

An Efficient Global Constraint Approach for Robust Contour Feature Points Extraction of Point Cloud

Xijiang Chen[✉] and Bufan Zhao[✉]

Abstract—The contour feature points of object point clouds are the main features of human perception on target and play an important role in many fields, such as indoor model reconstruction, and object detection and location. In this article, we present a new method to extract the contour feature points of point cloud, which mainly includes two main contents: 1) the conspicuous and inconspicuous boundary points are extracted according to the characteristics of distribution of the azimuth between adjacent vectors in 2-D view and 2) according to the direction of main feature vector, a 2-D projection plane of adjacent points in the bounding sphere is constructed, and the crease points are extracted according to the constraint parameters model of distribution mechanism of adjacent points in the 2-D view. We evaluate the performance of the proposed method using objects of different sizes in real-world scenarios. Simultaneously, the extraction effect of contour feature points is compared with other methods, and the results show that the extraction and antinoise performance of the proposed method are superior to the other methods. Simultaneously, it is suitable not only for regular flat-shaped buildings but also for objects with irregular curvilinear architecture. Moreover, the proposed method involves only one parameter that needs to be tuned, and the parameter can be quickly obtained based on the distance resolution.

Index Terms—Boundary points, crease points, feature points, point cloud, principal component analysis (PCA).

I. INTRODUCTION

T-D LASER scanning technology can record the 3-D coordinates of a large number of dense points on the surface of an object. The technology has been widely used in various fields, including slope monitoring [1], cultural relic protection [2], building monitoring [3], digital city [4], and so on. Although 3-D laser scanning can obtain a large amount of point cloud data, not all points are useful. The feature points of the point cloud mainly reflect the geometric properties of the point cloud model, mainly including crease and boundary points [5]. The feature points of point clouds play an important role in many fields, such as building contour extraction [6], building model reconstruction [7], road extraction [8], and registration [9]. In the field of image, there are currently numerous mature feature extraction technologies

Manuscript received 21 May 2023; revised 24 July 2023; accepted 21 August 2023. Date of publication 24 August 2023; date of current version 8 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 42171428; and in part by the Open Fund of Key Laboratory of Mine Environmental Monitoring and Improving Around Poyang Lake, Ministry of Natural Resources, under Grant MEMI-2021-2022-04. (*Corresponding author: Bufan Zhao.*)

The authors are with the School of Safety Science and Emergency Management, Wuhan University of Technology, Wuhan 430070, China (e-mail: 201102140030@whu.edu.cn; zbf@whut.edu.cn).

Digital Object Identifier 10.1109/TGRS.2023.3308376

available [10]. However, the research on 3-D point cloud feature extraction is not very mature. The main reason is that 3-D point clouds are different from 2-D images. The pixels of 2-D images are closely related to their neighbors, while unorganized point clouds lack connectivity information. Therefore, it is difficult to obtain the feature points of objects quickly and accurately from a large number of point cloud data. The extraction of point cloud contour feature points mainly includes two types of methods.

1) The point cloud is converted into an image, and the features are determined using the image feature extraction method. Subsequently, the image features are converted back into point cloud features.

2) Feature extraction is implemented directly on the 3-D point cloud. The method mainly uses the domain information of each point to determine the feature attributes of each point.

The disadvantage of the first type of approach is that it requires high-quality images, and it is sensitive to noise. The difficulty of the second type of method is that the number of neighboring points of each point cannot be accurately determined, and the accuracy of point cloud features determined by the domain information is easily affected by noise.

In view of this, we propose a novel method to extract contour feature points of point clouds, which can extract boundary and crease points. In this article, we do not perform direct point cloud contour feature point extraction on the original point cloud nor do we convert the point cloud into images for contour feature point extraction. Instead, we not only use the domain characteristics of each point but also use the 2-D point characteristics after the projection. The proposed method mainly includes five steps. First, we perform the feature decomposition according to the neighboring points and center points of the target points, and obtain the three eigenvalues and corresponding eigenvectors. According to the direction of the eigenvector corresponding to the minimum eigenvalue, the projection plane for the extraction of boundary points can be obtained. Second, the target point and its neighboring points are projected onto the projection plane, and the vectors between the target point and its neighboring points are constructed. The include angles between the adjacent vectors are calculated, and the boundary points are extracted according to the angle distribution between the adjacent vectors. Third, the optimal radius of the bounding sphere is judged according to the distribution of point clouds in the bounding sphere. We perform the feature decomposition according to the center point and points inside the bounding sphere. According to the direction of the eigenvector corresponding to the largest

