Available online at www.sciencedirect.com

ScienceDirect

Underground Space 25 (2025) 218–238

www.keaipublishing.com/undsp

Research Paper

AOC: An adaptive oriented contraction method for automatic trace recognition of rock tunnel excavation face based on 3D point cloud

Keshen Zhang [a,b], Min Zhang [a,c,*], Lianyang Zhang [a,d], Wei Wu [a,e]

[a] College of Civil Engineering, Tongji University, Shanghai 200092, China
[b] College of Civil Engineering, Qingdao University of Technology, Qingdao 266033, China
[c] Shudao Investment Group Co., Ltd., Chengdu 610095, China
[d] Department of Civil and Architectural Engineering and Mechanics, The University of Arizona, Tucson, AZ 85281, USA
[e] State Key Laboratory of Disaster Reduction in Civil Engineering, Tongji University, Shanghai 200092, China

Received 23 July 2024; received in revised form 7 October 2024; accepted 24 November 2024
Available online 11 September 2025

Abstract

Trace recognition is essential for rock discontinuity characterization of tunnel excavation faces. Traditional methods of trace identification based on three-dimensional (3D) point cloud curvatures require manual fine-tuning for curvature detection and lack consistency with orientation grouping results. This paper proposes a new automatic method for trace identification from 3D point cloud. An adaptive vector method based on neighbor assignment is proposed to accurately generate both normal vectors and directional vectors on sharp points. A principal component analysis-based oriented contraction (PWI-OC) method is presented to extract point cloud skeletons with good iterative conformality. A sparse growing method is proposed to generate extensive trace segments. Two rock excavation face cases, from a mining tunnel and a railway tunnel, are adopted for analysis. The significance of adaptive normal vectors is validated for improving the quality of orientation grouping, and the iterative conformality of PWI-OC is validated to generate more accurate and robust trace skeletons than the traditional method. The results show that the proposed method can achieve a more accurate trace identification than traditional methods, consistent with orientation grouping results, robust to overlapping traces, and automates curvature point detection.

Keywords: Rock tunnel; Excavation face; Discontinuity trace; 3D point cloud; Adaptive vector; Oriented contraction

1 Introduction

Discontinuity properties largely influence the mechanical behaviors, such as strength, deformability, and permeability of rock mass (Barton, 1978; Daghigh et al., 2022). Discontinuity trace description (e.g., trace length estimation) is essential for rock discontinuity characterizations (Mauldon, 1998; Zhang & Einstein, 1998; Umili et al., 2013; Li et al., 2014, 2016; Q. Zhang et al., 2021, 2022, 2023, 2024). The traditional discontinuity mapping requires manual contact measurements by geo-engineers. However, the manual measurement is often time-consuming, dangerous, and difficult to perform in inaccessible regions, and the accuracy is influenced by subjective experience (Kemeny & Post, 2003; Lato et al., 2012; Vöge et al., 2013; Assali et al., 2016; Guo et al., 2019). Recently, remote sensing technologies, such as photogrammetry (Sturzenegger & Stead, 2009; Assali et al., 2016) and laser scanning (Fisher et al., 2014; Riquelme et al., 2014), have been applied for the description of rock discontinuities. Discontinuity information can be extracted from the

* Corresponding author at: College of Civil Engineering, Tongji University, Shanghai 200092, China.
E-mail address: zhangmin@shudaojt.com (M. Zhang).
Peer review under the responsibility of Tongji University

https://doi.org/10.1016/j.undsp.2024.11.005

images or reconstructed three-dimensional (3D) point cloud of rock mass without direct access to the rock mass, which not only improves the efficiency and safety in data acquisition but also produces more objective results.

The two-dimensional (2D) image processing is one of the main methods for trace detection (Battulwar et al., 2021; Daghigh et al., 2022). Edge detection or line detection algorithms are often used to extract traces as intersection lines between discontinuity planes and the rock face plane based on pixel intensity variations (Crosta, 1997; Lemy & Hadjigeorgiou, 2003; Bolkas et al., 2018). However, the diversity of rock textures, illumination conditions, and threshold settings can cause difficulty in trace detection during image processing (Ferrero et al., 2009; Li et al., 2016). In addition, 2D approaches also suffer from the inaccuracy of the fitted discontinuity plane, the division of a single trace into several broken fragments, and the generation of redundant trace segments caused by irregular rock surfaces (Kemeny et al., 2006; Battulwar et al., 2021).

Recently, trace recognition methods based on 3D point cloud have also been widely studied (Roncella et al., 2005; Gigli & Casagli, 2011; Battulwar et al., 2021; Daghigh et al., 2022). Representatively, Umili et al. (2013) proposed a curvature-based method for extracting traces from the digital surface model (DSM) of rock mass. The vertices with large principal curvatures were selected and connected as concave edge paths of discontinuities using the random sample consensus (RANSAC). Then the edge paths were clustered into traces using the iterative self-organizing data analysis technique algorithm (ISODATA) (Ball & Hall, 1965). Cao et al. (2017) extracted the edges of triangular elements as discontinuity traces based on the angle variation between adjacent triangular units. Li et al. (2019) first extracted curvature points from rock mass DSM using the normal vector voting (NTV) theory. Then a growing method is used to connect curvature points as trace segments. Zhang et al. (2020) proposed a Laplacian-based contraction method to extract curvature point skeletons for trace segment generation. However, these methods require the tedious and time-consuming triangulation of raw points, and the curvature detection is sensitive to the user-defined threshold. In addition, Guo et al. (2018) combined the one-dimensional (1D) truncated Fourier series and the principal components analysis (PCA) to select trace points. A Laplacian smoothing method was used to extract thinned trace points for trace segment generation. Although this method can directly process point clouds, the curvature detection is still dependent on the user-defined threshold. In addition, the smoothing method is not controlled with smoothing directions, making the thinned points discrete to generate broken segments.

Point cloud vector calculation is one of the essential factors in discontinuity analysis such as the discontinuity orientation recognition based on normal vectors (Priest, 1993; Jimenez, 2008; Riquelme et al., 2014; Gao et al., 2019; Li et al., 2019; Wu et al., 2020; Cui & Yan, 2020; Singh et al., 2021, 2022) or the curvature detection in discontinuity trace recognition based on normal vector variations (Umili et al., 2013; Li et al., 2016; Cao et al., 2017; Guo et al., 2018; Zhang et al., 2020). The PCA method was one of the commonly used methods in normal vector calculation, and was accurate and fast, especially on smooth surfaces (Daghigh et al., 2022). However, the initial PCA did not perform well for point clouds with noise and regions near geometric singularities (Nurunnabi et al., 2015; Khaloo & Lattanzi, 2017). Although some robust estimation methods were used to improve the normal vector accuracy near the singularities in discontinuity orientation analysis (Wang et al., 2013; Kong et al., 2020), the robust estimation of directional vectors was not fully studied and used in discontinuity trace recognition.

This paper proposed an adaptive oriented contraction (AOC) method for automatic recognition of rock discontinuity trace based on 3D point cloud. An adaptive vector calculation based on iterative neighbor assignments of point clouds was proposed to accurately produce both normal vectors for orientation grouping and directional vectors for trace recognition. The sharp points of traces were extracted from the orientation grouping results generated by a fast optimal grouping method. A PCA-weighted iterative oriented contraction (PWI-OC) method was proposed to extract sharp point skeletons. A sparse growing method was proposed to generate trace segments from the sharp point skeletons.

The paper is organized as follows: an introduction to discontinuity trace recognition is presented in Section 1, the specific description of the methodology is presented in Section 2, several cases are analyzed in Section 3, the discussion of the proposed method is in Section 4, and some conclusions are drawn in Section 5.

2 Methodology

This paper proposed an adaptive oriented contraction method for automatic recognition of rock discontinuity traces based on 3D point clouds. The main flow chart is shown in Fig. 1. In Section 2.1, an adaptive vector method based on neighbor iterative assignment is proposed to accurately generate both normal vectors and directional vectors, especially on sharp points. In Section 2.2, a fast-optimal grouping of orientation is first performed based on the adaptive normal vectors of the point cloud. Then, the PWI-OC method is proposed to generate skeleton points from the boundary points (with adaptive directional vectors) of the optimal orientation grouping result. In Section 2.3, a sparse growing method is proposed to generate trace segments, which includes the initial generation of trace segments based on the neighbor connection of skeleton points, the trace segment linearization based on the sparse growing method, and the trace segment connection based on oriented extension.

In order to illustrate each step of the proposed method, a rock excavation face of a railway tunnel located in western China is adopted as case 1 for analysis. A Xiaomi Mi

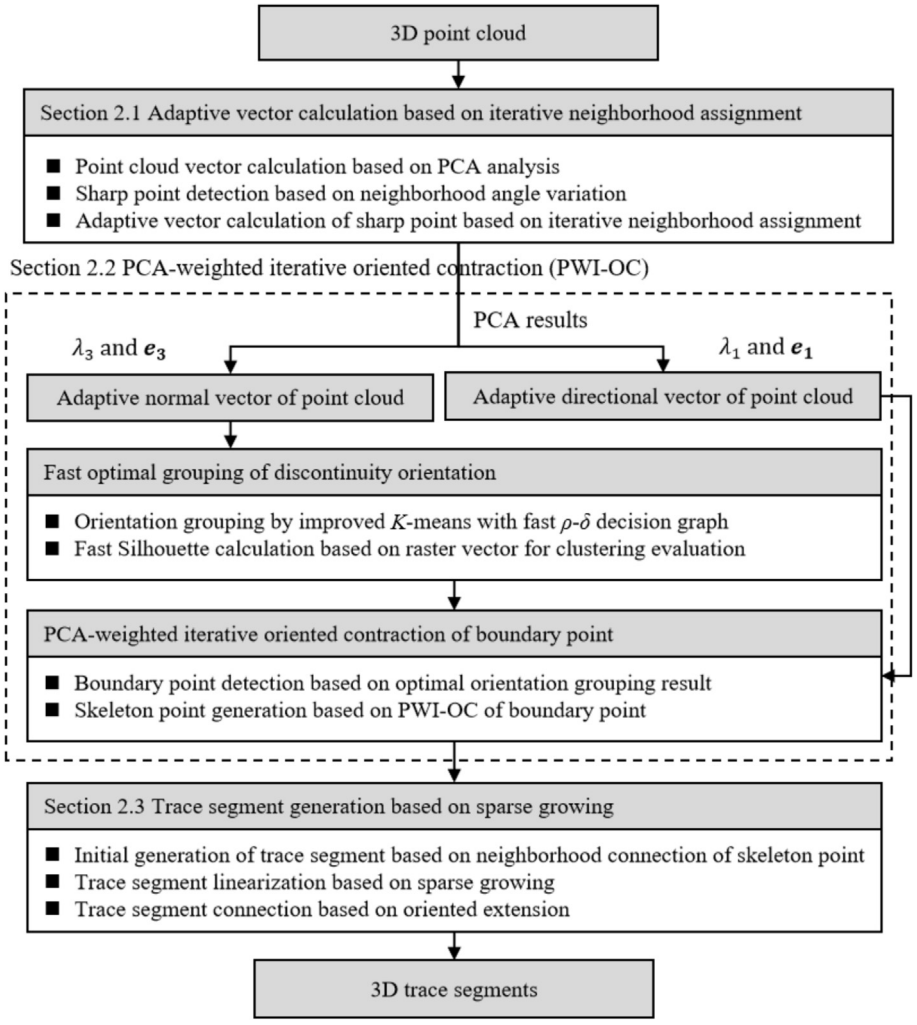

Fig. 1. Flow chart of the proposed method.

10 mobile phone was used to take 6 images (Fig. 2(a)–(f)) with the resolution of 5760 × 3240 at different positions in front of the excavation face. The Meshroom open-source computer vision software was used to reconstruct 3D RGB point clouds from the image sequence, generating a total of 1 412 061 points. The region of interest (ROI) (Fig. 2(g)) includes 341 348 points after manual cropping.

2.1 Adaptive vector calculation based on iterative neighbor assignment

Point cloud vector calculation is essential for rock discontinuity identification (Priest, 1993; Umili et al., 2013; Riquelme et al., 2014; Singh et al., 2021). Although PCA can accurately perform normal vector estimation on

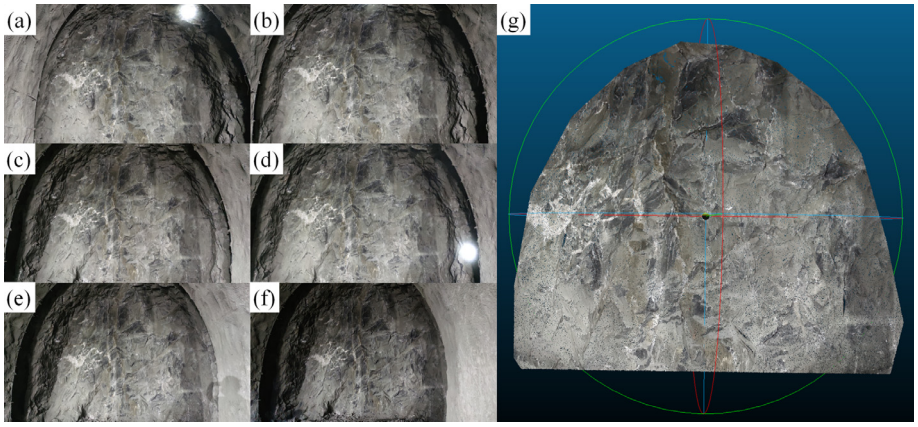

Fig. 2. Image acquisition and 3D reconstruction of case 1. (a)–(f) Image sequence of the excavation face, and (g) 3D reconstructed points in ROI.

smooth surfaces (Daghigh et al., 2022), the classical PCA cannot perform well on the sharp points of geometric singularities (Nurunnabi et al., 2015; Khaloo & Lattanzi, 2017). This paper proposes an adaptive vector estimation based on neighbor assignments to accurately generate both normal vectors and directional vectors of sharp points.

2.1.1 Point cloud vector calculation based on PCA analysis

Given the point cloud $P = \{p_1, p_2, \cdots, p_N\}$ (N denotes the point number), the vector of a point $p_c \in P$ requires calculating the covariance matrix M_{p_c} as

$$M_{p_c} = \frac{1}{k_{\mathrm{nn}}} \sum_{i=1}^{k_{\mathrm{nn}}} (p_i - p_c)(p_i - p_c)^{\mathrm{T}} = \sum_{i=1}^{3} \lambda_i \boldsymbol{e}_i \boldsymbol{e}_i^{\mathrm{T}}, \tag{1}$$

where k_{nn} is the nearest point number of p_c, and p_i is the ith point of k_{nn} nearest points of p_c with Euclidean distance. $\lambda_1 \geq \lambda_2 \geq \lambda_3$ are the eigenvalues with the corresponding eigenvectors of $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{e}_3$, respectively. As shown in Fig. 3, $\boldsymbol{e}_1$ denotes the longest extension direction of the neighbor points to serve as the directional vector of p_c, and $\boldsymbol{e}_3$ denotes the shortest extension direction to serve as the normal vector of p_c. Considering that small k_{nn} (e.g., $k_{\mathrm{nn}} = 15$) can cause significant noise in normal vector calculation and large k_{nn} (e.g., $k_{\mathrm{nn}} > 30$) can significantly smooth local curvature (Riquelme et al., 2014), $k_{\mathrm{nn}} = 15$ is set as the default value in the analysis of the following case studies.

