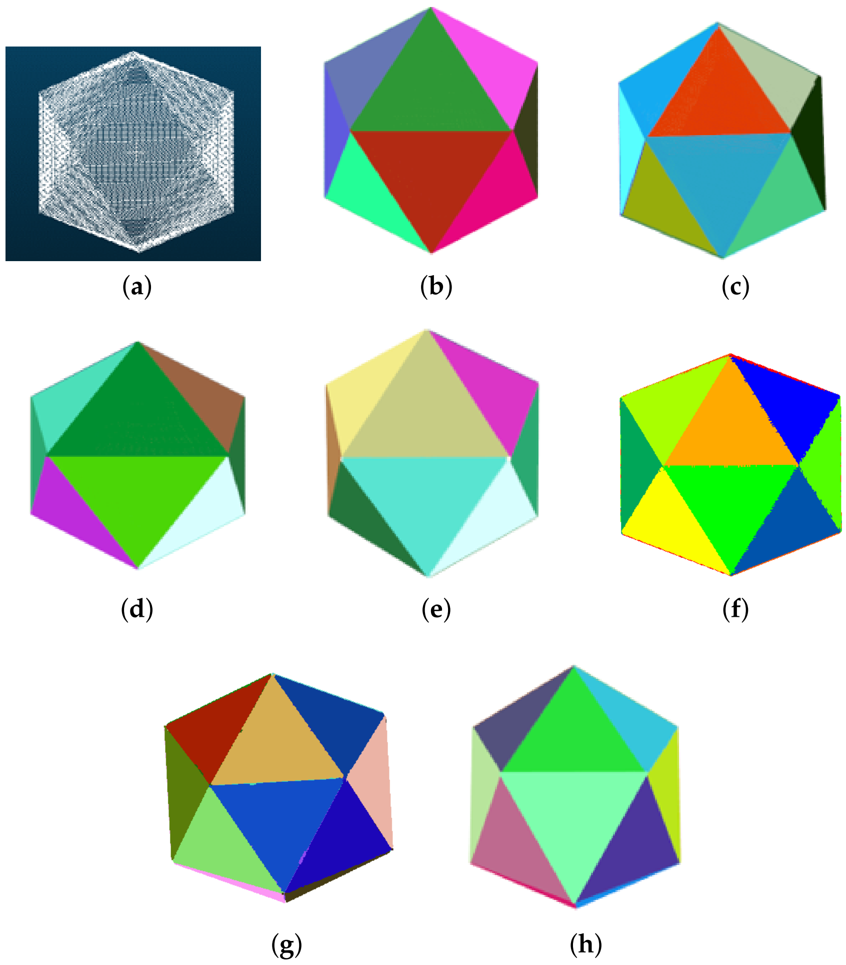

Figure 7. The feature surface extraction results of 7 methods for icosahedron. (**a**) Original icosahedron point cloud; (**b**–**h**) Results of RHT, RG, RAN_PCL, RAN_CGAL, DSE, HT-RG and our method respectively.

Table 4. Execution time of 7 different methods for icosahedron (in seconds).

Method	Time (s)	Number of Detected Surfaces
RHT	2.083	20
RG	7.678	20
RAN_PCL	19.068	20
RAN_CGAL	1.802	20
DSE	90.810	20
HT-RG	8.780	20
Ours	0.246	20

4.5. Results for Real Rock-Mass Point Clouds

Since the proposed method has been applied using a synthetic icosahedron point cloud, the next step will be to verify with real rock-mass point clouds. Here we used a total of 4 sets of rock-mass point clouds for testing and comparing the results with other 6 methods.

4.5.1. Qualitative Comparison

The rock surface extraction results of Rock1 by different methods are shown in Figure 8. Notice that, we use red polygons to point out the deficiencies of each result. The results of RHT and RAN_CGAL have the phenomenons of over-segmentation and discontinuity (as shown in Figure 8b,e). Figure 8c shows that the local surfaces of RG are incorrect and over-segmented. The result of RAN_PCL has the problems of over-segmentation, discontinuity and bad boundary (as shown in Figure 8d). The HT_RG and DSE method have the phenomenons of over-segmentation and missing some rock surfaces (see Figure 8f,g). Visually speaking, Figure 8h shows that the result of our method has better continuity and planar boundary without over-segmentation. For statistical analysis, please refer to Section 4.5.2.