eigenvalue obtained by decomposition, the projection plane used for crease point extraction is determined. Fourth, the points inside the bounding sphere of the point cloud are projected onto the projection plane and then rotated to the 2-D coordinate system using the quaternion algorithm. Simultaneously, the line equation is obtained according to the two farthest points of the 2-D coordinate system inside the bounding sphere. We use the distance standard deviation of the points to the line and the number of points on both sides of the line to judge the crease point cloud. Finally, the creased points are extracted according to the three constraint parameters.

II. RELATED WORKS

A. Method Based on Point Cloud 3-D Mesh Model

This type of method first used a point cloud to construct the 3-D mesh model. Then, the geometric topological information of each point was obtained by the 3-D point cloud model. Finally, the geometric topological information between adjacent points in the 3-D model was utilized to extract the feature points of a point cloud. Hildebrandt et al. [11] used discrete differential geometry to calculate the average third-order surface derivative and filtered higher order surface derivative to improve the stability of feature point extraction. Ohtake et al. [12] employed the implicit surface fitting method to compute the curvature of the point cloud model and extracted valleys and ridges in the point cloud mesh model based on the maximum and minimum curvature. By the same token, Ho et al. [13] adopted the multiscale curvature method to extract point cloud features, and the maximum and minimum of local curvature were employed to determine the feature points. The advantage of this method is that it can reduce the influence of point cloud noise to a certain extent. Gao et al. [14] proposed a new method for extracting feature lines of point clouds based on the point cloud mesh model. The method reduced the discontinuities and fluctuations of the feature lines by interpolating the vertices of the mesh, leading to the effective extraction of feature points. The 3-D mesh model of the point cloud smooths out the sharp features of the object's point cloud. In order to keep sharp features, Salman et al. [15] used the feature detection method based on the covariance matrix of the Voronoi diagram to extract sharp features from point clouds because the point cloud noise will affect the accuracy of point cloud 3-D mesh model, and the generation of point cloud 3-D mesh model will increase the computational cost. In view of this, many scholars have studied feature extraction directly from discrete 3-D point cloud data.

B. Method Based Directly on Discrete Point Cloud

1) Neighboring Point Feature-Based Methods:

Gumhold et al. [16] used the eigenvalues of the covariance matrix of neighboring points of each point to determine the feature points. The advantage of this method is that it does not need to reconstruct the surface in advance and only needs to calculate the neighbor graph between neighboring points. Similarly, Demarsin et al. [17] estimated sharp edges using normal vectors and clustered candidate feature points using the

region-growing algorithm. Pauly et al. [18] used multiscale covariance analysis of different local neighborhoods to determine the feature points. Liu and Jin [19] used the difference of adjacent normal vectors to determine the multiscale factor of feature line extraction and realized the detection of feature points based on this factor. Hackel et al. [20] used eigenvalue decomposition to extract a set of features from the neighboring points and used a binary classifier to predict the score of each point belonging to the contour. Elkhrachy [21] used neighboring points to calculate the normal vector of the target point, and then, the object edge points are defined by the points where normal vector changes exceed a given threshold value. The disadvantage of this type of method is that the extraction effect of sharp edge points is very sensitive to the accuracy of normal vector and neighborhood size. However, it is difficult to guarantee the accuracy of the normal vector of each point.

Curvature can directly express the surface features of point clouds. Therefore, many scholars employed curvature to extract feature points. For example, Altantsetseg et al. [22] used a truncated Fourier series to calculate the curvature of every point and then detected the feature points according to the curvature. Prochazkova et al. [23] used the log-least squares regression algorithm to determine the best input value of the curvature threshold and used the traditional region-growing algorithm to detect the point cloud features. Bazazian et al. [24] calculated the covariance matrix defined by the k-nearest neighbors (KNNs) of each point and detected the point cloud feature according to the eigenvalues of the covariance matrix. Kim [25] first adopted the moving least squares method to fit the local plane to obtain the curvature. Then, the neighborhood information of valley feature points was constructed by the Voronoi diagram. Finally, the curvature derivative of Delaunay is used to detect the ridge feature points.