2.1.2 Sharp point detection based on neighbor angle variation

In this paper, sharp points denote the edge points and corner points located in large geometry curvatures (Wang et al., 2013). The variation of the neighbor angle is used for sharp point detection.

Firstly, the distance function is defined as the acute angle of normal vectors (Jimenez-Rodriguez & Sitar, 2006). Given $P = \{p_1, \cdots, p_N\}$ the point cloud, $N_{\mathrm{v}} = \{\boldsymbol{n}_1, \cdots, \boldsymbol{n}_N\}$ the normal vector set, the distance function is defined as

$$\mathrm{dist}(\boldsymbol{n}_i, \boldsymbol{n}_j) = \arccos\left(\frac{|\boldsymbol{n}_i \cdot \boldsymbol{n}_j|}{|\boldsymbol{n}_i| \cdot |\boldsymbol{n}_j|}\right), \tag{2}$$

where $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$ denote any two different normal vectors, and $|\cdot|$ denotes the norm of vectors.

Then, the variation of the neighbor angle δ_i of point p_i in P is defined as

$$\delta_i = \frac{1}{k_{\mathrm{nn}}} \sum_{j=1}^{k_{\mathrm{nn}}} \mathrm{dist}(\boldsymbol{n}_i, \boldsymbol{n}_j), \tag{3}$$

where $\boldsymbol{n_j}$ denotes the normal vector of the jth k_{nn} nearest point of p_i.

Finally, sharp points are selected as the points with neighbor angle variations larger than the mean δ of all points in P. Accordingly, the sharp point set S_{p} is defined as

$$S_{\mathrm{p}} = \left\{ i \middle| \delta_i > \frac{1}{N} \sum_{j=1}^{N} \delta_j,\ i \in \{1, \cdots, N\} \right\}. \tag{4}$$

2.1.3 Adaptive vector calculation of a sharp point based on iterative neighbor assignment

Figure 4(a) shows the sharp point detection result of case 1. As shown in Fig. 4(b), normal vectors (the red lines) of sharp points are often the transition vectors of adjacent discontinuities (I and II), which can interfere with the normal-vector-based orientation analysis. Therefore, the adaptive normal vector method is proposed as follows.

(1) Given a point cloud set $P = \{p_1, \cdots, p_N\}$, normal vector set $N_{\mathrm{v}} = \{\boldsymbol{n}_1, \cdots, \boldsymbol{n}_N\}$, sharp point index set in P set $S_{\mathrm{p}} = \{s_1, \cdots, s_{N_{\mathrm{s}}}\}$ (N_{s} denotes the sharp point number), the non-sharp point number set $Q_{\mathrm{n}} = \{q_1, \cdots, q_{N_{\mathrm{s}}}\}$ in the k_{nn} neighbors of each sharp

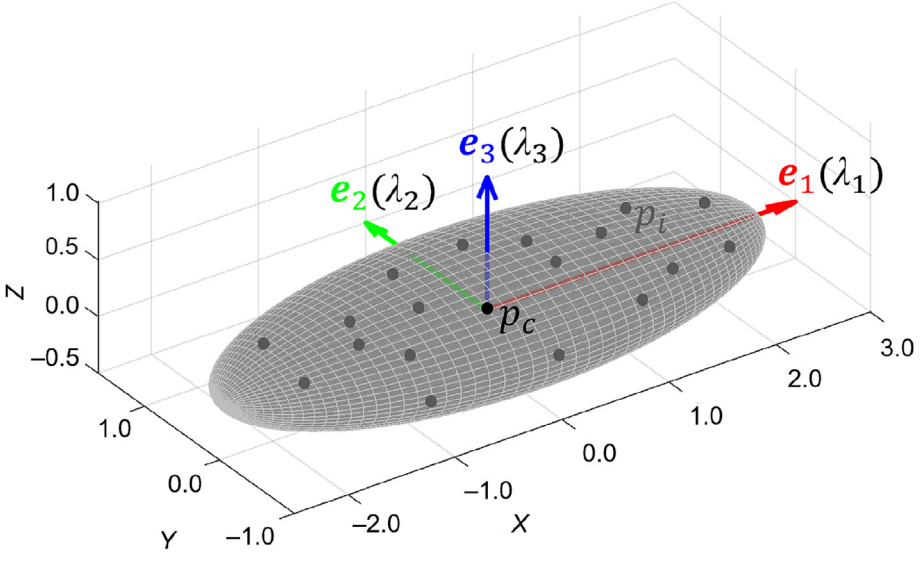

Fig. 3. Illustration of PCA.

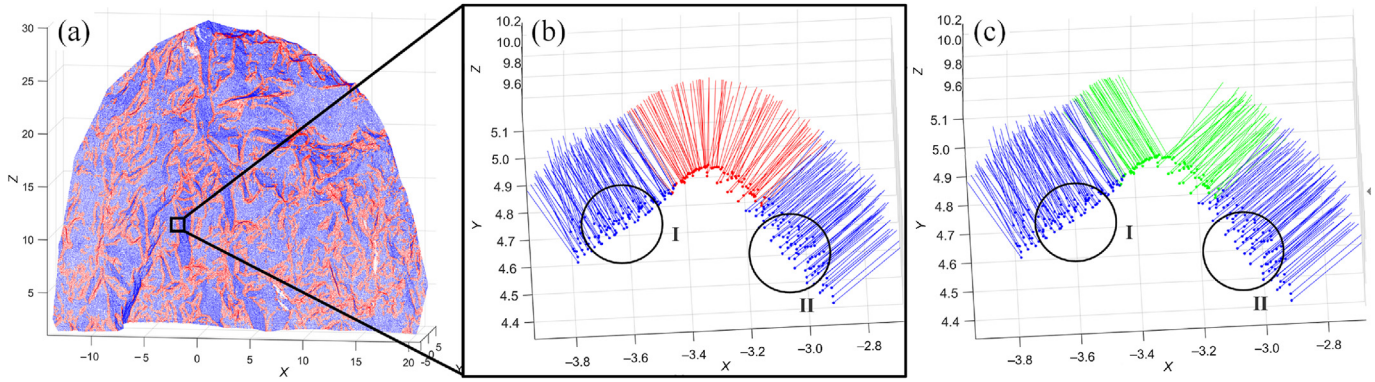

Fig. 4. Adaptive normal vectors of case 1. (a) Sharp points. Red points denote sharp points and blue points denote non-sharp points, (b) initial normal vectors (red lines) of sharp points, and (c) adaptive normal vectors (green lines) of sharp points.

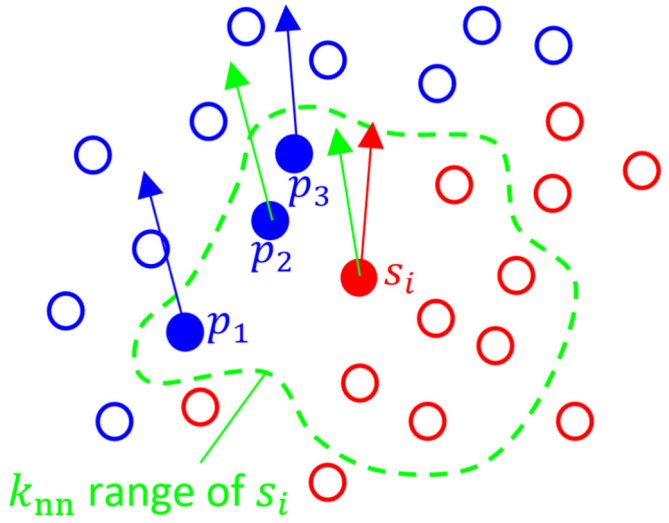

Fig. 5. Normal vector neighbor assignment of sharp points (Note: The red normal vector of sharp point s_i is assigned with the green normal vector from the nearest non-sharp point p_2 of s_i's k_{nn} nearest neighbor points.

point. As shown in Fig. 5, red points denote sharp points and blue points denote non-sharp points. The non-sharp k_{nn} neighbor points of the sharp point s_i include p_1, p_2 and p_3. Therefore, $q_i = 3$.

(2) In order to control the neighbor assignment that starts from the sharp points near the non-sharp points, an effective neighbor number threshold T is set as $\lfloor k_{nn}/3 \rfloor$, where $\lfloor x \rfloor$ denotes the maximum integer smaller than x.

(3) Search for points in S_p whose non-sharp k_{nn} neighbor point numbers are equal to T, and let the set as $\widetilde{S}_p = \{\widetilde{s}_1, \cdots, \widetilde{s}_{N_t}\}$ (N_t denotes the element number in $\widetilde{S}_p$).

(4) If $\widetilde{S}_p \neq \varnothing$, then the vector of each sharp point in $\widetilde{S}_p$ is assigned with the vector of its nearest non-sharp neighbor point. Given $\widetilde{s}_i \in \widetilde{S}_p$, the non-sharp neighbor point set of $\widetilde{s}_i$ is $\{p_1, \cdots, p_{N_m}\}$(N_m denotes the element number in the set), the Euclidean distance to $\widetilde{s}_i$ is $D = \{d_1, \cdots, d_{N_m}\}$, then the assigned normal vector $\boldsymbol{n}_i$ of $\widetilde{s}_i$ is defined as

$$\boldsymbol{n}_i = \{\boldsymbol{n}_j | d_j = \min(D)\}. \quad (5)$$

As shown in Fig. 5, the red normal vector of s_i is assigned to the green normal vector, which is the nearest non-sharp point p_2 of the k_{nn} nearest points of s_i.

Then set $S_p = S_p - \widetilde{S}_p$. If $S_p \neq \varnothing$, repeat step (2); if $S_p = \varnothing$, then perform step (6).

(5) If $\widetilde{S}_p = \varnothing$, then set $T = T - 1$ and repeat step (3).

(6) The iterative assignment stops, and all sharp points have been assigned with adaptive normal vectors.

The adaptive normal vectors of all sharp points can be calculated by steps (1)–(6). Figure 6 shows the process of adaptive iterative assignment of normal vectors. The blue points denote non-sharp points, the red points denote sharp points that have not been assigned with adaptive vectors, and the green points denote sharp points assigned with adaptive vectors. All sharp points in case 1 can be assigned after six iterations. As shown in Fig. 4(c), the adaptive normal vectors can better reflect the direction of adjacent discontinuity planes (I and II).

The directional vectors of sharp points can also cause large errors at the junction of different trace segments and near the median axis of trace segments. As shown in Fig. 7(b), the red directional vectors are more scattered and do not maintain an approximate direction to the extension trend of the blue directional vectors. Therefore, the red directional vectors also need to be assigned adaptively.

The iterative process of adaptive directional vectors is different from that of adaptive normal vectors. Firstly, the normal sharp point for the calculation of adaptive normal vectors is selected out of the whole point cloud, while the directional sharp points for the calculation of adaptive directional vectors are selected out of the normal sharp points. Therefore, the k_{nn} search used in Eqs. (3) and (4)

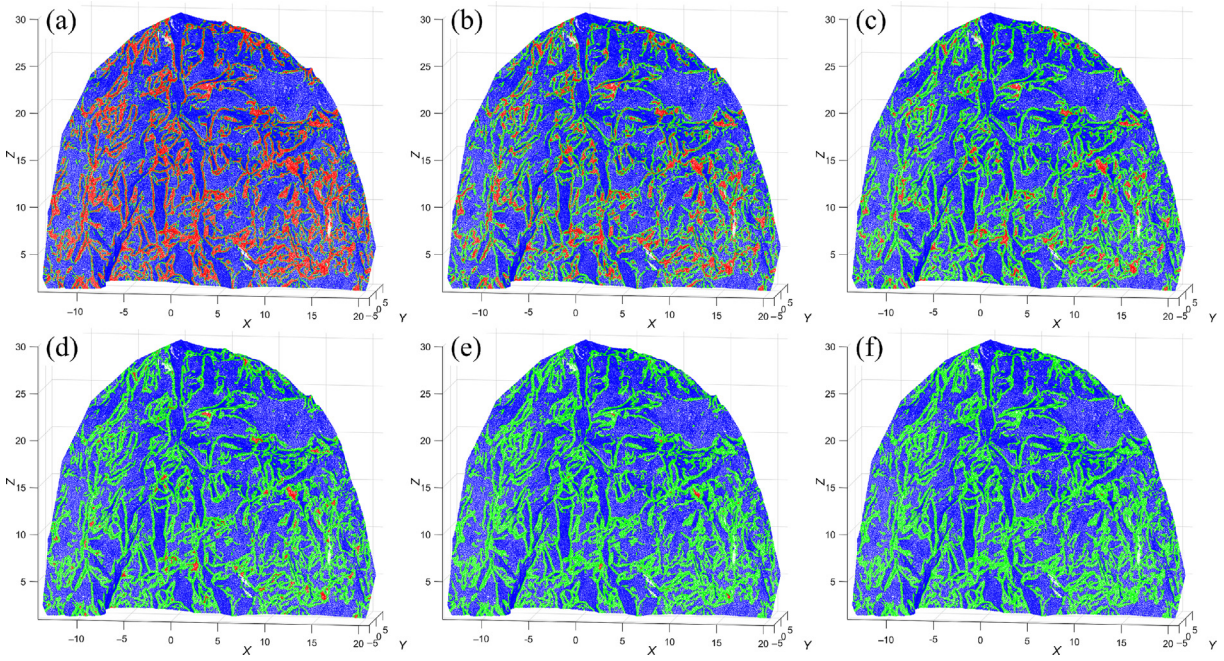

Fig. 6. Adaptive normal vector iteration of case 1. (a) 1st iteration, (b) 2nd iteration, (c) 3rd iteration, (d) 4th iteration, (e) 5th iteration, and (f) 6th iteration (Note: Blue points denote non-sharp points, red points denote sharp points not assigned with adaptive vectors, and green points denote sharp points assigned with adaptive vectors).