We also test all methods on 3 other Rock-Mass point clouds. As shown in Figure 9, the first row represents 3 different Rock-Mass point clouds, and other rows represent the results of rock surface extraction by each method on 3 datasets, respectively. The results are visual similar, but our method is significantly faster.

4.5.2. Quantitative Comparison

To quantitatively evaluate the results of rock surface extraction, we manually split the Rock1 point cloud into a reference point cloud containing 20 surfaces by using the CloudCompare software (see Figure 10). In Figure 10b, the black part represents sample points without coplanar feature and each of the other colors represents a separated surface. Since the rock mass itself is rough and bumpy, there may be cases that points with coplanar characteristics outside the reference surfaces are not classified as rock surfaces. We compare and analyze the time consumption and the correctness of rock surface extraction in the following part.

The time consumption of different methods for Rock1 is shown in Table 5. It can be clearly seen from the table that the efficiency of our method is much higher than other methods, especially for the method of HT-RT and DSE. Table 5 also shows the number of detected surfaces of our method is fewer than other methods. The main reason is that the results of other methods have the phenomenon of over-segmentation (see Figure 8). The reference point cloud was labeled with 20 surfaces, and our method finally detected 23 surfaces. First of all, this is because our method has detected all the surfaces being labeled (see Figure 8h). Secondly, since the surfaces of the rock-mass point cloud are rough and uneven, we only mark up the surfaces with obvious coplanar characteristics (see Figure 10).