In addition, some scholars introduced tensor voting into feature extraction [26], and the surface points, edge points, and corner points can be distinguished according to the eigenvalues obtained from the tensor voting matrix. Lin et al. [27] proposed an analytical tensor voting algorithm based on the eigenvalue comparison, which reduced the complexity and heavy computational burden of traditional tensor voting to extract feature points from unorganized point clouds. The advantage of this method is its low computational cost and easy implementation.

2) Clustering Segmentation-Based Methods: Demarsin et al. [17] first calculated the normal vector of the point and its neighborhood and then performed clustering segmentation according to the angle threshold of the normal vector to segment the sharp points. This method is suitable for extracting closed sharp feature points. Lin et al. [6] used the region-growing algorithm to segment the point cloud into a series of small planes, and then, the feature points were determined by finding the α -shape vertices of each facet. This method can only be applied to the extraction of crease points of the plane area. Lu et al. [28] first used the region-growing algorithm to segment the point cloud into different planes and then projected the points onto the plane to form a 2-D image. Finally, the contour feature points of the

2-D image were extracted by the binary image (BI) method. Bazazian et al. [29] segmented the point cloud into different regions based on the shortest distance between two points on the ground. For different regions, multiscale operators were used to determine the best scale of boundary point extraction, so as to improve the accuracy of edge detection. The advantage of this method is that it can extract boundary points with adaptive thresholds. The defect is that it is not robust to the point cloud noises. Wang et al. [30] first segmented the point cloud into different subregions by the social particle swarm optimization (SPSO)-fuzzy C-means clustering algorithm. Then, the angle criterion was used to detect the edges of the segmented subregions. Ultimately, they used the mean curvature to detect sharp feature points. This method is less affected by noise and has a good effect on the feature extraction of nonuniform sampling point clouds. Chen et al. [31] used vector clustering to extract the contour feature points. This type of method can only be applied to the extraction of crease points in the plane region or relatively simple surface region.

3) *Other Methods*: Manno-Kovacs and Sziranyi [32] determined the main direction of the building contour according to the local gradient and used the direction information to extract the feature points of the building edge. Awrangjeb [33] detected building features by fusing light detection and ranging (LiDAR) point clouds and aerial images, and obtained building contour points through regularization and mathematical morphology. The Hough transform is a well-known feature line extraction method, which was used to extract shape (line, circle, and so on) boundaries [34]. Many scholars used the Hough transform to extract object features. For example, Widyaningrum et al. [35] used ordered points to construct an extended Hough transform method and used this method to conduct the detection and segmentation of building boundary points. Widyaningrum et al. [36] used point order information and Hough transform to construct a novel method to extract the building contour of a point cloud. The disadvantage of this type of method is that it is easy to produce wrong corner points; especially for complex buildings, it is easy to detect wrong boundary points. Medial axis transform (MAT) can be used to describe the geometry and topology of the object. Based on this theory, Widyaningrum et al. [37] roughly estimated the 2-D MAT and used the α -shape algorithm to determine the boundary points of buildings. Kustraa et al. [38] performed feature conversion on medial points of complex objects and used the clustering method to classify the medial point cloud. For the classification point cloud, the skeleton features of the point cloud can be detected by regularization.

III. CONSTRUCTION OF PROJECTION PLANE OF NEIGHBORING POINTS

We set the neighboring points of point p_i to be $p_j = (x_j, y_j, z_j)$, $j = (1, 2, \dots, k)$. The local covariance matrix of point p_i can be expressed as

$$M = \frac{1}{k} \sum_{j=1}^k (p_j - \vec{p}) \cdot (p_j - \vec{p})^T \quad (1)$$

where $\vec{p} = (1/k) \sum_{j=1}^k p_j$.