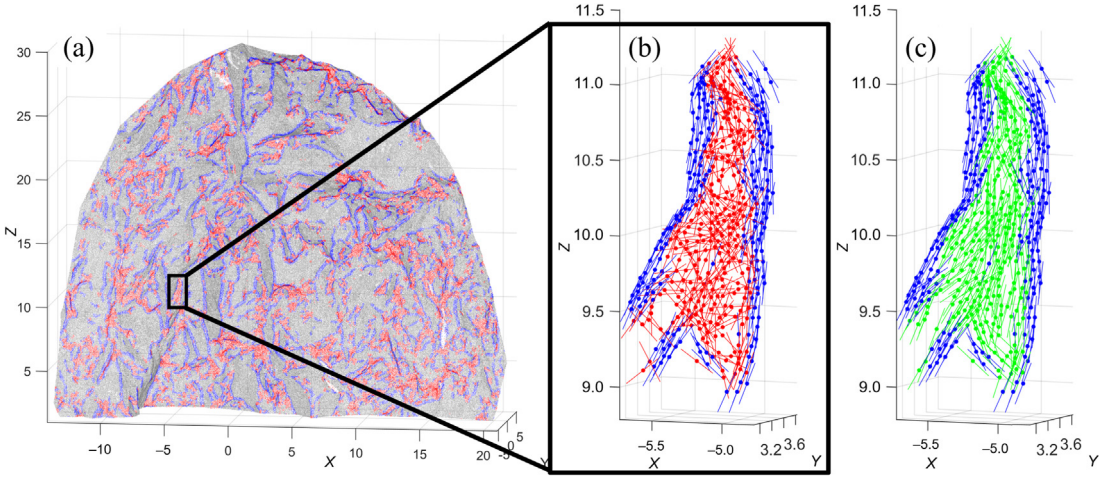

Fig. 7. Adaptive directional vectors of case 1. (a) Directional sharp points. Blue points denote directional non-sharp points and red points denote directional sharp points, (b) directional vectors (red lines) before adaptive assignment, and (c) directional vectors (green lines) after adaptive assignment.

needs to be performed using normal sharp points. Secondly, the eigenvector $\boldsymbol{e}_1$ corresponding to the largest eigenvalue λ_1 calculated by Eq. (1) is used to represent directional vectors. After the above modifications, the calculation of adaptive directional vectors can be carried out by performing the same process as steps (1)–(6).

Figure 8 shows the process of adaptive iterative assignment of directional vectors. The blue points denote directional non-sharp points, the red points denote directional sharp points not assigned with adaptive vectors, and the green points denote points assigned with adaptive directional vectors. All directional sharp points in case 1 can be assigned after 4 iterations. As shown in Fig. 7(c), the adaptive directional vectors can effectively reduce the disordered directions (Fig. 7(b)) and better reflect the extension trend of the trace.

2.2 PCA-weighted iterative oriented contraction (PWI-OC)

After adaptive vector calculation, the trace skeleton points are extracted for trace generation. The Laplace thinning (Guo et al., 2018) and Laplace contraction (Zhang et al., 2020) methods have been used to identify traces through point skeletons. However, these methods do not

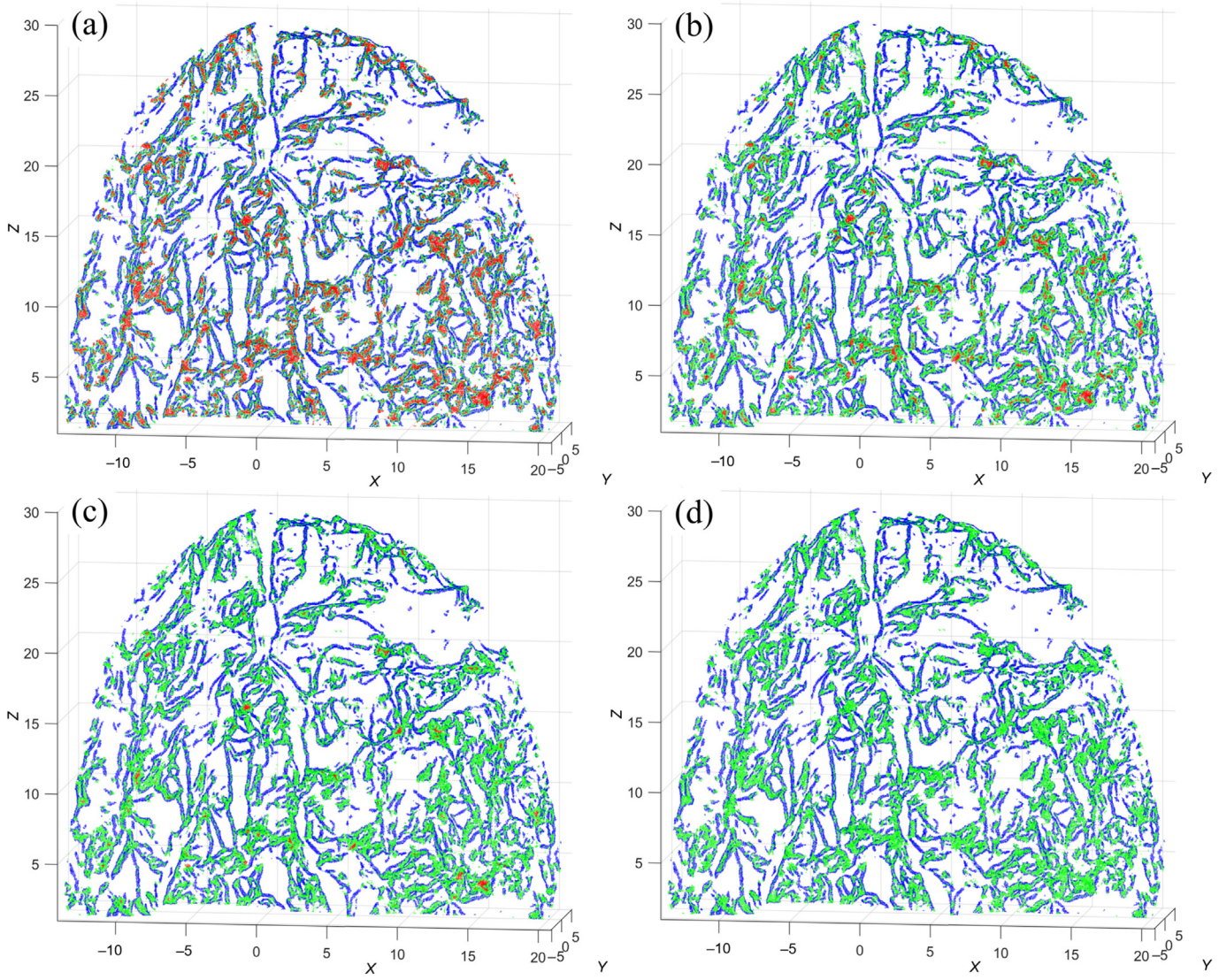

Fig. 8. Adaptive directional vector iteration of case 1. (a) 1st iteration, (b) 2nd iteration, (c) 3rd iteration, and (d) 4th iteration (Note: Blue points denote directional non-sharp points, red points denote directional sharp points that have not been assigned by adaptive directional vectors, and green points denote directional sharp points that have been assigned by adaptive directional vectors).

restrict the direction of thinning or contraction, resulting in broken point skeletons, which affect the accuracy of trace generation. In addition, these methods require manual fine-tuning of the threshold to control curvature point generation for thinning and contraction. Therefore, a PWI-OC method is proposed to improve these restrictions. The contraction direction is controlled to generate coherent point cloud skeletons, and the boundary points of automatic optimal orientation grouping are extracted as curvature points without manually fine-tuning the curvature threshold.

2.2.1 Orientation grouping by improved K-means with fast ρ-δ decision graph

After the adaptive normal vectors are generated in Section 2.1, a modified *K*-means based on a ρ-δ decision graph for fast optimization of initial clustering centers is used for orientation grouping.

Firstly, in order to search for all possible initial clustering centers in the 3D normal vector space, the Fibonacci series is adopted to generate raster vectors which are evenly distributed on the upper unit hemisphere. The specific procedures are as follows (González, 2010).

Given N_r the number of raster vectors, $\widetilde{N}_r = 2N_r + 1$, $n \in \{-N_r, -N_r + 1, \cdots, N_r - 1, N_r\}$, the longitude and latitude coordinates $\left(l_i, \widetilde{l}_i\right)$ of each raster vector are defined as

$$\begin{cases} l_n = \mod(n, \phi) \times 360^\circ/\phi \\ \widetilde{l}_n = \arcsin\left(\frac{2n}{N_r}\right) \times 180^\circ/\pi \end{cases}, \tag{6}$$

where $\phi = 1.618$, and l_n is modified as

$$\begin{cases} l_n = 360^\circ + l_n, & \text{if } l_n < -180^\circ \\ l_n = l_n - 360^\circ, & \text{if } l_n \geq 180^\circ \end{cases}. \tag{7}$$

Then $\left(l_i, \widetilde{l}_i\right)$ is transformed into Cartesian coordinates as

$$\begin{cases} x_i = \cos(\widetilde{l}_n)\cos(l_n) \\ y_i = \cos(\widetilde{l}_n)\sin(l_n) \\ z_i = \sin(\widetilde{l}_n) \end{cases}. \tag{8}$$

Next, calculate each n by Eq. (6)–(8) and $\widetilde{N}_\mathrm{r}$ normal vectors can be obtained. Finally, the raster vectors can be obtained as the N_r normal vectors in $\widetilde{N}_\mathrm{r}$ on the upper unit hemisphere.

Figure 9 shows the stereographic projection of raster vectors corresponding to different N_r. Although a large number of raster vectors can accurately reflect different orientations in the 3D normal vector space, they also increase computational overhead. Considering that the International Society of Rock Mechanics (ISRM) uses 5° as the recommended error for manual orientation measurement, the raster vectors corresponding to $N_\mathrm{r} = 6000$ are used as default to balance the efficiency and accuracy. At this time, the average adjacent angle is 2.49°, which is less than half of the ISRM recommended error.

Next, a fast ρ-δ decision graph of clustering by fast search and find of density Peaks (CFSFDP, Rodriguez & Laio, 2014) based on raster vectors is used to calculate the likelihood of all raster vectors to be used as initial clustering centers. In order to calculate the density ρ_i of each raster vector, the normal vectors in the neighbors within a threshold angle T_ρ of each raster vector need to be calculated. Given the normal vector set of all points $N_\mathrm{v} = \{\boldsymbol{n}_1, \cdots, \boldsymbol{n}_N\}$ and the raster vector set $R_\mathrm{v} = \{\boldsymbol{r}_1, \cdots, \boldsymbol{r}_{N_\mathrm{r}}\}$. Then the neighbor normal vector set $\widetilde{N}_{\mathrm{v},i}$ within T_ρ of $\boldsymbol{r}_i$ is defined as

$$\widetilde{N}_{\mathrm{v},i} = \left\{\boldsymbol{n}_j | \mathrm{dist}(\boldsymbol{n}_j, \boldsymbol{r}_i) < T_\rho, \boldsymbol{n}_j \in N_\mathrm{v}\right\}. \tag{9}$$

ρ_i is the element number in $\widetilde{N}_{\mathrm{v},i}$. The value of T_ρ can directly influence the density calculation. Considering that ISRM's recommended orientation error of manual measurement is 5°, T_ρ is set as 5° by default.

δ_i is the minimum distance of $\boldsymbol{r}_i$ to the raster vector with a higher density in R_v, which is

$$\delta_i = \min_{\boldsymbol{r}_j} \mathrm{dist}(\boldsymbol{r}_i, \boldsymbol{r}_j), \tag{10}$$

where $\boldsymbol{r}_j \in R_\mathrm{v}$ and $\rho_j > \rho_i$.

In order to avoid the initial clustering centers being too close, δ_i is modified as 0 if it is smaller than 5°.

Next, calculate the likelihood score of raster vectors to be initial clustering centers based on ρ_i and δ_i. Given $\rho = \{\rho_1, \cdots, \rho_{N_\mathrm{r}}\}$, $\delta = \{\delta_1, \cdots, \delta_{N_\mathrm{r}}\}$ and the score $S_\mathrm{c} = \{c_1, \cdots, c_{N_\mathrm{r}}\}$, then $c_i \in S_\mathrm{c}$ is defined as

$$c_i = \frac{\rho_i \cdot \delta_i}{\max(\rho) \cdot \max(\delta)}. \tag{11}$$

Given the cluster number k, the raster vectors corresponding to the top k largest values in S_c are selected as the initial clustering centers.

Figure 10 shows that all the initial clustering centers are located near the center region with dense distribution of normal vectors, which shows the effectiveness of the fast ρ-δ decision graph for the selection of initial clustering centers.

After obtaining the initial clustering centers, K-means can be performed to generate orientation results of both normal vectors and raster vectors.

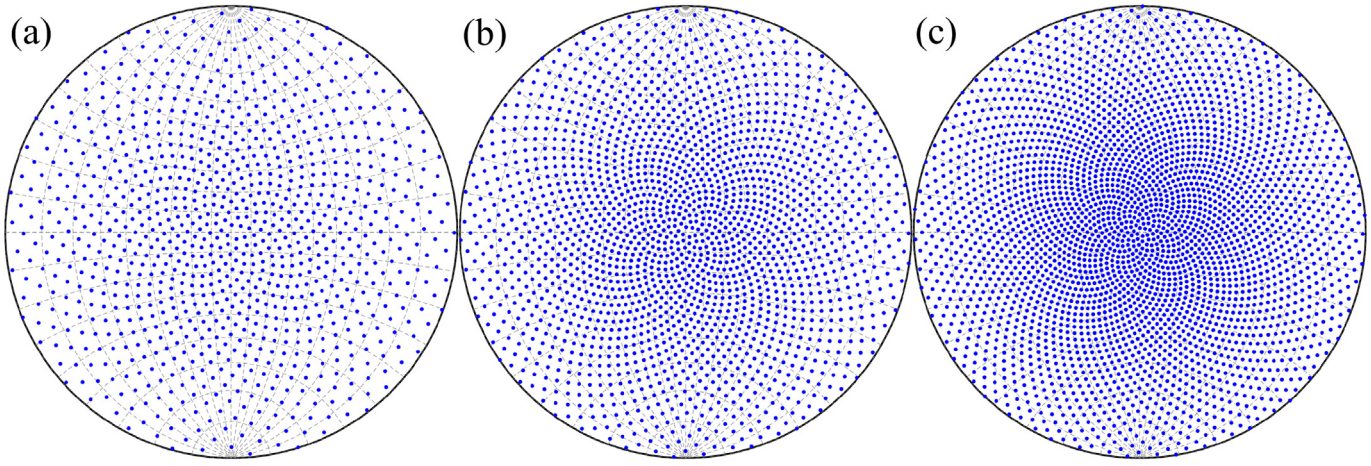

Fig. 9. Stereographic projection of raster vectors with different N_r. (a) $N_\mathrm{r} = 2000$ with the average adjacent angle of 4.3°, (b) $N_\mathrm{r} = 4000$ with the average adjacent angle of 3.09°, and (c) $N_\mathrm{r} = 6000$ with the average adjacent angle of 2.49°.