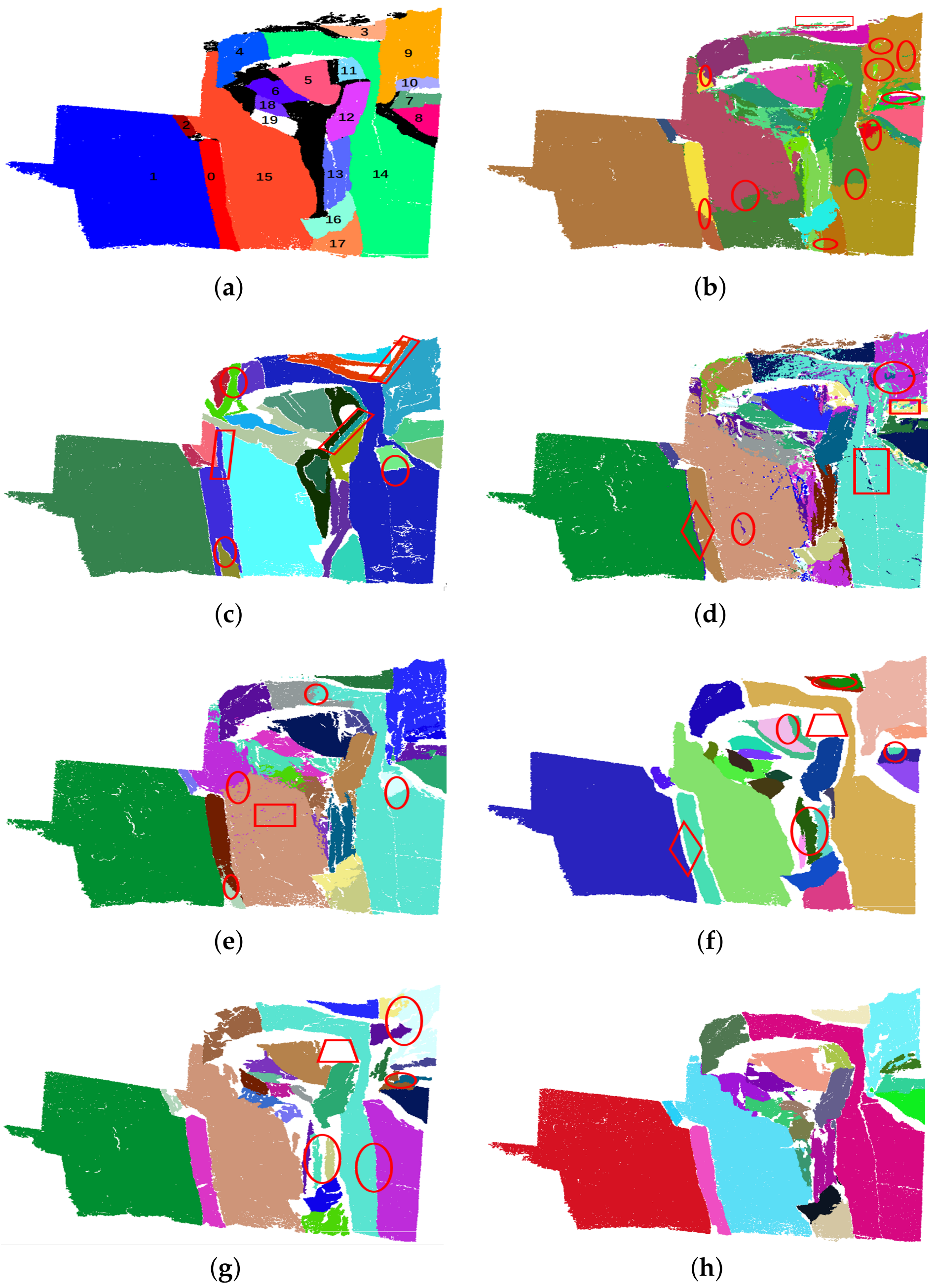

Figure 8. The rock surface extraction results of seven different methods for Rock1. (**a**) The reference point cloud of Rock1. (**b**–**h**) Results of RHT, RG, RAN_PCL, RAN_CGAL, HT_RG, DSE and ours, respectively. Each color represents a surface. Notice that, we use red polygons to point out the deficiencies of each result. Ellipse indicates over-segmentation, rectangle represents discontinuity, parallelogram represents incorrect result, rhombus represent bad boundary and trapezoid represents missing detection.