Since the matrix M is a symmetric positive definite matrix, its eigenvalues are nonnegative. We perform the Cholesky decomposition of the covariance matrix. The matrix can be expressed as

$$M = [e_1 \ e_2 \ e_3] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix} \quad (2)$$

where λ_1 , λ_2 , and λ_3 are the eigenvalues of M ; e_1 , e_2 , and e_3 are the eigenvectors

The eigenvalues are all nonnegative and sorted by size. Therefore, $\lambda_1\lambda_2\lambda_3 > 0$. λ_1 , λ_2 , and λ_3 represent the fitting deviation in the direction of the corresponding eigenvector of the local plane. The minimum eigenvalue λ_3 corresponds to the direction deviation in the normal direction. The eigenvector e_3 corresponding to λ_3 is the local normal vector of one point. The local plane is constructed according to the local normal vector and center point \vec{p} of the neighboring points, as described in the following equation:

$$ax + by + cz + d = 0 \quad (3)$$

where (a, b, c) is the eigenvector e_3 .

IV. JUDGMENT OF BOUNDARY POINTS

A. Judgment of the Optimal Number of Neighboring Points

For the three eigenvalues λ_1 , λ_2 , and λ_3 , let $\delta_1 = ()^{1/2}\lambda_1$, $\delta_2 = ()^{1/2}\lambda_2$, and $\delta_3 = ()^{1/2}\lambda_3$. δ_1 , δ_2 , and δ_3 represent the fitting residuals in the three directions, respectively. We use the fitting residuals to construct the probability that the point belongs to the geometric feature

$$a_1 = \frac{\delta_1 - \delta_2}{\delta_1}, a_2 = \frac{\delta_2 - \delta_3}{\delta_1}, a_3 = \frac{\delta_3}{\delta_1} \quad (4)$$

where a_1 , a_2 , and a_3 , respectively, represent the probability that a point belongs to the linear points, planar points, and scattered points.

According to the Shannon entropy, the information contained in the neighborhood of each point is defined by

$$E_P = -a_1 \ln(a_1) - a_2 \ln(a_2) - a_3 \ln(a_3). \quad (5)$$

The smaller E_P means that the neighborhood of the point contains less information, and the more single the geometric feature of the point is. The different numbers of neighboring points correspond to different values of E_P . Therefore, the change of E_P can be used to determine the optimal number of neighboring points, as described in the following equation:

$$\begin{cases} k_{\text{opt}} = \operatorname{argmin}_{k_i} E_P(k_i) \\ k_i \in [k_{\text{min}}, k_{\text{max}}] \end{cases} \quad (6)$$

where k_{opt} is the optimal number of neighboring points and $[k_{\text{min}}, k_{\text{max}}]$ is the minimum and maximum number of neighboring points.

According to (6), the optimal number of neighboring points of each point can be determined.

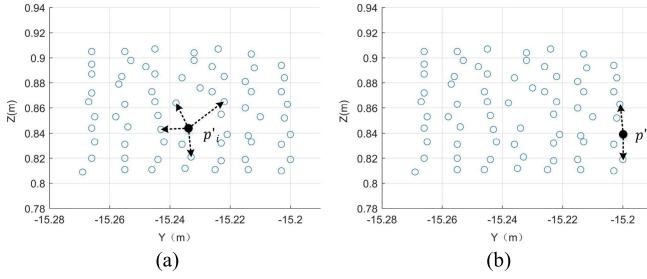


Fig. 1. Vectors between projection point p'_i and neighboring points. (a) Nonboundary points. (b) Boundary points.

B. Fine Extraction of Boundary Points

For the judgment of boundary points, we use the KNN method [16] to find the neighbors of each point, and the optimal number of neighboring points can be determined according to Section IV-A. The projection of point p_i and its neighboring point $p_j = (x_j, y_j, z_j)$ on the projection plane is $p'_j = (x'_j, y'_j, z'_j)$, as shown in the following equation:

$$\begin{cases} ax'_j + by'_j + cz'_j + d = 0 \\ \frac{x'_j - x_j}{a} = \frac{y'_j - y_j}{b} = \frac{z'_j - z_j}{c} = \lambda. \end{cases} \quad (7)$$

According to (7), λ is obtained by

$$\lambda = \frac{-ax_j - by_j - cz_j - d}{a^2 + b^2 + c^2}. \quad (8)$$

By substituting λ into (7), the coordinates of the projection points of neighboring points on the projection plane can be obtained by

$$\begin{cases} x'_j = a\lambda + x_j \\ y'_j = b\lambda + y_j \\ z'_j = c\lambda + z_j. \end{cases} \quad (9)$$

The vectors $\overrightarrow{p'_i p'_j}$ ($i = 1, 2, \dots, n; j = 1, 2, \dots, k$) between projection point p'_i and neighboring points can be obtained, as shown in Fig. 1.