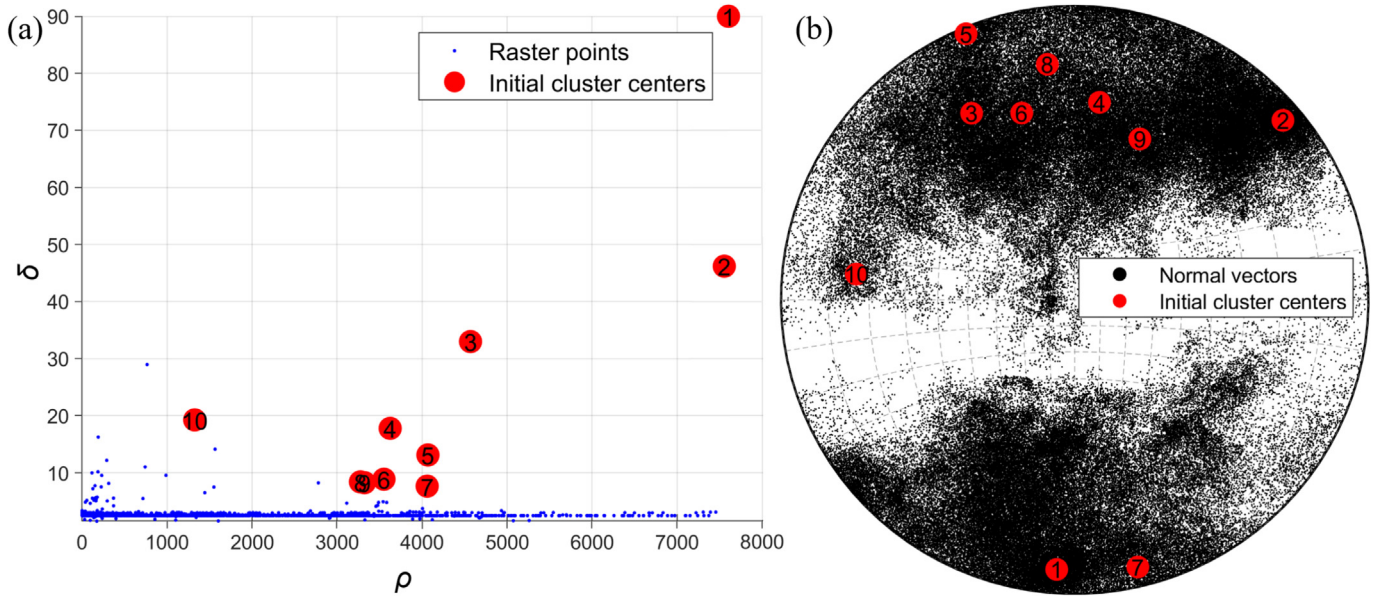

Fig. 10. Ten initial clustering centers generated by the fast ρ-δ decision graph. (a) ρ-δ decision graph, and (b) stereographic projection of initial clustering centers.

2.2.2 Fast silhouette calculation based on raster vector for clustering evaluation

Since K-means cannot determine the optimal cluster number, the cluster number k is usually set from 2 to 8 (Singh et al., 2022). Then the optimal grouping result is selected by validating the grouping result of each k using clustering validity indexes. The commonly used clustering validity indexes include the silhouette coefficient index (Rousseeuw, 1987), Calinski-Harabasz index (Caliński & Harabasz, 1974), Davies Bouldin index (Davies & Bouldin, 1979), and Xie Beni index (Xie & Beni, 1991), etc. The silhouette index is used in this section because it can reflect both the dissertation and the aggregation of clusters. However, the original silhouette requires calculating the distance of any two normal vectors of the point cloud, which has the time complexity of $O(N^2)$. This section converts the calculation of point cloud normal vectors to the approximate calculation of raster vectors with the density of ρ, which reduces the time complexity to $O(1)$.

Specifically, given the cluster number k, the raster vector set $R_v = \{\boldsymbol{r}_1, \cdots, \boldsymbol{r}_{N_r}\}$, the grouping result of raster vectors $G_r = \{g_1, \cdots, g_k\}$, the group id of raster vectors $I_d = \{i_{d,1}, \cdots, i_{d,N_r}\}$, and the density of raster vectors $\rho = \{\rho_1, \cdots, \rho_{N_r}\}$, the silhouette value $s_{il}(i)$ of the ith raster vector is defined as

$$s_{il}(i) = \frac{b(i) - a(i)}{\max\{b(i), a(i)\}}, \tag{12}$$

where $a(i)$ is defined as

$$a(i) = \frac{\sum_{j \in g_{i_{d,J}},\, j \neq i} \mathrm{dist}(\boldsymbol{r}_i, \boldsymbol{r}_j) \cdot \rho_j}{\sum_{j \in g_{i_{d,J}},\, j \neq i} \rho_j}. \tag{13}$$

In Eq. (12), $b(i)$ is defined as

$$b(i) = \min_{\substack{1 \le h \le k \\ h \neq i_{d,i}}} \left\{ \frac{\sum_{j \in g_h} \mathrm{dist}(\boldsymbol{r}_i, \boldsymbol{r}_j) \cdot \rho_j}{\sum_{j=1}^{N_r} \rho_j} \right\}. \tag{14}$$

Finally, the silhouette value of the grouping result with k is defined as

$$S_{il}(k) = \frac{\sum_{i=1}^{N_r} s_{il}(i) \cdot \rho_i}{\sum_{i=1}^{N_r} \rho_i}. \tag{15}$$

Silhouette values range from –1 to 1, and a large value indicates a good clustering quality. The silhouette value $S_{il}(k)$ is calculated with k from 2 to 8, then the optimal k_{opt} is selected corresponding to the largest silhouette value.

As shown in Fig. 11(a), the adaptive normal vectors have larger silhouette values corresponding to each cluster number k than the non-adaptive normal vectors, indicating that the adaptive normal vectors are effective in improving grouping quality. In addition, the optimal grouping number of the adaptive normal vectors is the same as the manual grouping number of 3 (Fig. 11(a)), which effectively generates integrated discontinuities (such as region I). However, the optimal grouping number of non-adaptive normal vectors is 6 (Fig. 11(a)), which not only generates irregular discontinuities in region I, but also generates trivial discontinuities (e.g., regions II and III) that should not be recognized as a set. Both results show the significance of adaptive normal vectors for orientation recognition.

2.2.3 Boundary point detection based on optimal orientation grouping result

Traditional methods often perform trace identification directly based on the curvature sharp points (Umili et al., 2013; Li et al., 2016; Guo et al., 2018), which require man-

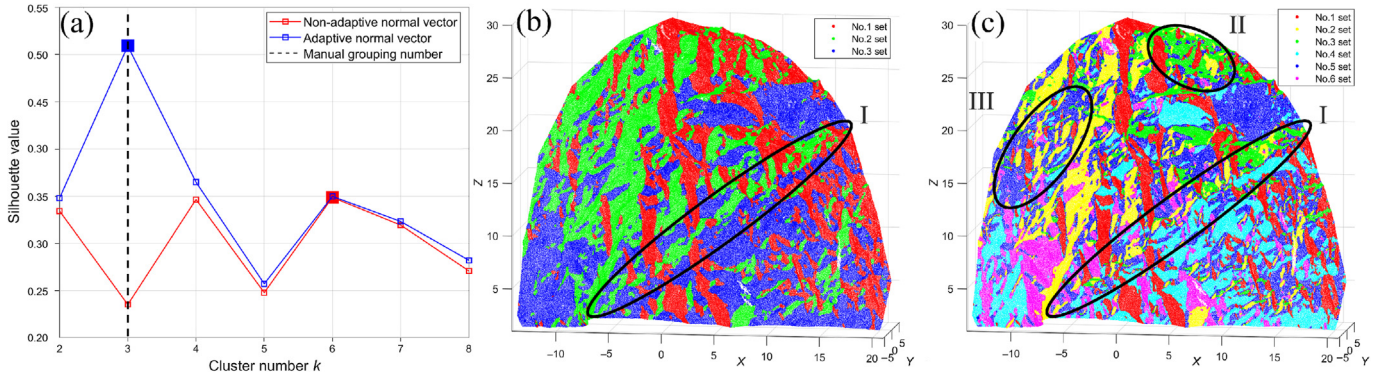

Fig. 11. Grouping results by different types of normal vectors. (a) Comparison of silhouette values of adaptive normal vectors and non-adaptive normal vectors, (b) optimal grouping result of adaptive normal vectors with $k_{\text{opt}} = 3$, and (c) optimal grouping result of non-adaptive normal vectors with $k_{\text{opt}} = 6$.

ual fine-tuning of curvature thresholds and generate traces inconsistent with grouping results. This section extracts curvature points as boundary points of the optimal grouping results without manual fine-tuning.

Given the point cloud set $P = \{p_1, \cdots, p_N\}$, the optimal grouping results $G_{\text{r}} = \{g_1, \cdots, g_k\}$, the group id of each point $I_{\text{d}} = \{i_{\text{d},1}, \cdots, i_{\text{d},N_{\text{r}}}\}$, and the boundary points of grouping results $B_{\text{p}} = \{b_1, \cdots, b_{N_{\text{b}}}\}$ (N_{b} denotes the element number in B_{p}). For each $b_i \in B_{\text{p}}$, set the k_{nn} nearest point index set of b_i in B_{p} as $K_{\text{b}} = \{k_{\text{b},1}, \cdots, k_{\text{b},k_{\text{nn}}}\}$, then $k_{\text{b},i}$ should satisfy

$$C_{\text{t}}\{U_{\text{nq}}(i_{\text{d},k_{\text{b},i}})\} > 1, \tag{16}$$

where $C_{\text{t}}(x)$ denotes the element number in set x ,and $U_{\text{nq}}(x)$ denotes the set of all elements not repeated in set x.

Figure 12 shows the comparison of sharp points generated by different methods. It indicates that the traditional curvature points (Fig. 12(a)) are less integrated than boundary points (Fig. 12(b)) generated from the orientation grouping of adaptive normal vectors, such as the regions I–V. In addition, the boundary points generated by non-adaptive normal vectors are more trivial than adaptive normal vectors, which is caused by the incorrect grouping number identification of 6 instead of 3 (Fig. 11(a)). Therefore, the boundary point generated from the orientation grouping of adaptive normal vectors is more detailed and robust than the curvature points of Li et al. (2016).

2.2.4 Skeleton point generation based on PWI-OC of boundary point

After obtaining the boundary points in Section 2.2.3, an iterative oriented contraction based on PCA weighting is proposed to extract the point cloud skeleton of traces. Considering the eigenvector $\boldsymbol{e}_1$ corresponding to the largest eigenvalue λ_1 denotes the longest extension direction of the neighbor point set (Fig. 3), the proportion of λ_1 in the eigenvalues is used as the weight for boundary point contraction. The specific steps are as follows.

(1) Given the boundary point set $B_{\text{p}} = \{b_1, \cdots, b_{N_{\text{b}}}\}$ calculate the directional vectors of each point in B_{p} according to Eq. (1) as $D_{\text{v}} = \{\boldsymbol{d}_{\text{v},1}, \cdots, \boldsymbol{d}_{\text{v},N_{\text{b}}}\}$,
(2) For each $b_i \in B_{\text{p}}$, given the eigenvalues are $\lambda_{i,1} \geq \lambda_{i,2} \geq \lambda_{i,3}$, then the contraction weight w_i is defined as

$$w_i = \left(\frac{\lambda_{i,1}}{\sum_{j=1}^{3}\lambda_{i,j}}\right)^2. \tag{17}$$

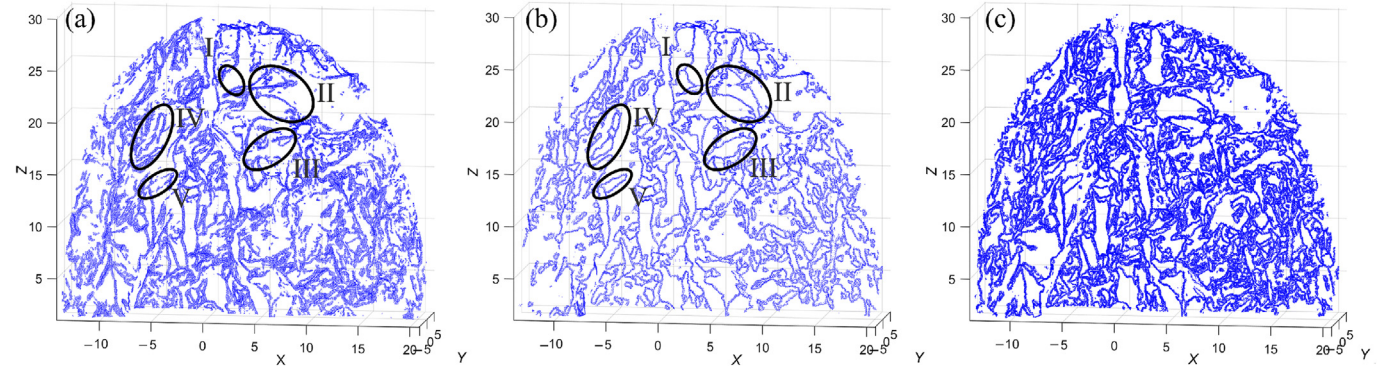

Fig. 12. Comparison of feature points generated by different methods. (a) Curvature points generated by Li et al. (2016), (b) boundary points of the optimal grouping result ($k_{\text{opt}} = 3$) generated by non-adaptive normal vectors, and (c) boundary points of the optimal grouping result ($k_{\text{opt}} = 6$) generated by adaptive normal vectors.

Let the k_{nn} nearest point index of b_i in B_p as $K_b = \{k_{b,1}, \cdots, k_{b,k_{nn}}\}$, the contraction point b_i' of b_i is defined as

$$b_i' = \frac{\sum_{j \in k_{b,i}} w_j \cdot b_j}{\sum_{j \in k_{b,i}} w_j}. \tag{18}$$

(3) In order to make the contracted points evenly distributed along the trace extension direction, the displacement direction of b_i' needs to be corrected. Specifically, in order to move b_i perpendicular rather than parallel to the trace center axis, the parallel displacement relative to the trace center axis needs to be corrected. As shown in Fig. 13, given $\boldsymbol{d}_{v,i}$ the directional vector of b_i, the projection of the displacement $\boldsymbol{b}_i = b_i' - b_i$ on $\boldsymbol{d}_{v,i}$ is defined as

$$\boldsymbol{\Delta} = \boldsymbol{b}_i \frac{|\boldsymbol{b}_i \cdot \boldsymbol{d}_{v,i}|}{|\boldsymbol{b}_i||\boldsymbol{d}_{v,i}|}. \tag{19}$$

Therefore, the final contraction position b_i'' of b_i is defined as

$$b_i'' = b_i' - \boldsymbol{\Delta}. \tag{20}$$

(4) One-time contraction usually cannot make the boundary points sufficiently contracted, which can generate a less even distribution of skeleton points with an unclear skeleton morphology (Fig. 14(c)). Therefore, the iterative contraction is required, which is achieved by performing steps (1)–(3) multiple times. In order to determine the contraction number, the proportion of the largest eigenvalue is used to define the contraction threshold T_c as

$$T_c = \frac{\sum_{i=1}^{N_b} \lambda_{i,1}}{\sum_{i=1}^{N_b} \sum_{j=1}^{3} \lambda_{i,j}}. \tag{21}$$

Figure 14 shows the effect of multiple iterations of the oriented contraction. It can be seen that as the iteration number increases (Fig. 14(c)–(f)), the skeleton points extracted after contraction are more uniformly distributed along trace axis directions, and the point cloud skeleton morphology is more obvious. It also shows that the change of contraction effects is not obvious when $T_c > 0.85$ (Fig. 14(f)). Therefore, the default contraction threshold T_c is set as 0.85 by default. Figure 14(c)–(e) shows that 3 times of contraction can meet the T_c requirements for case 1. Figure 14(b) shows the skeleton points after 3 times of contraction.