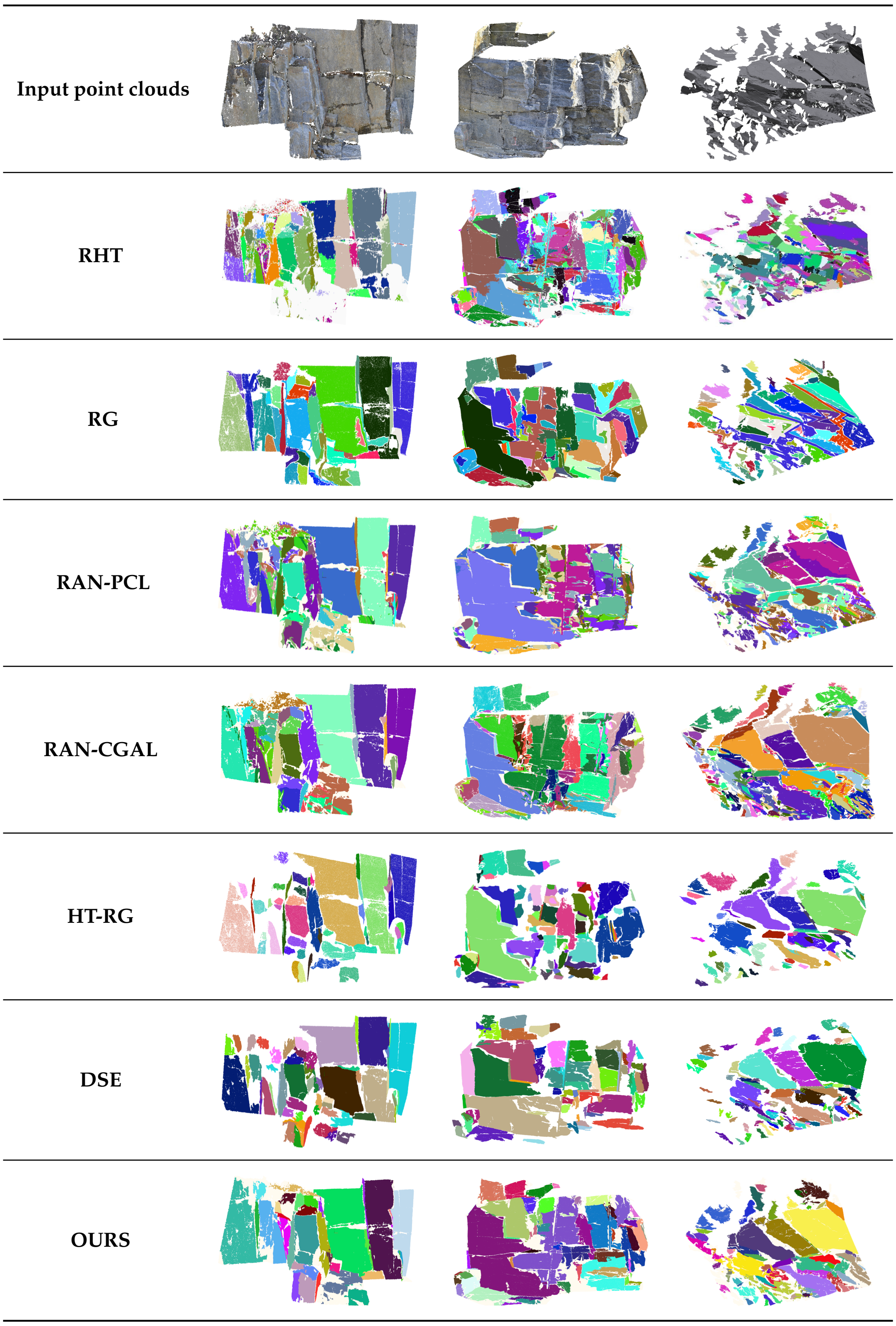

Figure 9. The results of different methods for three other Rock-Mass point clouds.

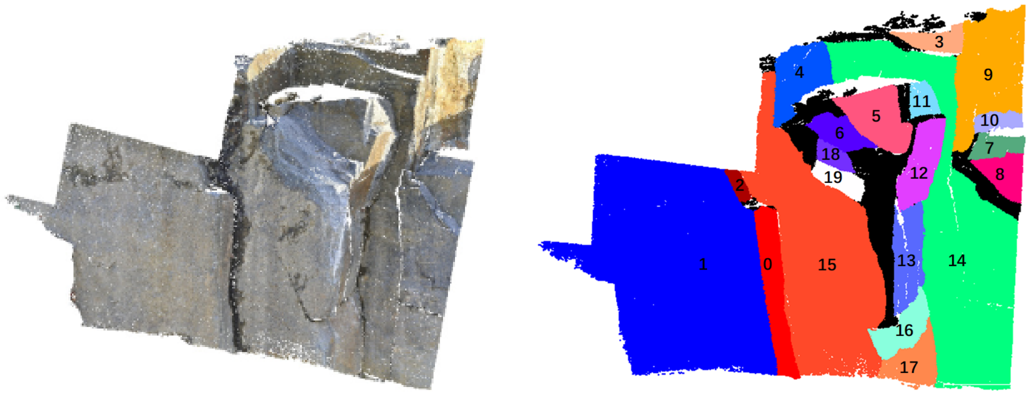

Figure 10. The reference point cloud of rock mass. (**a**) The original point cloud of Rock1. (**b**) The reference point cloud of Rock1. In addition to black, each of the other colors represents a separated surface.

Table 5. Execution time of 7 different methods for Rock1 (in seconds).

Method	Time (s)	Number of Detected Surfaces
RHT	7.312	25
RG	15.122	25
RAN_PCL	22.342	25
RAN_CGAL	2.869	26
HT_RG	59.606	29
DSE	108.684	30
Ours	0.445	23

In this paper, the time distribution of our method for icosahedron point cloud and rock-mass point clouds is shown in Table 6. We can find that the major orientation estimation and region growing are more time-consuming. The efficiency of major orientation estimation can be greatly increased by using one standard deviation to calculate the voting thresholds (see Section 3.2.3). The main reason of time-consuming in region growing stage is that points in coplanar voxels may not fully exhibit coplanar features (see Sections 3.1 and 3.3).

Table 6. Time distribution of our method (in seconds).

Date	Number of Points	Clustering	Major Orientation Estimation	Region Growing	Total
Icosahedron	372,140	0.03644	0.08542	0.12414	0.24600
Rock1	387,610	0.05572	0.12329	0.26621	0.44522
Rock2	264,309	0.06060	0.12159	0.24291	0.42510
Rock3	1,178,578	0.11082	0.18914	1.36345	1.66341
Rock4	1,024,521	0.07757	0.13365	0.51241	0.72363

While all the results of rock surface extraction are visually similar, We make use of three evaluation metrics to quantify the accuracy of the surface extraction results. As shown in Table 7, the precision, recall and F1 scores of our method are all the highest. Notice that, the recall scores of the other 6 methods are much lower than ours. This means that the other methods excluded many points near the boundary of feature surfaces (see Figure 8). Figures 11 and 12 show that the F1 score and recall score for each surface of our method is the highest.