We calculate the angles between adjacent vectors, as described in the following equation:

$$\alpha_j = \arccos \left(\frac{\overrightarrow{p'_i p'_1} \cdot \overrightarrow{p'_i p'_j}}{\|\overrightarrow{p'_i p'_1}\| \|\overrightarrow{p'_i p'_j}\|} \right) j = 1, 2, \dots, k. \quad (10)$$

According to the included angle α_j , we obtain the azimuth \tilde{a}_i of each vector, and the difference \tilde{a}_j between adjacent azimuth is obtained.

Fig. 1(b) shows that the border points are recognized as points for which \tilde{a}_j has a large value. Assume that the threshold of angles \tilde{a}_j is α_{thr} .

If $\tilde{a}_j > \alpha_{thr}$, the projection point p'_i is the boundary point.

If $\tilde{a}_j < \alpha_{thr}$, the projection point p'_i is not the boundary point.

We can use the above method to extract most of the boundary points, but the inconspicuous boundary points cannot be accurately extracted, as shown in Fig. 2(a). The reason for this phenomenon is that the neighboring points of object point p'_i exist on the upper and lower sides of the boundary

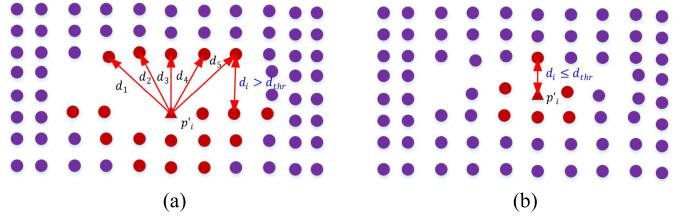


Fig. 2. Inconspicuous boundary points. (a) $d_i > d_{thr}$. (b) $d_i \leq d_{thr}$.

points, resulting in no larger \tilde{a}_j to judge the boundary points. In view of this, we combine the azimuth \tilde{a}_i and distance d_i of neighboring points into a new matrix $A = [\tilde{a}_i d_i]$, and the azimuth \tilde{a}_i is sorted from the smallest to largest, as shown in the following equation:

$$A = [\tilde{a}_i d_i] = \begin{bmatrix} \tilde{a}_1 & d_1 \\ \tilde{a}_2 & d_2 \\ \vdots & \vdots \\ \tilde{a}_k & d_k \end{bmatrix} \quad \tilde{a}_1 < \tilde{a}_2 < \dots < \tilde{a}_k \quad (11)$$

where \tilde{a}_i is the azimuth of each vector and d_i is the distance between the object point and neighboring points.

The maximum range \tilde{a}_{kl} of azimuth corresponding to the distance d_i that is continuously greater than the threshold value d_{thr} is determined. According to the three-sigma rule of thumb, the threshold value d_{thr} is obtained by [39], [40]

$$d_{thr} = R + 3\sigma_P \quad (12)$$

where R is the range resolution and σ_P is the point accuracy.

From Fig. 2(a), it is clearly visible that the distances d_1, d_2, d_3, d_4 , and d_5 are continuously greater than the threshold value d_{thr} .

If $\tilde{a}_{kl} < \alpha_{thr}$, the projection point p'_i is not the inconspicuous boundary points.

If $\tilde{a}_{kl} > \alpha_{thr}$, the projection point p'_i is the inconspicuous boundary points.

For a small number of inconspicuous boundary points, they cannot be extracted when the distance between different boundaries is less than the threshold d_{thr} , as shown in Fig. 2(b).

V. DETECTION OF CREEP POINTS

A. Determination of Bounding Ball

The accurate determination of the neighboring points is the basis of judging the crease points. The KNN method is a common algorithm to determine the neighboring points. For the crease points, the disadvantage of this method is that the same value of k produces different regions, as shown in Fig. 3. Fig. 3(a) shows that the neighboring points are concentrated on the left surface of the crease. Fig. 3(b) shows that the neighboring points are distributed on four surfaces with three creases. In view of this, we use the bounding sphere to determine the neighboring points.