2.3 Trace segment generation based on sparse growing

After obtaining the point skeletons, they are next connected to generate trace segments. Li et al. (2016) proposed a growth method for trace segment generation. Zhang et al. (2020) connected the skeleton points to generate trace segments based on radial, axial, and angular thresholds. However, both methods require triangulation and cannot directly process the raw point cloud. Moreover, the orientation grouping information is not considered in the trace generation process, resulting in a mismatch between the generated traces and the orientation grouping result. In this section, we propose a sparse growing method that directly analyzes the raw point cloud and combines the orientation grouping result for trace segment generation.

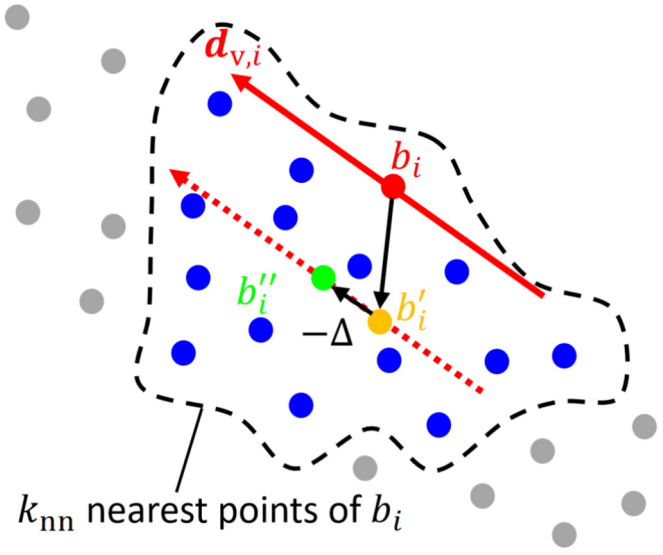

Fig. 13. Oriented contraction process of boundary point bp_i.

2.3.1 Initial generation of trace segment based on the neighbor connection of the skeleton point

Given the point cloud set P, the optimal orientation grouping result $G_r = \{g_1, \cdots, g_k\}$, the boundary point index set $I_b = \{i_{b,1}, \cdots, i_{b,N_b}\}$, the contracted skeleton point set $C_p = \{c_1, \cdots, c_{N_b}\}$, the k_{nn} nearest point set $K_c = \{k_{c,1}, \cdots, k_{c,N_b}\}$ of each point in C_p, and the corresponding k_{nn} distance set $D_c = \{d_{c,1}, \cdots, d_{c,N_b}\}$, the initial trace generation is as follows.

(1) Given a trace segment set $S_g = \{\}$, the index set $U_I = \{\}$ of the used points in I_b, then the basic distance metric M_d of the average distance between two adjacent points is defined as

$$M_d = \frac{\sum_{i=1}^{N_b} \min(d_{c,i})}{N_b}. \tag{22}$$

(2) Choose the index of two adjacent orientation sets k_i and k_j for the generation of traces located on the intersection lines.

(3) Find the intersection point index of set k_i and k_j as $I_{ij} = g_{k_i} \cap g_{k_j}$, then find the skeleton point index set I_{ij} for connection as

$$I_{ij} = \{\gamma | \alpha \in I_{ij}, \beta \in k_{c,\alpha}, \gamma \in \beta, d_{c,\beta\gamma} < M_d\}, \tag{23}$$

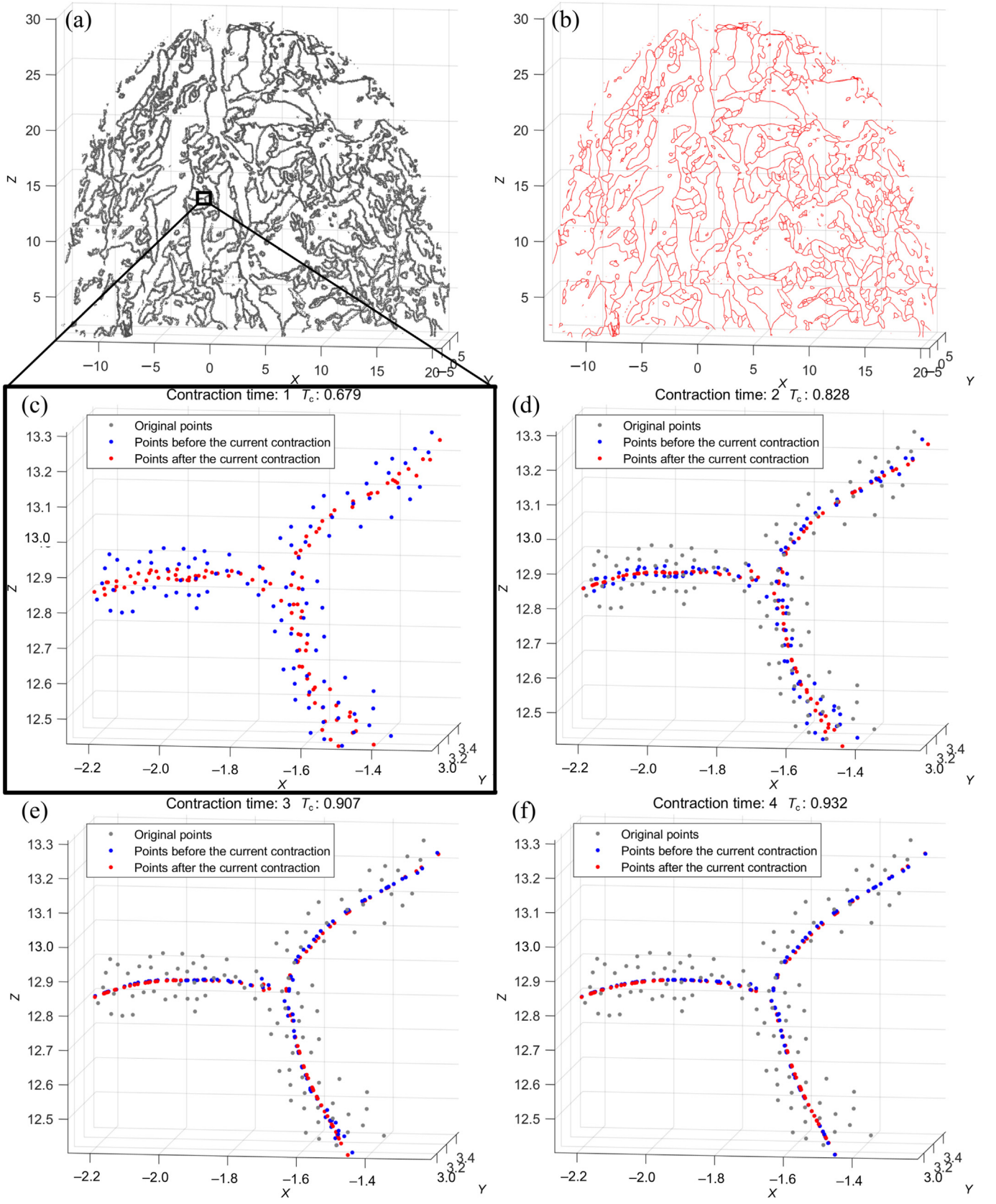

Fig. 14. PCA-weighted iterative oriented contraction (PWI-OC) of boundary points. (a) Boundary points of orientation grouping results, (b) the point skeleton after 4 times of contraction, (c) one-time contraction, (d) two times contraction, (e) three times contraction, and (f) four times contraction.

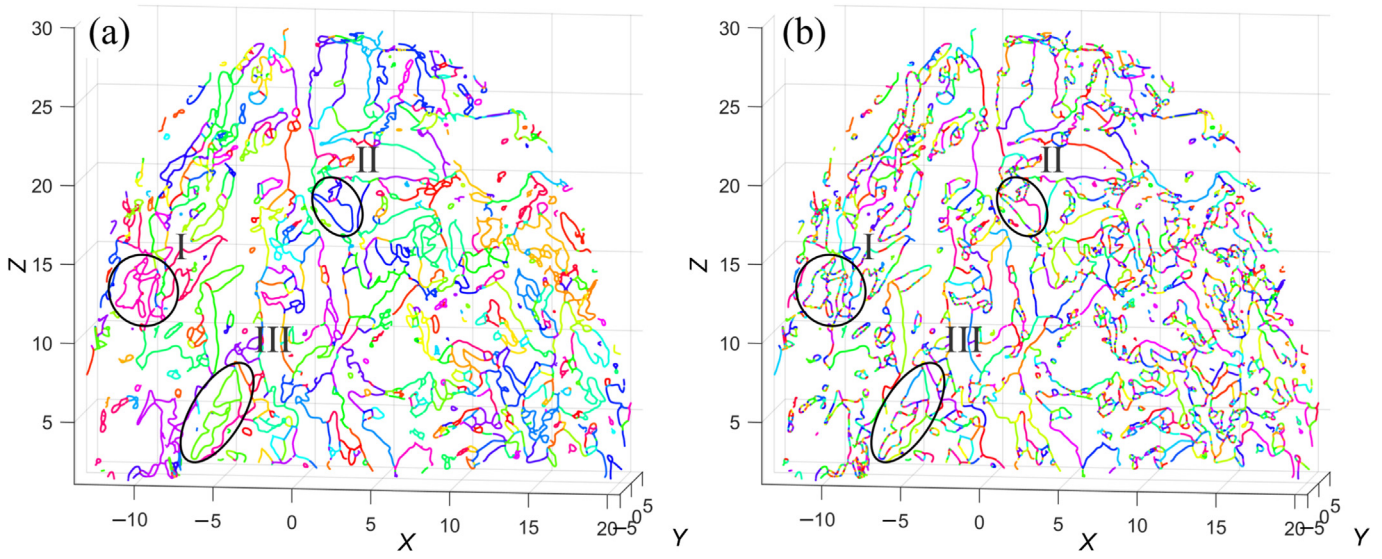

Fig. 15. Initial generation and linearization of trace segments. (a) Initial generation of trace segments, and (b) trace segment linearization.

where $d_{c,\beta\gamma}$ denotes the γth k_{nn} distance of point β. Then let $U_I = U_I \cup I_{ij}$.

(4) Generate trace segments using the neighbor connection of K_c and I_{ij}, generate a new trace segment set $\tilde{S}_g$, and let $S_g = S_g \cup \tilde{S}_g$.
(5) Iterate over any two different grouping numbers by repeating steps (3) and (4). Finally, $S_g = \{s_{g,1}, \cdots, s_{g,N_g}\}$ (N_g denotes the element number in S_g) denotes all the generated trace segments.

Figure 15(a) shows the initial generation results of trace segments. Each color represents an individual trace segment in S_g.

2.3.2 Trace segment linearization based on sparse growing

The initial generation of trace segments without direction control can cause different trace segments to be incorrectly connected together (e.g., regions I to III in Fig. 15 (a)). Li et al. (2016) used a growing method to generate trace segments with direction control. However, this method requires iterative judgements of all skeleton points during the growing process, which is inefficient when dealing with large-scale point clouds. Therefore, this section proposes a sparse growing method to generate trace segments from parts of the skeleton points. As shown in Fig. 16, the specific procedures are as follows.

(1) Given the contracted skeleton point set $C_p = \{c_1, \cdots, c_{N_b}\}$, the mean distance value M_d calculated by Eq. (22), the initial trace segment set $S_g = \{s_{g,1}, \cdots, s_{g,N_g}\}$, and the linearized trace segment set $S_{gl} = \{\}$.
(2) Given a trace segment $s_{g,i} \in S_g$ contains the skeleton point indexes $C_{pt} = \{c_{t,1}, \cdots, c_{t,N_{bt}}\}$, calculate the k_{nn} nearest point set $K_{ct} = \{k_{ct,1}, \cdots, k_{ct,N_{bt}}\}$, the corresponding distance $D_{ct} = \{d_{ct,1}, \cdots, d_{ct,N_{bt}}\}$.
(3) Calculate the directional vector $\boldsymbol{d}_{vt}$ of points in C_{pt} according to Eq. (1).
(4) Divide the C_{pt} uniformly along the $\boldsymbol{d}_{vt}$ direction into 100 steps, generating the C_{pt} point index in each step as $S_{tp} = \{s_{tp,1}, \cdots, s_{tp,100}\}$.
(5) Choose the farthest point in C_{pt} along the $\boldsymbol{d}_{vt}$ direction as the selected seed point $s_{lp,1}$ (e.g., the red point

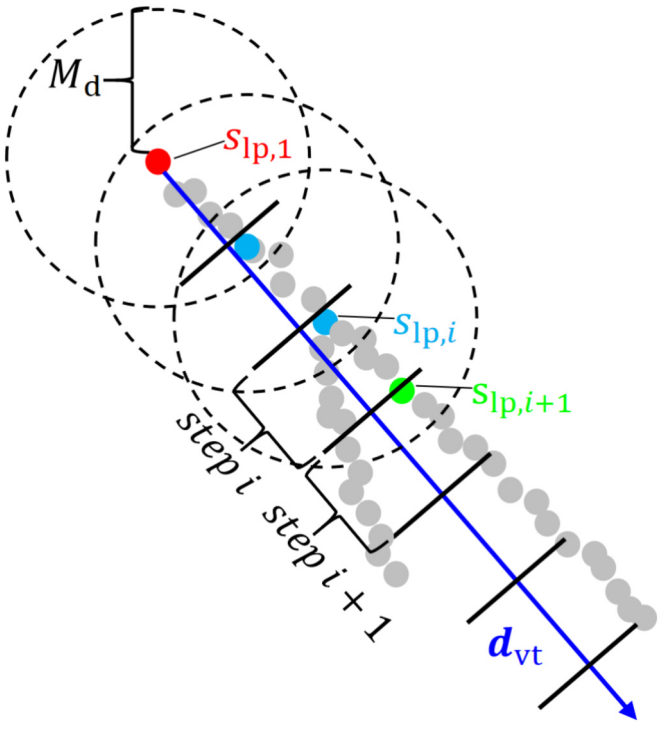

Fig. 16. Trace segment linearization based on sparse growing.

in Fig. 16) in the current step, then let the current trace segment point set as $s_{gl} = \{s_{lp,1}\}$.

(6) Given the point index $s_{lp,i}$ of C_{pt} in the current step i, then find the point $s_{lp,i+1}$ in step $i+1$ by

$$\begin{cases} \left|c_{t,s_{lp,i+1}} - c_{t,s_{lp,i}}\right| = \min\limits_{x\in(k_{ct,s_{lp,i}}\cap s_{tp,i+1})} \left|c_{t,x} - c_{t,s_{lp,i}}\right| \\ \left|c_{t,s_{lp,i+1}} - c_{t,s_{lp,i}}\right| < M_d \end{cases}. \tag{24}$$

(7) If Eq. (24) is satisfied, then let $s_{gl} = s_{gl} \cup s_{lp,i+1}$ and repeat step (6); if not, then let $S_{gl} = S_{gl} \cup k_{ct,s_{gl}}$, $C_{pt} = C_{pt} - S_{gl}$ and repeat step (8).