Table 7. Rock1's accuracy.

Method	Precision	Recall	F1
RHT	74.23%	77.53%	75.85%
RG	77.98%	76.23%	77.10%
RAN-PCL	81.70%	81.17%	81.43%
RAN-CGAL	85.42%	83.66%	84.53%
HT-RG	91.18%	86.37%	88.71%
DSE	81.28%	74.77%	77.89%
OURS	91.92%	91.67%	91.80%

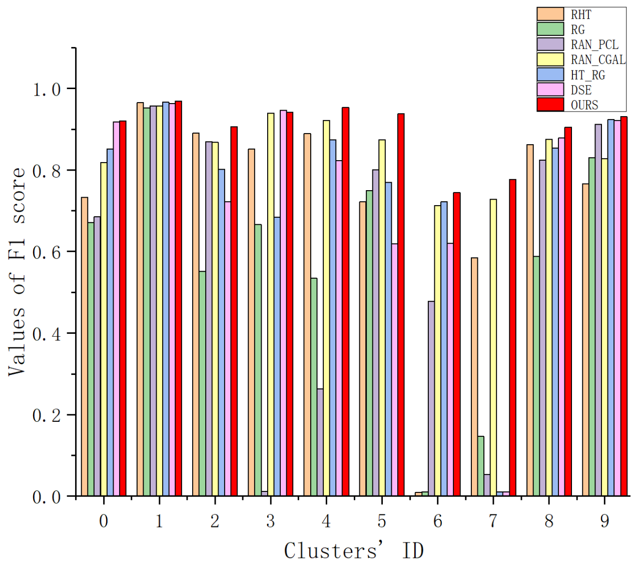

Figure 11. F1 scores of Rock1's feature surfaces for different methods.

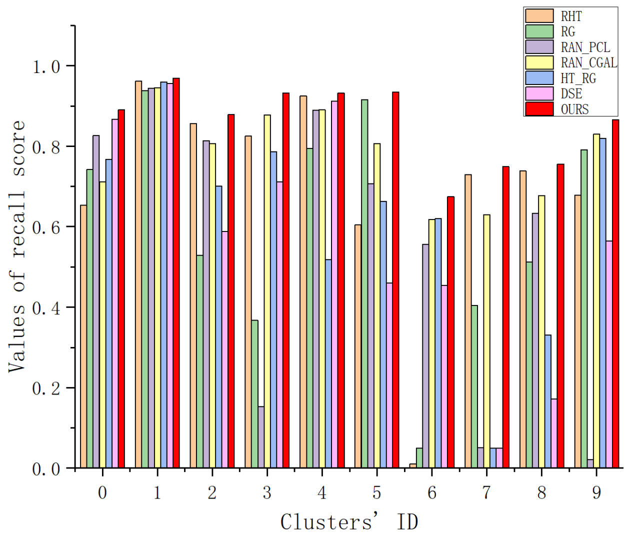

Figure 12. Recall scores of Rock1's feature surfaces for different methods.

In our experimental tests, the HT-RG method performs better than other conventional feature surface extraction methods of urban point clouds in terms of accuracy. But as a whole, the proposed method performs better than the HT-RG method in terms of efficiency and accuracy.

5. Conclusions

This paper focuses on rock surface extraction in unorganized point clouds of rock mass. An efficient and accurate rock surface extraction method was proposed to deal with the challenge of roughness and unevenness in rock-mass point clouds. We segment point clouds via a framework with three stages, i.e., (i) clustering based on voxels; (ii) estimating major orientations based on Gaussian Kernel and (iii) rock surface extraction. To avoid the problem of computational complexity, the two-scale spatial grid and a bivariate Gaussian Kernel are used to estimate major orientations for selecting seed voxels. The voxel-based region growing is performed to ensure the connectivity of rock surfaces. Eventually, sets of parallel surface clusters and their giant component are obtained with high efficiency.

The proposed method was benchmarked with one synthetic icosahedron point cloud and four rock-mass point clouds. Overall, a high level of detail with better boundary in the rock surface extraction was achieved. When compared to ground-truth data of Rock1 by using the evaluation metrics of precision, recall and F1, the each score of our method was more than 91%. Additional, the proposed method is several orders of magnitude faster than other techniques for rock surface extraction in rock-mass point cloud. The experiments showed the availability and efficiency of the proposed method.