According to the radius r of the bounding sphere, the points p_s , $s = 1, 2, \dots, m$, in the bounding sphere of each point are determined. The radius of the bounding sphere should neither be too large nor too small. If the radius is too large, the calculation amount will increase. Too small

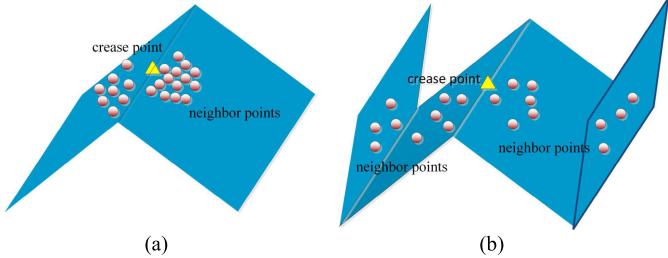


Fig. 3. Neighbor points of crease points. (a) Neighbor points are mainly on the right surface. (b) Neighbor points are on the four surfaces.

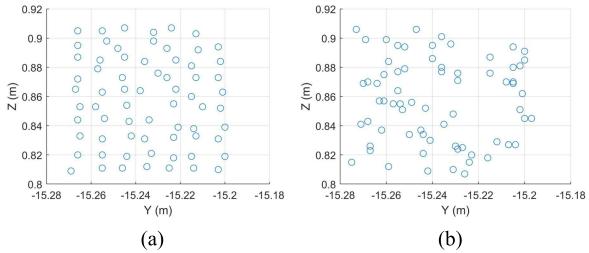


Fig. 4. Two different types of point clouds. (a) Homogeneous distribution. (b) Heterogeneous distribution.

will result in too few points in the bounding sphere, thus influencing the accuracy of the crease point extraction. There are two types of point clouds: one type of point cloud is uniform, as shown in Fig. 4(a), and another is unevenness, as shown in Fig. 4(b).

For the different types of point clouds, we give a novel method to determine the initial radius of the bounding sphere. First, the vectors $\overrightarrow{p_i p_s}$ ($i = 1, 2, \dots, n; s \in i, s \neq i$) between each point and all remaining points are obtained, and the distances d_{is} ($i = 1, 2, \dots, n; s \in i, s \neq i$) between each point and all remaining points are calculated. Simultaneously, the vectors $\overrightarrow{p_i p_s}$ are sorted according to the order of distances d_{is} from the smallest to the largest, and the sorted vectors $\overrightarrow{p_i p_s}$ are obtained. Then, the vectors $\overrightarrow{p_i p_s}$ are projected onto the local plane, as described in the following equation:

$$\overrightarrow{p'_i p'_s} = \overrightarrow{p_i p_s} - e_3 * (\overrightarrow{p_i p_s} \cdot e_3). \quad (13)$$

We set $\overrightarrow{p'_i p'_1}$ as the starting vector. The angles θ between the starting vector $\overrightarrow{p'_i p'_1}$ and other vectors $\overrightarrow{p'_i p'_s}, s \neq 1$, are calculated, as shown in the following equation:

$$\theta = \arccos \left(\frac{\overrightarrow{p'_i p'_1} \cdot \overrightarrow{p'_i p'_s}}{\|\overrightarrow{p'_i p'_1}\| \|\overrightarrow{p'_i p'_s}\|} \right). \quad (14)$$

Fig. 5(b) shows that the starting vector $\overrightarrow{p'_i p'_1}$ is on the left-hand side. In order to guarantee that there are points all around, there should be a vector on the right-hand side. Therefore, $\theta \cong 180$. The nearest distance d_{is} corresponding to $\theta \cong 180$ is the initial radius of the bounding sphere of each point. When the point p_i is on the boundary point, there is no vector on the right-hand side, as shown in Fig. 5(a). Therefore, the initial radius of the bounding sphere corresponding to the boundary points should be eliminated.

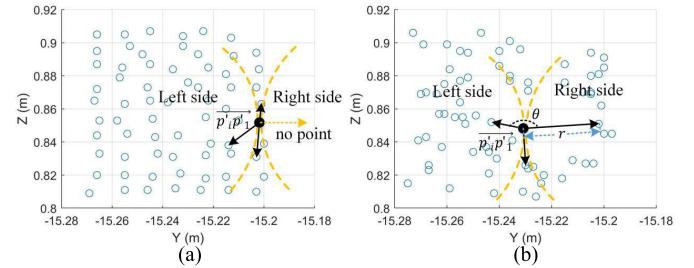


Fig. 5. Bounding sphere of point p'_i : (a) point p'_i is on the boundary and (b) point p'_i is not on the boundary.