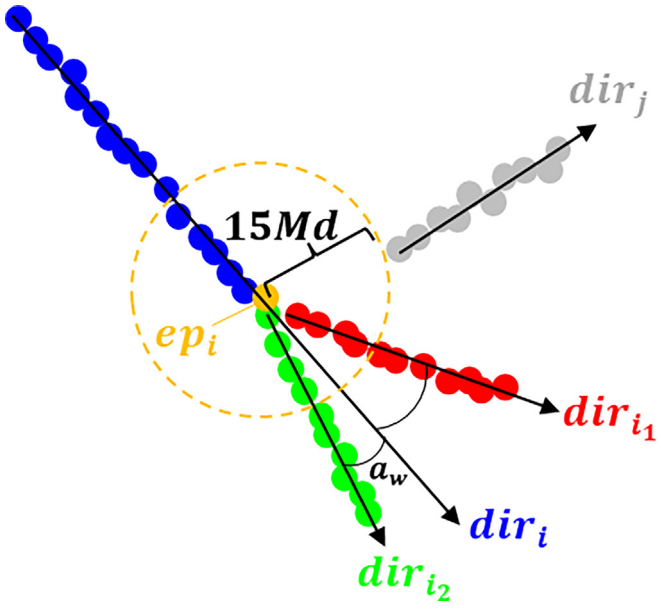

Fig. 17. Trace segment connection based on oriented extension.

(8) If $C_{pt} \neq \varnothing$, then repeat steps (2)–(7); if $C_{pt} = \varnothing$, then the linearization of the trace segment $s_{g,i}$ is stopped.

Perform steps (2)–(8) to linearize all trace segments $s_{g,i} \in S_g$. Figure 15(b) shows the linearization result of Fig. 15(a). It can be seen that the linearization (e.g., regions I–III in Fig. 15(b)) can subdivide the initial trace segments, and the misconnections of different trace segments (e.g., regions I–III in Fig. 15(a)) have been effectively improved.

2.3.3 Trace segment connection based on oriented extension

Trace linearization can generate broken trace segments. Therefore, trace segment connection is performed to improve the trace extensivity. Li et al. (2016) used the average edge length of triangular patches for connection evaluation. However, it requires tedious triangulation for the raw point cloud. This section used an oriented extension of traces directly based on the raw point cloud without triangulation. As shown in Fig. 17, the specific procedures are as follows.

(1) Given the contracted skeleton point set $C_p = \{c_1, \cdots, c_{N_b}\}$, the average adjacent distance M_d calculated by Eq. (22), the k_{nn} nearest point set $K_c = \{k_{c,1}, \cdots, k_{c,N_b}\}$, the k_{nn} nearest distance set $D_c = \{d_{c,1}, \cdots, d_{c,N_b}\}$, the trace segments after linearization S_{gl}, the directional vector set of trace segments $D_{vt} = \{\boldsymbol{d}_{vt,1}, \cdots, \boldsymbol{d}_{vt,N_t}\}$ (N_t denotes the trace segment number), and the trace index set $S_I = \{s_{I,1}, \cdots, s_{I,N_t}\}$ of each point in C_p.

(2) Given the endpoint e_i and the directional vector $\boldsymbol{d}_{vt,i}$ of the ith trace segment.

(3) Search for K_c and D_c to find the point set $F_p = \{f_{p,1}, \cdots, f_{p,N_{fp}}\}$ (N_{fp} denotes the element number in F_p) with the distance smaller than $15M_d$ (Li et al., 2016).

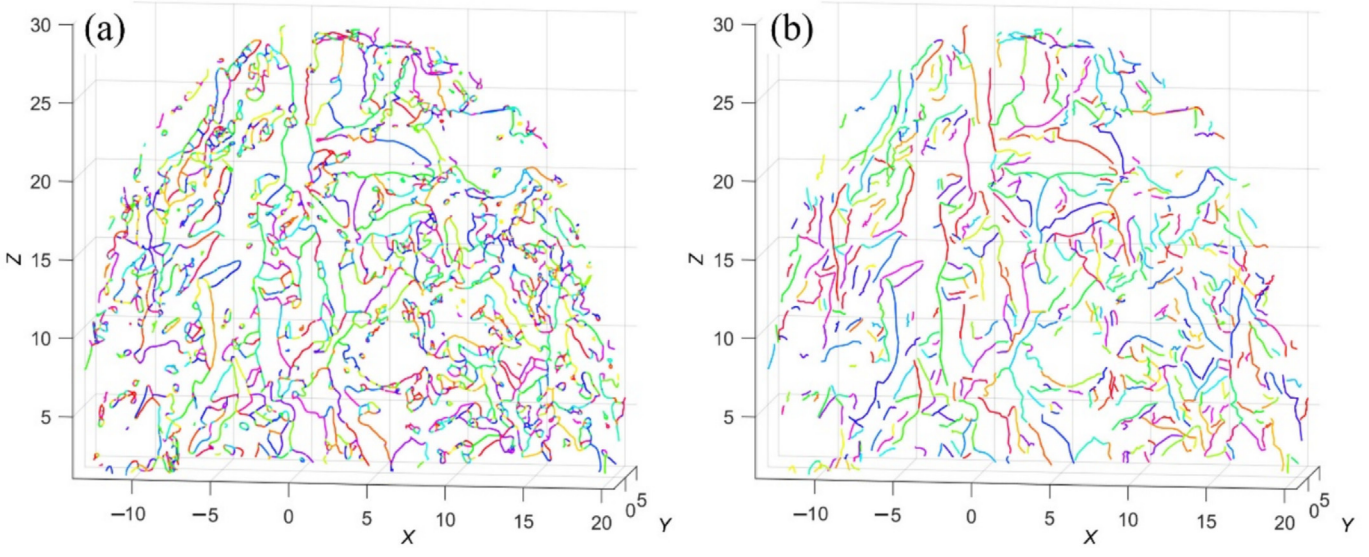

Fig. 18. Oriented extension results of trace connection. (a) Original trace segments, and (b) 10-point trace segments with noise reduction.

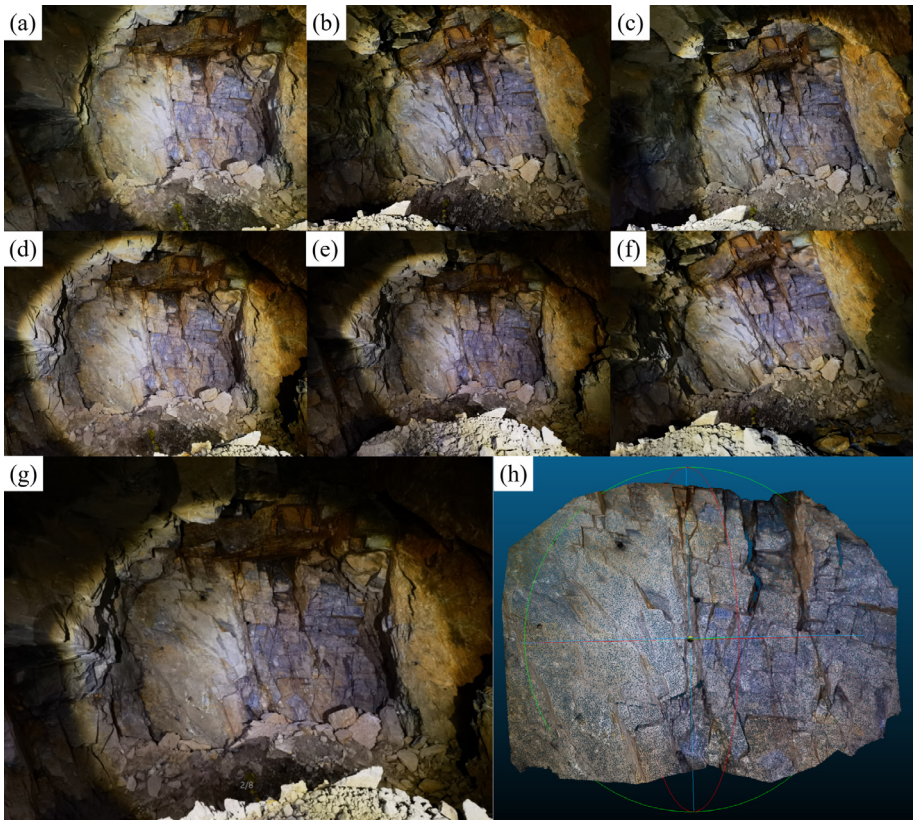

Fig. 19. Image acquisition and 3D reconstruction of case 2. (a)–(g) Image sequence of the excavation face, and (h) 3D reconstructed points in ROI.

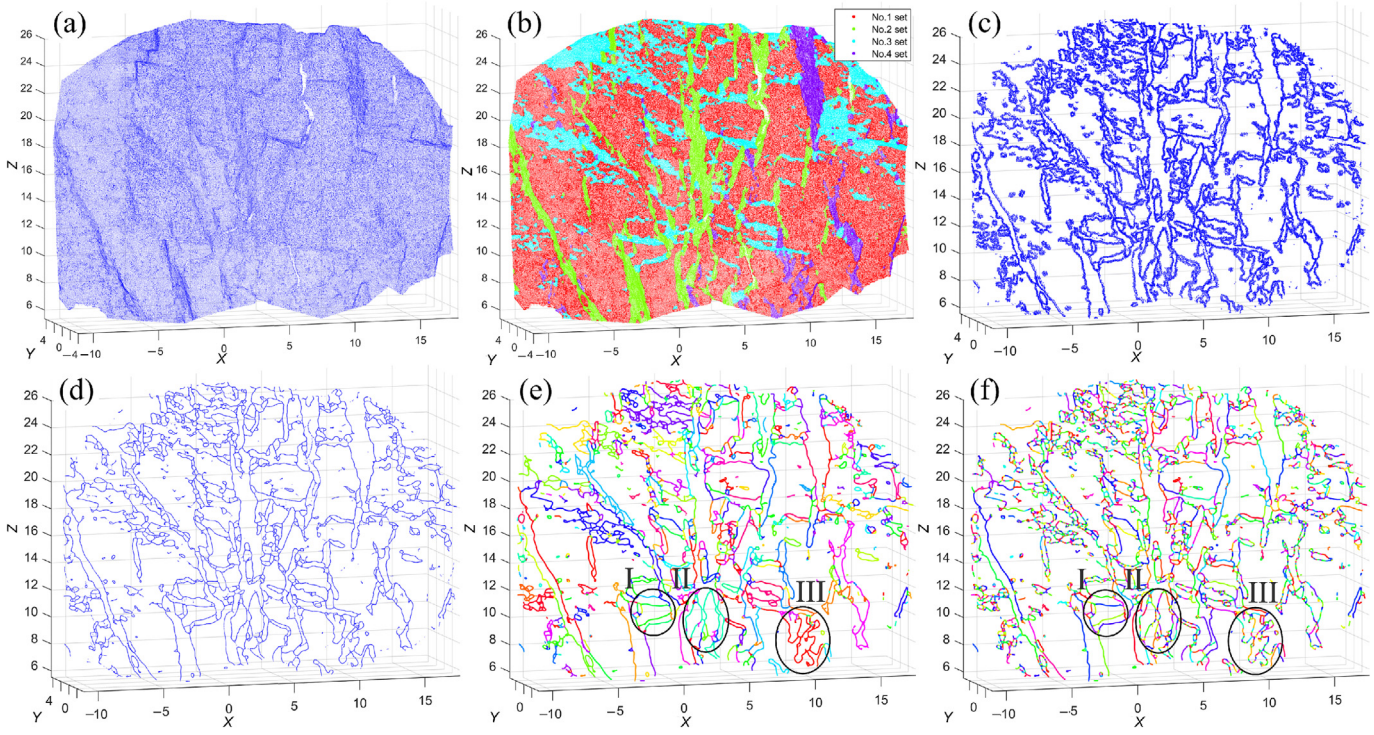

Fig. 20. Trace extraction process of case 2. (a) Raw point cloud, (b) optimal orientation grouping result of $k_{opt} = 4$, (c) boundary points of the grouping results, (d) skeleton points after PWI-OC, (e) initial generation of trace segments, and (f) trace segment linearization.

(4) According to F_p, find the trace segments in S_I that connect with e_i, and let the trace segment set as $F_I = \{f_{I,1}, \cdots, f_{I,N_{fI}}\}$ (N_{fI} denotes the element number in F_I).
(5) If $F_I = \varnothing$, then turn to step (7); if $F_I \neq \varnothing$, then turn to step (6).
(6) Calculate the angle $A_g = \{a_{g,1}, \cdots, a_{g,N_{fI}}\}$ between $\boldsymbol{d}_{vt,i}$ and each directional vector $\boldsymbol{d}_{vt,j}$ of traces in F_I. Given $a_w = \min(A_g)$ (as shown in Fig. 17), if $a_w \leq 30°$ (Li et al., 2016), then connect the ith trace segment with the trace segment $f_{I,w} \in F_I$ and repeat step (2); if $a_w > 30°$, then turn to step (7).
(7) Stop the connection for the ith trace segment.
(8) Perform the connection for all trace segments by repeating steps (2)–(7).

Figure 18(a) shows the oriented extension results of the trace connection. In order to smoothly display the main morphological features of traces, each trace is sampled using 10 points with equal spacing. The 10-point trace result (with the denoising of removing trace segments shorter than $50M_d$) is shown in Fig. 18(b).

3 Case study

In order to validate the proposed method, another excavation face case from a mining tunnel is analyzed for comparison with the traditional method of Zhang et al. (2020). This case (case 2) was collected by using an iPhone 11 mobile phone to take 7 images (Fig. 19(a)–(g)) with the resolution of 4032 × 3024 at different positions in front of the excavation face. A total of 1 001 305 3D points were reconstructed using the Meshroom software. The ROI contained 271 313 points after manual cropping.