However, there is a limitation exists in our method, i.e., clustering based on voxels adapts fixed resolution for each voxel. As a result, the voxels may not retain well the boundaries of the point cloud. In future work, we will try to solve this problem with adaptive resolutions (such as supervoxel [48,55,56]).

Author Contributions: Conceptualization, L.L.; Funding acquisition, J.X.; Project administration, J.X.; Supervision, J.X. and Y.W.; Validation, L.L., J.X. and Y.W.; Visualization, L.L.; Writing—original draft, L.L.; Writing—review & editing, J.X. and Y.W.

Funding: This research was funded by the National Natural Science Foundation of China (grant no. 61471338), the Youth Innovation Promotion Association CAS (2015361), the Key Research Program of Frontier Sciences CAS (QYZDY–SSW–SYS004), the Beijing Nova Program (Z171100001117048), and the Beijing Science and Technology Project (Z181100003818019).

Acknowledgments: The authors would like to thank the reviewers for their valuable comments and suggestions..

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ochmann, S.; Vock, R.; Wessel, R.; Klein, R. Automatic reconstruction of parametric building models from indoor point clouds. *Comput. Graph.* **2016**, *54*, 94–103. [CrossRef]
2. Gupta, S.; Arbeláez, P.; Girshick, R.; Malik, J. Indoor Scene Understanding with RGB-D Images: Bottom-up Segmentation, Object Detection and Semantic Segmentation. *Int. J. Comput. Vis.* **2015**, *112*, 133–149. [CrossRef]
3. Guo, Y.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2270–2287. [CrossRef]
4. Chekhlov, D.; Gee, A.P.; Calway, A.; Mayol-Cuevas, W. Ninja on a Plane: Automatic Discovery of Physical Planes for Augmented Reality Using Visual SLAM. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Washington, DC, USA, 13–16 November 2007; pp. 1–4.
5. Yan, W.Y.; Shaker, A.; El-Ashmawy, N. Urban land cover classification using airborne LiDAR data: A review. *Remote Sens. Environ.* **2015**, *158*, 295–310. [CrossRef]
6. Riquelme, A.J.; Abellán, A.; Tomás, R.; Jaboyedoff, M. A new approach for semi-automatic rock mass joints recognition from 3D point clouds. *Comput. Geosci.* **2014**, *68*, 38–52. [CrossRef]
7. Hough, V.; Paul, C. Method and Means for Recognizing Complex Patterns. U.S. Patent US1771560A, 18 December 1962.