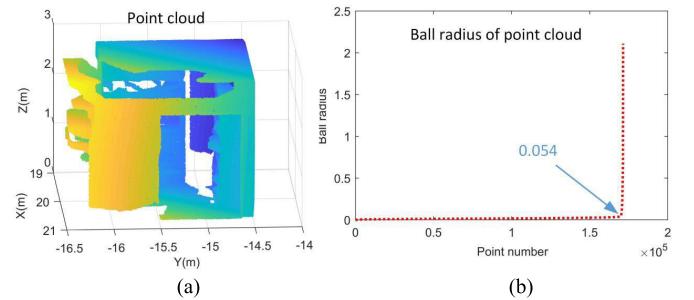


Fig. 6. Ball radius of all points in a point cloud. (a) Point cloud. (b) Ball radiiuses of 171 369 points.

We use the above method to determine the initial radius of the bounding ball of all points of a building point cloud, as shown in Fig. 6.

Fig. 6 shows the radiiuses of the bounding ball for all points of one room point cloud. We sort these radiiuses and construct the 2-D discrete data \mathbf{R} , as described in the following equation:

$$\mathbf{R} = [L_i \quad r_i] = \begin{bmatrix} 1 & r_1 \\ 2 & r_2 \\ \vdots & \vdots \\ n & r_n \end{bmatrix}. \quad (15)$$

The 2-D discrete data \mathbf{R} are normalized to $[-1]$ and then obtain the normalized discrete data $\vec{\mathbf{R}}$

$$\vec{\mathbf{R}} = [\vec{L}_i \quad \vec{r}_i] = \begin{bmatrix} -1 & -1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix}. \quad (16)$$

For the normalized discrete data $\vec{\mathbf{R}}$, the first data \vec{R}_1 and last data \vec{R}_n are extracted, and the other remaining data are \vec{R}_i ($i = 2, 3, \dots, n - 1$). According to the first and last data, the vectors of each point of the other remaining data are constructed, as described in the following equation:

$$\begin{cases} V_{1i} = \vec{R}_1 - \vec{R}_i, \\ V_{2i} = \vec{R}_n - \vec{R}_i, \end{cases} \quad i = 2, 3, \dots, n - 1. \quad (17)$$

According to (17), the angles between the vector V_{1i} and V_{2i} are calculated

$$\theta_i = \frac{V_{1i} \cdot V_{2i}}{\|V_{1i}\| \|V_{2i}\|}, \quad i = 2, 3, \dots, n - 1. \quad (18)$$

According to (18), the inflection point No in Fig. 6(b) is obtained, as shown in the following equation:

$$\text{No} = i + 1, \quad i \in \arg\min(\theta_i). \quad (19)$$

According to (19), the radius of the bounding sphere is determined as r_{No} . Thus, the inflection point is 0.054. The radius of the bounding sphere of almost all points is less than 0.054 m. Only a few points have a radius higher than 0.054 m, and some even exceed 1 m. The reason for this phenomenon is that these few points are on the boundary. Finally, the initial radius of the bounding ball in the building point cloud is determined to be 0.054.

B. Projection of Point Cloud in the Bounding Sphere

According to the bounding ball obtained in Section V-A, the point cloud $p_s = (x_s, y_s, z_s)$ within the bounding ball of each point p_i is determined. Then, we use (1) to calculate the covariance matrix of domain points of point p_i . According to (2), the Cholesky decomposition of the covariance matrix is carried out to obtain the three eigenvalues λ_1, λ_2 , and λ_3 and corresponding eigenvectors e_1, e_2 , and e_3 . The eigenvector e_1 corresponding to the largest eigenvalue λ_1 is perpendicular to the local normal vector e_3 of this point. Therefore, we can construct the local projection plane F according to the feature vector e_1 and the center point \bar{p} of the point cloud within the bounding sphere. This plane is perpendicular to the fitting plane of the point cloud within the bounding sphere, as described in the following equation:

$$\tilde{a}x + \tilde{b}y + \tilde{c}z + d = 0 \quad (20)$$

where $(\tilde{a}, \tilde{b}, \tilde{c})$ is eigenvector e_1 and $d = -e_1 \bar{p}^T$.