Figure 19 shows the identification process of the proposed method for case 2. It can be seen that the optimal

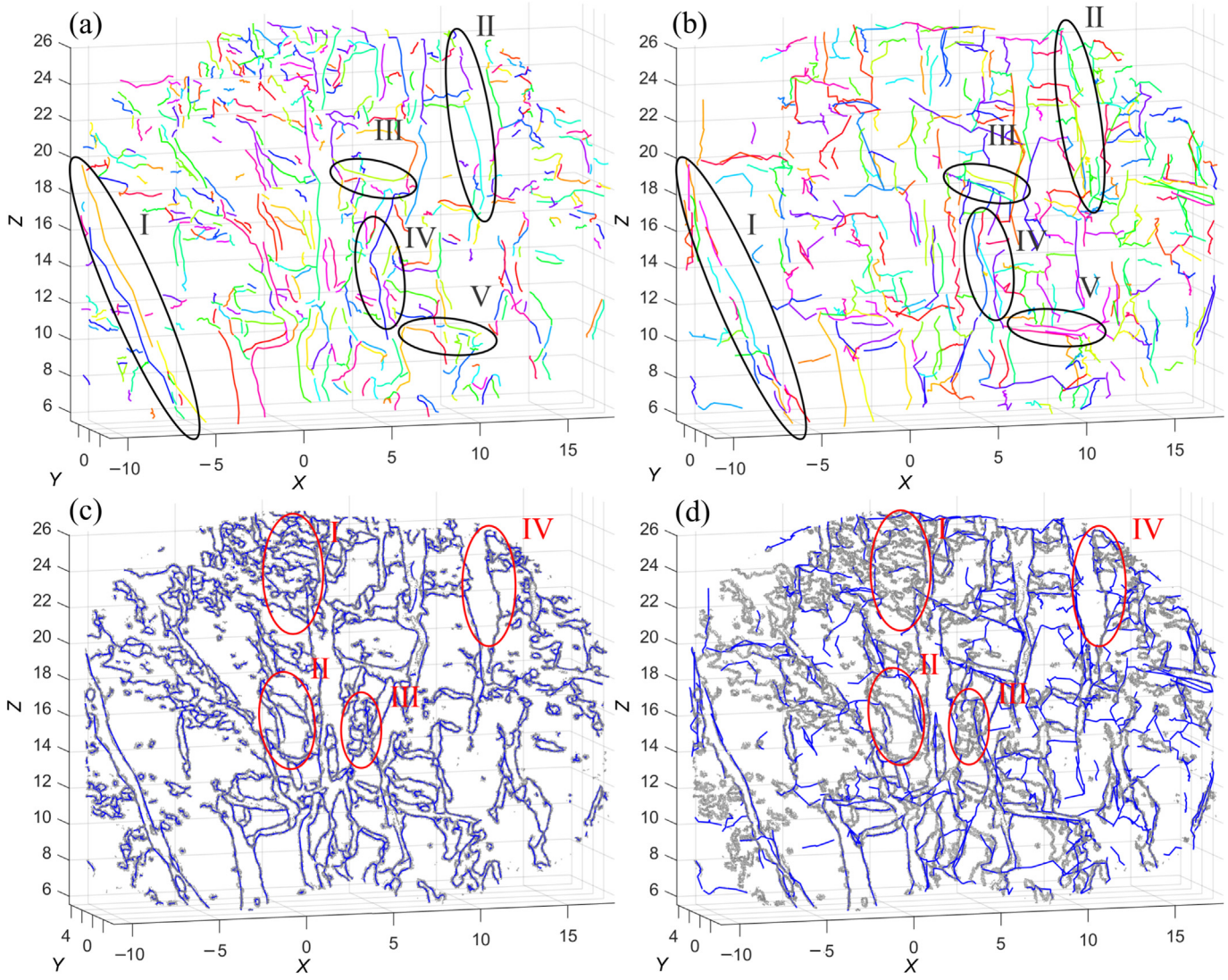

Fig. 21. Comparison of trace results by different methods. (a), (b) Trace overlapping comparison of the proposed method and Zhang et al. (2020), respectively, and (c), (d) comparison of the matching between traces and orientation grouping of the proposed method and Zhang et al. (2020), respectively. The grey points in (c) and (d) denote the boundary points of the optimal grouping results.

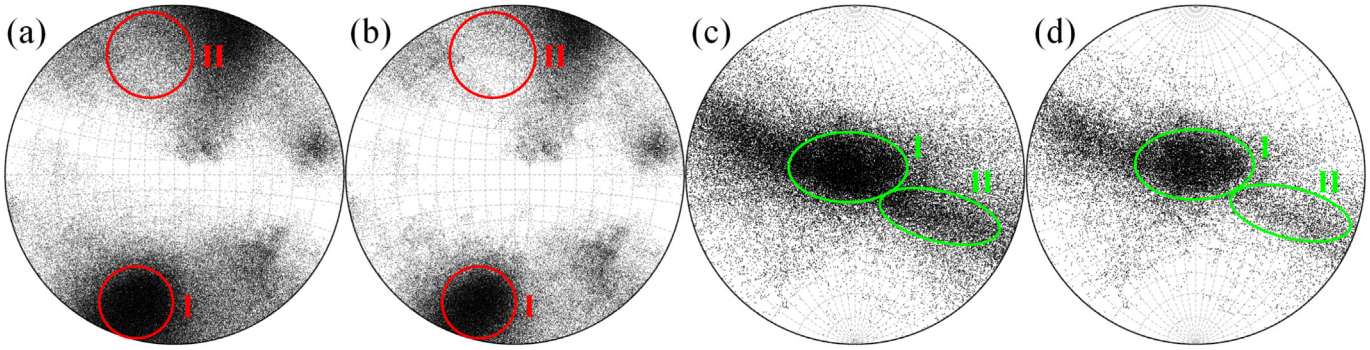

Fig. 22. Comparison of adaptive and non-adaptive vectors on the stereographic projection plane. (a) Non-adaptive normal vectors of case 2, (b) adaptive normal vectors of case 2, (c) non-adaptive directional vectors of case 2, and (d) adaptive directional vectors of case 2.

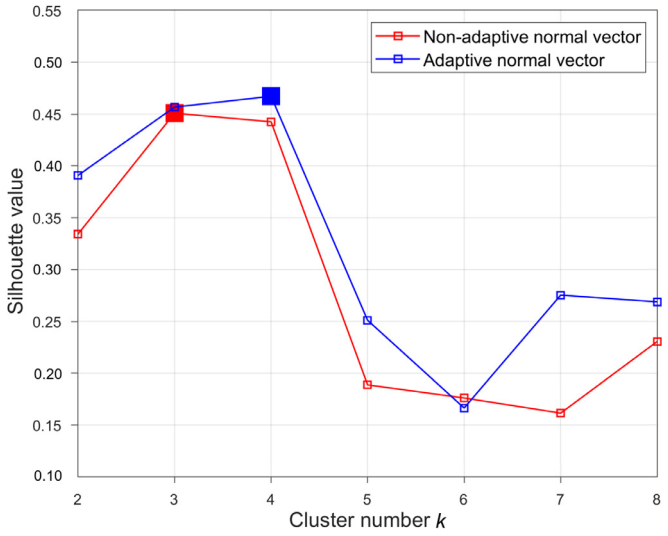

Fig. 23. Silhouette comparison for the orientation grouping of adaptive and non-adaptive normal vectors of case 2.

orientation grouping number is calculated as 4 (Fig. 20(b)). The PWI-OC procedure generates thin and continuous point skeletons (Fig. 20(d)) from the boundary points (Fig. 20(c)) of orientation grouping. As shown in Fig. 20 (e), the initial trace generation contains false connections of different traces (e.g., regions of I to III). The linearization effectively divides them into linearized segments (e.g., the regions of I to III in Fig. 20(f)). Figure 21(a) shows the generated 10-point trace segments after trace connection of the proposed method. Compared with traces (Fig. 21(b)) recognized by Zhang et al. (2020), it can be seen that the heavy overlapping of different traces (e.g., regions of I to V in Fig. 21(b)) can be effectively reduced to the accurate traces (e.g., regions I to V in Fig. 21(a)) with no obvious overlapping. In addition, Fig. 21(c) and (d) shows the comparison of the matching between traces and orientation grouping. The method of Zhang et al. (2020) shows an obvious trace lacking in the orientation grouping boundaries (e.g., regions of I to III in Fig. 21 (d)). Comparatively, traces of the proposed method are

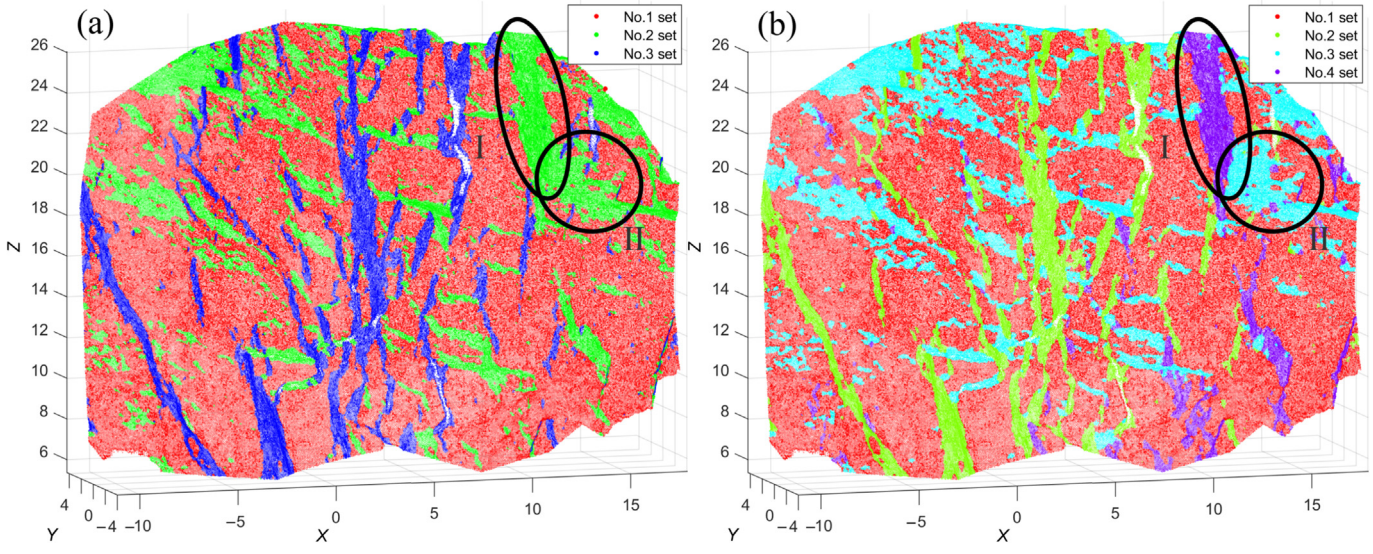

Fig. 24. Optimal grouping results of case 2 using different normal vectors. (a) Non-adaptive normal vectors with $k_{opt} = 3$, and (b) adaptive normal vectors with $k_{opt} = 4$.

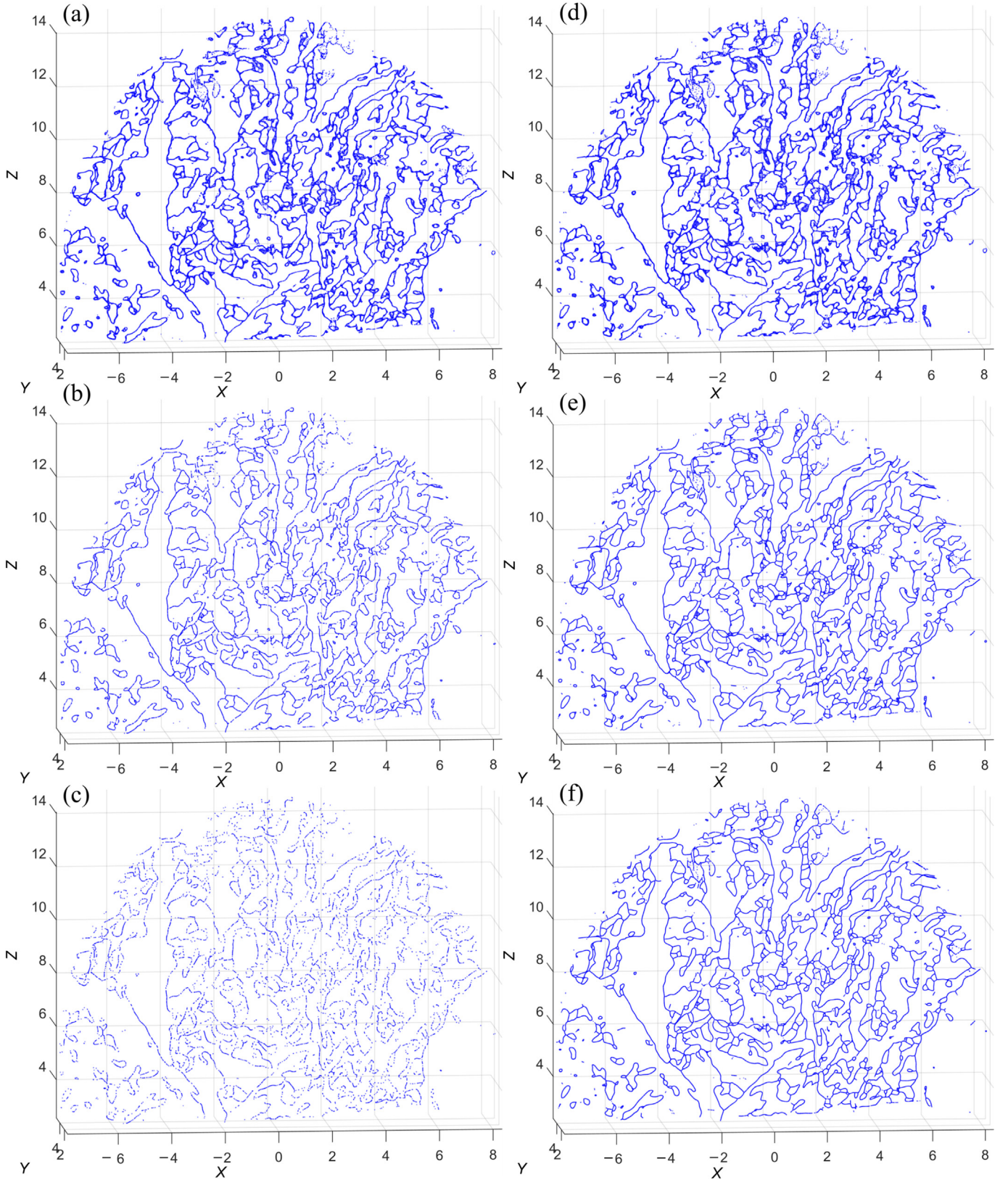

Fig. 25. Comparison of the point cloud contraction by different methods. (a)–(c) Laplacian smoothing by Guo et al. (2018) with the iteration times of 1, 3, and 5, and (d)–(f) the proposed PWI-OC method with the iteration times of 1, 3, and 5.

Table 1
Comparison of axial and radial displacements of different methods.

Methods	Metrics	Contraction numbers				
		1	2	3	4	5
Laplacian smoothing method (Guo et al., 2018)	AD	0.0113	0.0169	0.0231	0.0284	0.0480
	RD	0.0226	0.0253	0.0273	0.0288	0.0526
	RD/AD	2.00	1.49	1.18	1.01	1.09
The proposed method	AD	0.0019	0.0028	0.0034	0.0040	0.0044
	RD	0.0226	0.0254	0.0272	0.0287	0.0299
	RD/AD	11.89	9.07	8.00	7.17	6.79

located exactly on the boundary points (e.g., regions of I to III in Fig. 21(c)), which shows a better matching between traces and orientation grouping than Zhang et al. (2020).