8. Fischler, M.A.; 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. [CrossRef]
9. Chen, D.S. A data-driven intermediate level feature extraction algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *11*, 749–758. [CrossRef]
10. Besl, P.J.; Jain, R.C. Segmentation through variable-order surface fitting. *IEEE Trans. Pattern Anal. Mach. Intell.* **1988**, *10*, 167–192. [CrossRef]
11. Duda, R.O.; Hart, P.E. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM* **1972**, *15*, 11–15. [CrossRef]
12. Dube, D.; Zell, A. Real-time plane extraction from depth images with the randomized hough transform. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 1084–1091.
13. Fujiwara, T.; Kamegawa, T.; Gofuku, A. Plane detection to improve 3D scanning speed using RANSAC algorithm. In Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), Melbourne, VIC, Australia, 19–21 June 2013; pp. 1863–1869.
14. Masuta, H.; Makino, S.; Lim, H. 3D plane detection for robot perception applying particle swarm optimization. In Proceedings of the 2014 World Automation Congress (WAC), Waikoloa, HI, USA, 3–7 August 2014; pp. 549–554.
15. Lato, M.J.; Vöge, M. Automated mapping of rock discontinuities in 3D lidar and photogrammetry models. *Int. J. Rock Mech. Min. Sci.* **2012**, *54*, 150–158. [CrossRef]
16. Abellán, A.; Oppikofer, T.; Jaboyedoff, M.; Rosser, N.J.; Lim, M.; Lato, M.J. Terrestrial laser scanning of rock slope instabilities. *Earth Surf. Process. Landf.* **2014**, *39*, 80–97. [CrossRef]
17. Bieniawski, Z.T. Engineering Rock Mass Classifications: A complete manual for engineers and geologists in mining, civil and petroleum engineering. *Petroleum* **1989**, *251*, 357–365.
18. Goodman, R.E.; Shi, G.H. Block Theory and Its Application to Rock Engineering. *Prentice Hall* **1985**, *26*, 103–105.
19. Shi, G.H. Discontinuous deformation analysis: A new numerical model for the statics and dynamics of deformable block structures. *Eng. Comput.* **1992**, *9*, 157–168. [CrossRef]
20. Ma, G.; An, X.; He, L. The numerical manifold method: A review. *Int. J. Comput. Methods* **2010**, *7*. [CrossRef]
21. Vosselman, G.; Gorte, B.G.; Sithole, G.; Rabbani, T. Recognising structure in laser scanner point clouds. *Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci.* **2004**, *46*, 33–38.
22. Limberger, F.A.; Oliveira, M.M. Real-time detection of planar regions in unorganized point clouds. *Pattern Recognit.* **2015**, *48*, 2043–2053. [CrossRef]
23. Xiao, J.; Zhang, J.; Adler, B.; Zhang, H.; Zhang, J. Three-dimensional point cloud plane segmentation in both structured and unstructured environments. *Robot. Auton. Syst.* **2013**, *61*, 1641–1652. [CrossRef]
24. Borrmann, D.; Elseberg, J.; Lingemann, K.; Nüchter, A. The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design. *3D Res.* **2011**, *2*, 3. [CrossRef]
25. Huang, H.; Brenner, C. Rule-based roof plane detection and segmentation from laser point clouds. In Proceedings of the 2011 Joint Urban Remote Sensing Event, Munich, Germany, 11–13 April 2011; pp. 293–296.
26. Hulik, R.; Spanel, M.; Smrz, P.; Materna, Z. Continuous plane detection in point-cloud data based on 3D Hough Transform. *J. Vis. Commun. Image Represent.* **2014**, *25*, 86–97. [CrossRef]
27. Rusu, B.R.; Cousins, S. 3D is here: Point Cloud Library (PCL). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4.
28. Fernandes, L.A.; Oliveira, M.M. Real-time line detection through an improved Hough transform voting scheme. *Pattern Recognit.* **2008**, *41*, 299–314. [CrossRef]
29. Trevor, A.J.B.; Gedikli, S.; Rusu, R.B.; Christensen, H.I. Efficient organized point cloud segmentation with connected components. In Proceedings of the 3rd Workshop on Semantic Perception, Mapping and Exploration (SPME), Karlsruhe, Germany, 5 May 2013.
30. Poppinga, J.; Vaskevicius, N.; Birk, A.; Pathak, K. Fast plane detection and polygonalization in noisy 3D range images. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3378–3383.
31. Vo, A.V.; Truong-Hong, L.; Laefer, D.F.; Bertolotto, M. Octree-based region growing for point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 88–100. [CrossRef]

32. Yang, M.Y.; Förstner, W. Plane detection in point cloud data. In Proceedings of the 2nd International Conference on Machine Control Guidance, Bonn, Germany, 9–11 March 2010; Volume 1, pp. 95–104.
33. Xu, B.; Jiang, W.; Shan, J.; Zhang, J.; Li, L. Investigation on the Weighted RANSAC Approaches for Building Roof Plane Segmentation from LiDAR Point Clouds. *Remote Sens.* **2016**, *8*, 5. [CrossRef]
34. Li, L.; Yang, F.; Zhu, H.; Li, D.; Li, Y.; Tang, L. An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells. *Remote Sens.* **2017**, *9*, 433. [CrossRef]
35. Holz, D.; Holzer, S.; Rusu, R.B.; Behnke, S. Real-time plane segmentation using RGB-D cameras. In *Robot Soccer World Cup*; Springer: Berlin, Germany, 2011; pp. 306–317.
36. Yamazaki, I.; Natarajan, V.; Bai, Z.; Hamann, B. Segmenting point-sampled surfaces. *Visual Comput.* **2010**, *26*, 1421–1433. [CrossRef]
37. Jaboyedoff, M.; Metzger, R.; Oppikofer, T.; Couture, R.; Derron, M.H.; Locat, J.; Turmel, D. New Insight Techniques to Analyze Rock-slope Relief Using DEM And 3Dimaging Cloud Points: COLTOP-3D Software. In *Rock Mechanics: Meeting Society's Challenges and Demands*; CRC Press: Boca Raton, FL, USA, 2007; Volume 1, pp. 61–68.
38. Slob, S.; van Knapen, B.; Hack, R.; Turner, K.; Kemeny, J. Method for Automated Discontinuity Analysis of Rock Slopes with Three-Dimensional Laser Scanning. *Transp. Res. Record J. Transp. Res. Board* **2005**, *1913*, 187–194. [CrossRef]
39. Gigli, G.; Casagli, N. Semi-automatic extraction of rock mass structural data from high resolution LIDAR point clouds. *Int. J. Rock Mech. Min. Sci.* **2011**, *48*, 187–198. [CrossRef]
40. Riquelme, A.J.; Abellán, A.; Tomás, R. Discontinuity spacing analysis in rock masses using 3D point clouds. *Eng. Geol.* **2015**, *195*, 185–195. [CrossRef]
41. Riquelme, A.; Tomás, R.; Cano, M.; Pastor, J.L.; Abellán, A. Automatic Mapping of Discontinuity Persistence on Rock Masses Using 3D Point Clouds. *Rock Mech. Rock Eng.* **2018**, *51*, 3005–3028. [CrossRef]
42. Leng, X.; Xiao, J.; Wang, Y. A multi-scale plane-detection method based on the Hough transform and region growing. *Photogramm. Record* **2016**, *31*, 166–192. [CrossRef]
43. Hansen, P.; Mladenović, N.; Todosijević, R.; Hanafi, S. Variable neighborhood search: Basics and variants. *EURO J. Comput. Optim.* **2017**, *5*, 423–454. [CrossRef]
44. Allheeib, N.; Islam, M.S.; Taniar, D.; Shao, Z.; Cheema, M.A. Density-based reverse nearest neighbourhood search in spatial databases. *J. Ambient Intell. Humaniz. Comput.* **2018**, 1–12. [CrossRef]
45. Wernimont, G. Probability and Experimental Errors in Science. An Elementary Survey. *Math. Gaz.* **1961**, *57*, 148. [CrossRef]
46. Tong, Y.L. *The Multivariate Normal Distribution*; Springer Series in Statistics; Springer: New York, NY, USA, 1990.
47. Lato, M.; Kemeny, J.; Harrap, R.; Bevan, G. Rock bench: Establishing a common repository and standards for assessing rockmass characteristics using LiDAR and photogrammetry. *Comput. Geosci.* **2013**, *50*, 106–114. [CrossRef]
48. Lin, Y.; Wang, C.; Chen, B.; Zai, D.; Li, J. Facet Segmentation-Based Line Segment Extraction for Large-Scale Point Clouds. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4839–4854. [CrossRef]
49. Awrangjeb, M.; Fraser, C.S. An Automatic and Threshold-Free Performance Evaluation System for Building Extraction Techniques From Airborne LIDAR Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 4184–4198. [CrossRef]
50. Yan, J.; Shan, J.; Jiang, W. A global optimization approach to roof segmentation from airborne lidar point clouds. *ISPRS J. Photogramm. Remote Sens.* **2014**, *94*, 183–193. [CrossRef]
51. Ni, H.; Lin, X.; Zhang, J. Classification of ALS point cloud with improved point cloud segmentation and random forests. *Remote Sens.* **2017**, *9*, 288. [CrossRef]
52. Mukhopadhyay, P.; Chaudhuri, B.B. A survey of Hough Transform. *Pattern Recog.* **2015**, *48*, 993–1010. [CrossRef]
53. Hassanein, A.S.; Mohammad, S.; Sameer, M.; Ragab, M.E. A Survey on Hough Transform, Theory, Techniques and Applications. *Int. J. Comput. Sci. Issues (IJCSI) Mahebourg* **2015**, *12*, 139–156.
54. Schnabel, R.; Wahl, R.; Klein, R. Efficient RANSAC for point-cloud shape detection. *Comput. Graph. Forum* **2007**, *26*, 214–226. [CrossRef]

55. Lin, Y.; Wang, C.; Zhai, D.; Li, W.; Li, J. Toward better boundary preserved supervoxel segmentation for 3D point clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 39–47. [CrossRef]
56. Papon, J.; Abramov, A.; Schoeler, M.; Worgotter, F. Voxel cloud connectivity segmentation—Supervoxels for point clouds. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2027–2034.