4 Discussions

4.1 Comparison between adaptive and non-adaptive vectors

The difference between adaptive and non-adaptive vectors can be visualized by their distributions on the stereographic projection plane (SPP). For case 2, it can be seen that the adaptive normal vectors on SPP show good stability of dense regions (e.g., region I in Fig. 22(a)) and show effective noise reduction on sparse regions (e.g., region II in Fig. 22 (a)) compared with the non-adaptive normal vectors (Fig. 22(b)). The adaptive directional vectors also show similar dense region (e.g., region I in Fig. 22(c)) stability and noise reduction (e.g., region II in Fig. 22(c)) compared with non-adaptive directional vectors (e.g., Fig. 22 (d)). The noise reduction is caused by the neighbor assignment of normal and directional vectors of sharp points. The mean neighbor angle (MNA) of k_{nn} nearest neighbor vectors is used to evaluate both the concentration of vectors. The MNA of normal vectors is reduced from 0.29° to 0.22°, and the MNA of directional vectors is reduced from 0.67° to 0.49°, which indicates an obvious reduction of noisy vectors. A high concentration of normal vectors can contribute to a good quality of orientation grouping. As shown in Fig. 23, except for $k = 6$, the silhouette values of adaptive normal vectors of all cluster numbers are larger than non-adaptive normal vectors, indicating the contribution of adaptive vectors for orientation grouping. In addition, the optimal grouping number of adaptive normal vectors is 4 (Fig. 23), which can effectively segment the different discontinuities in regions I and II in Fig. 24(b). However, the optimal grouping number of non-adaptive normal vectors is 3 (Fig. 23), which falsely segments the different discontinuities (in regions I and II in Fig. 24(a)) into the same orientation group.

In conclusion, the adaptive vectors show better dense region stability and noise reduction in sparse regions than non-adaptive vectors, which effectively contributes to the accuracy improvement in orientation grouping.

4.2 Verification for the PWI-OC method

In order to validate the effectiveness of PWI-OC, the method of Guo et al. (2018) with Laplacian smoothing for point thinning is used for comparison. Figure 25(a)–(c) shows the contraction effects with 1, 3, and 5 iterations, respectively. It can be seen that the distribution of skeleton points along the axial direction of traces becomes sparser as the iteration number increases. This is mainly because Guo et al. (2018) do not restrict the contraction direction, making the contraction displacement contain both the radial displacement (RD) perpendicular to the trace and the axial displacement (AD) parallel to the trace. Although RD can thin the point cloud skeleton as the iteration number increases, AD can lead to an inhomogeneous distribution of the skeleton points along the axial direction of traces, which reduces the shape conformality of point skeletons. Also, as shown in Table 1, after 5 times of contraction, RD/AD of the proposed method is 6.79, which is obviously larger than the Laplacian smoothing method with RD/AD = 1.09. The unevenly distributed point skeleton is not conducive to generating complete traces. In contrast, the proposed method carries out a modified displacement which eliminates AD during the point cloud contraction (Fig. 13). As shown in Fig. 25(d)–(f), the contraction iterations only refine the point cloud skeleton and effectively keep the homogeneous distribution of points along the axial direction of skeletons. Therefore, the proposed PWI-OC has better iterative conformality than Guo et al. (2018), contributing to generating more coherent and robust trace segments.

5 Conclusions

This paper proposes an adaptive oriented contraction method for automatic trace recognition of rock tunnel excavation faces based on 3D point cloud, named AOC. The main contributions include:

(1) an adaptive vector method based on neighbor assignment to improve the accuracy of normal and directional vectors at sharp points.

(2) a PCA-weighted iterative oriented contraction (PWI-OC) method to improve the quality of point cloud skeleton extraction.
(3) a sparse growing method including initial generation, linearization, and oriented extension for trace segment extraction.

Three rock tunnel excavation face cases from a mining tunnel and two railway tunnels are used to analyze and compare the proposed method with traditional methods. Case 1 is analyzed to illustrate the significance of each step of the proposed method. Case 2 is adopted to validate the proposed method in the robustness to overlapping traces and the consistency with orientation grouping results. The significance of adaptive normal vectors is validated for improving the quality of discontinuity orientation grouping. The proposed PWI-OC method is verified to generate accurate point skeletons with good iterative conformality. The results show that the proposed method can be effectively applied to the discontinuity trace identification in rock engineering with better accuracy than traditional methods.

Future work can focus on the improvement of computational efficiency and the robustness to inaccurate orientation grouping results.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

CRediT authorship contribution statement

Keshen Zhang: Visualization, Software, Conceptualization, Writing – original draft, Validation, Methodology. **Min Zhang:** Funding acquisition, Supervision, Investigation. **Lianyang Zhang:** Writing – review & editing, Supervision, Resources. **Wei Wu:** Supervision, Resources, Writing – review & editing.

Declaration of competing interest

Dr. Lianyang Zhang is a managing editor for *Underground Space* and was not involved in the editorial review or the decision to publish this article. All authors declare that there are no competing interests.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grant Nos. 52508444, 42272338, 41827807, and 41902275); Shandong Provincial Natural Science Foundation (Grant No. ZR2025QC1138); Shanghai Sailing Program (Grant No. 18YF1424400); Joint Fund for Basic Research of High-speed Railway of National Natural Science Foundation of China; China Railway Corporation (Grant No. U1934212); China State Railway Group Co., Ltd. (Grant No. P2019G038); Department of Transportation of Zhejiang Province (Project No. 202213); China Railway First Survey and Design Institute Group Co., Ltd. (19-21-1, 2022KY53ZD(CYH)-10); China Railway Tunnel Group Co., Ltd. (CZ02-02-08); PowChina Hebei Transportation Highway Investment Development Co., Ltd. (TH-201908); Sichuan Railway Investment Group Co., Ltd. (SRIG2019GG0004); The Science and Technology major program of Guizhou Province [2018] 3011.

References

Assali, P., Grussenmeyer, P., Villemin, T., Pollet, N., & Viguier, F. (2016). Solid images for geostructural mapping and key block modeling of rock discontinuities. *Computers & Geosciences, 89*, 21–31.
Ball, G. H., & Hall, D. J. (1965). *ISODATA: A novel method of data analysis and pattern classification*. Stanford Research Institute.
Barton, N. R. (1978). Suggested methods for the quantitative description of discontinuities in rock masses: International Society for Rock Mechanics. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts, 15*, 319–368.
Battulwar, R., Zare-Naghadehi, M., Emami, E., & Sattarvand, J. (2021). A state-of-the-art review of automated extraction of rock mass discontinuity characteristics using three-dimensional surface models. *Journal of Rock Mechanics and Geotechnical Engineering, 13*(4), 920–936.
Bolkas, D., Vazaios, I., Peidou, A., & Vlachopoulos, N. (2018). Detection of rock discontinuity traces using terrestrial LiDAR data and space-frequency transforms. *Geotechnical and Geological Engineering, 36*(3), 1745–1765.
Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics, 3*(1), 1–27.
Cao, T., Xiao, A. C., Wu, L., & Mao, L. G. (2017). Automatic fracture detection based on terrestrial laser scanning data: A new method and case study. *Computers & Geosciences, 106*, 209–216.
Crosta, G. (1997). Evaluating rock mass geometry from photographic images. *Rock Mechanics and Rock Engineering, 30*(1), 35–58.
Cui, X. J., & Yan, E. C. (2020). Fuzzy C-means cluster analysis based on variable length string genetic algorithm for the grouping of rock discontinuity sets. *KSCE Journal of Civil Engineering, 24*(11), 3237–3246.
Daghigh, H., Tannant, D. D., Daghigh, V., Lichti, D. D., & Lindenbergh, R. (2022). A critical review of discontinuity plane extraction from 3D point cloud data of rock mass surfaces. *Computers & Geosciences, 169*, 105241.
Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1*(2), 224–227.
Ferrero, A. M., Forlani, G., Roncella, R., & Voyat, H. I. (2009). Advanced geostructural survey methods applied to rock mass characterization. *Rock Mechanics and Rock Engineering, 42*(4), 631–665.
Fisher, J. E., Shakoor, A., & Watts, C. F. (2014). Comparing discontinuity orientation data collected by terrestrial LiDAR and transit compass methods. *Engineering Geology, 181*, 78–92.
Gao, F., Chen, D. P., Zhou, K. P., Niu, W. J., & Liu, H. W. (2019). A fast clustering method for identifying rock discontinuity sets. *KSCE Journal of Civil Engineering, 23*(2), 556–566.
Gigli, G., & Casagli, N. (2011). Semi-automatic extraction of rock mass structural data from high resolution LIDAR point clouds. *International Journal of Rock Mechanics and Mining Sciences, 48*(2), 187–198.
González, Á. (2010). Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences, 42*(1), 49–64.
Guo, J. T., Liu, Y. H., Wu, L. X., Liu, S. J., Yang, T. H., Zhu, W. C., & Zhang, Z. R. (2019). A geometry- and texture-based automatic discontinuity trace extraction method for rock mass point cloud. *International Journal of Rock Mechanics and Mining Sciences, 124*, 104132.

Guo, J. T., Wu, L. X., Zhang, M. M., Liu, S. J., & Sun, X. Y. (2018). Towards automatic discontinuity trace extraction from rock mass point cloud without triangulation. *International Journal of Rock Mechanics and Mining Sciences, 112*, 226–237.

Jimenez, R. (2008). Fuzzy spectral clustering for identification of rock discontinuity sets. *Rock Mechanics and Rock Engineering, 41*(6), 929–939.

Jimenez-Rodriguez, R., & Sitar, N. (2006). A spectral method for clustering of rock discontinuity sets. *International Journal of Rock Mechanics and Mining Sciences, 43*(7), 1052–1061.

Kemeny, J., & Post, R. (2003). Estimating three-dimensional rock discontinuity orientation from digital images of fracture traces. *Computers & Geosciences, 29*(1), 65–77.

Kemeny, J., Turner, K., & Norton, B. (2006). LIDAR for rock mass characterization: Hardware, software, accuracy and best-practices. In *Proceedings of the Workshop on Laser and Photogrammetric Methods for Rock Face Characterization* (pp. 49–62).

Khaloo, A., & Lattanzi, D. (2017). Robust normal estimation and region growing segmentation of infrastructure 3D point cloud models. *Advanced Engineering Informatics, 34*, 1–16.

Kong, D. H., Wu, F. Q., & Saroglou, C. (2020). Automatic identification and characterization of discontinuities in rock masses from 3D point clouds. *Engineering Geology, 265*, 105442.

Lato, M. J., Diederichs, M. S., Hutchinson, D. J., & Harrap, R. (2012). Evaluating roadside rockmasses for rockfall hazards using LiDAR data: Optimizing data collection and processing protocols. *Natural Hazards, 60*(3), 831–864.

Lemy, F., & Hadjigeorgiou, J. (2003). Discontinuity trace map construction using photographs of rock exposures. *International Journal of Rock Mechanics and Mining Sciences, 40*(6), 903–917.

Li, X. J., Chen, J. Q., & Zhu, H. H. (2016). A new method for automated discontinuity trace mapping on rock mass 3D surface model. *Computers & Geosciences, 89*, 118–131.

Li, X. J., Chen, Z. Y., Chen, J. Q., & Zhu, H. (2019). Automatic characterization of rock mass discontinuities using 3D point clouds. *Engineering Geology, 259*, 105131.

Li, X. J., Zuo, Y. L., Zhuang, X. Y., & Zhu, H. H. (2014). Estimation of fracture trace length distributions using probability weighted moments and L-moments. *Engineering Geology, 168*, 69–85.

Mauldon, M. (1998). Estimating mean fracture trace length and density from observations in convex windows. *Rock Mechanics and Rock Engineering, 31*(4), 201–216.

Nurunnabi, A., West, G., & Belton, D. (2015). Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. *Pattern Recognition, 48*(4), 1404–1419.

Priest, S. D. (1993). *Discontinuity analysis for rock engineering*. Springer.

Riquelme, A. J., Abellán, A., Tomás, R., & Jaboyedoff, M. (2014). A new approach for semi-automatic rock mass joints recognition from 3D point clouds. *Computers & Geosciences, 68*, 38–52.

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science, 344*(6191), 1492–1496.

Roncella, R., Forlani, G., & Remondino, F. (2005). Photogrammetry for geological applications: Automatic retrieval of discontinuity orientation in rock slopes. In *Proceedings of SPIE-The International Society for Optical Engineering* (pp. 17–27). January 16–20, 2005. San Jose, CA, USA.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20*, 53–65.

Singh, S. K., Banerjee, B. P., Lato, M. J., Sammut, C., & Raval, S. (2022). Automated rock mass discontinuity set characterisation using amplitude and phase decomposition of point cloud data. *International Journal of Rock Mechanics and Mining Sciences, 152*, 105072.

Singh, S. K., Raval, S., & Banerjee, B. P. (2021). Automated structural discontinuity mapping in a rock face occluded by vegetation using mobile laser scanning. *Engineering Geology, 285*, 106040.

Sturzenegger, M., & Stead, D. (2009). Close-range terrestrial digital photogrammetry and terrestrial laser scanning for discontinuity characterization on rock cuts. *Engineering Geology, 106*(3–4), 163–182.

Umili, G., Ferrero, A., & Einstein, H. H. (2013). A new method for automatic discontinuity traces sampling on rock mass 3D model. *Computers & Geosciences, 51*, 182–192.

Vöge, M., Lato, M. J., & Diederichs, M. S. (2013). Automated rockmass discontinuity mapping from 3-dimensional surface data. *Engineering Geology, 164*, 155–162.

Wang, Y. T., Feng, H. Y., Delorme, F. É., & Engin, S. (2013). An adaptive normal estimation method for scanned point clouds with sharp features. *Computer-Aided Design, 45*(11), 1333–1348.

Wu, W., Zhang, K. S., & Zhu, H. H. (2020). A fast automatic extraction method for rock mass discontinuity orientation using fast k-means++ and fast silhouette based on 3D point cloud. *IOP Conference Series: Earth and Environmental Science, 570*(5), 052075.

Xie, X. L., & Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(8), 841–847.

Zhang, K. S., Wu, W., Liu, Y. S., Xie, T., Zhou, J. B., & Zhu, H. H. (2024a). OCFMD: An automatic optimal clustering method of discontinuity orientation based on fisher mixed distribution. *Rock Mechanics and Rock Engineering, 57*(3), 1735–1763.

Zhang, K. S., Wu, W., Zhu, H. H., Zhang, L. Y., Li, X. J., & Zhang, H. (2020). A modified method of discontinuity trace mapping using three-dimensional point clouds of rock mass surfaces. *Journal of Rock Mechanics and Geotechnical Engineering, 12*(3), 571–586.

Zhang, L., & Einstein, H. H. (1998). Estimating the mean trace length of rock discontinuities. *Rock Mechanics and Rock Engineering, 31*(4), 217–235.

Zhang, Q., Pei, Y. C., Wang, X. J., Li, X. J., & Shen, Y. X. (2024b). Probabilistic prediction on three-dimensional roughness of discontinuity based on two-dimensional traces under rock tunnel excavation based on Bayesian theory. *Underground Space, 14*, 338–356.

Zhang, Q., Wang, X. J., Zhu, H. H., Cai, W. Q., & Li, X. J. (2023). Estimation of three-dimensional diameter distributions of multiple fracture sets clustered by a multi-level clustering method. *Acta Geotechnica, 18*, 4429–4452.

Zhang, Q., Wang, X. J., Zhu, H. H., Zhang, K. S., & Li, X. J. (2022). Mixture distribution model for three-dimensional geometric attributes of multiple discontinuity sets based on trace data of rock mass. *Engineering Geology, 311*, 106915.

Zhang, Q., Wang, X. J., He, L., & Tian, L. G. (2021). Estimation of fracture orientation distributions from a sampling window based on geometric probabilistic method. *Rock Mechanics and Rock Engineering, 54*(6), 3051–3075.