According to (7)–(9), we project the point cloud p_s inside the bounding sphere onto the plane F. The projection points on the plane F are $p'_s = (x'_s, y'_s, z'_s)$. In order to analyze the features of the point cloud in the bounding sphere, the 3-D projection points need to be converted to the 2-D coordinate system. According to the eigenvector e_1 and the unit vector $e_z = (0, 0, 1)$ of the z-axis, the cross product of these two vectors is calculated, as described in the following equation:

$$u = e_z \times e_1. \quad (21)$$

We conduct the unitization of u

$$e_u = \frac{u}{|u|}. \quad (22)$$

According to the quaternion transforms, the rotation angle is calculated by

$$\varphi = \arccos\left(\frac{e_1 \cdot e_z}{|e_1||e_z|}\right)/2. \quad (23)$$

Meanwhile, the quaternion is obtained by

$$q = [\cos(\varphi) \sin(\varphi) * e_u]. \quad (24)$$

According to the quaternion, the rotation matrix is obtained, as described in (25), shown at the bottom of the page.

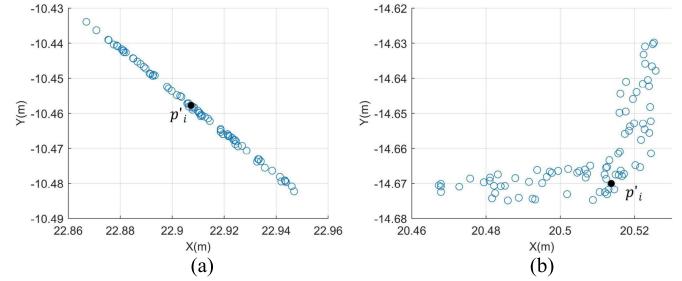


Fig. 7. Coordinates of crease and noncrease points after rotation in a 2-D coordinate system. (a) Noncrease points. (b) Crease points.

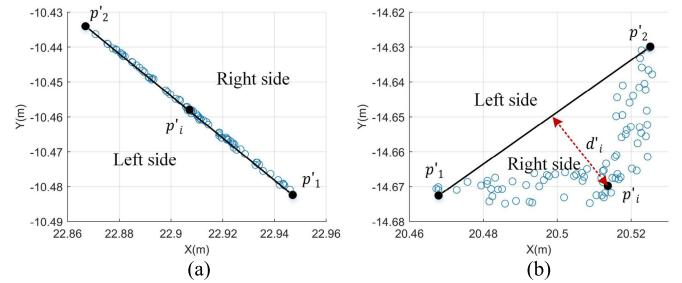


Fig. 8. Distance d'_i from two types of projection points p'_i to line $p'_1p'_2$. (a) Noncrease points. (b) Crease points.

According to (25), the coordinates of the projection points after rotation are obtained as follows:

$$p'_R = p'_s * R. \quad (26)$$

According to (26), the coordinates of the projection of crease and noncrease points in the 2-D coordinate system are obtained, as shown in Fig. 7.

From Fig. 7(a), it is clearly visible that the noncrease points after rotation in a 2-D coordinate system have a straight-line shape. However, the crease points have the shape of a broken line, as shown in Fig. 7(b). Based on this observation, we can develop a novel method for extracting the crease point cloud.

C. Extraction of Crease Points

For the point cloud $p'_R = (x'_R, y'_R)$ after projection, we determine the two farthest points p'_1 and p'_2 . According to the coordinates of these two points $p'_1(x'_1, y'_1)$ and $p'_2(x'_2, y'_2)$, the straight line equation between two points is set, as described in the following equation:

$$y' = ax' + b. \quad (27)$$

According to the coordinate of point p'_i and (27), the distance d'_i from point $p'_i(x'_i, y'_i)$ to line $p'_1p'_2$ can be calculated by

$$d'_i = \frac{|ax' + b - y'|}{\sqrt{a^2 + 1}}. \quad (28)$$

According to (28), we can get the distance d'_i from projection points p'_i to line $p'_1p'_2$. From Fig. 8(a), it is clearly

$$R = \begin{bmatrix} 2q(1)^2 - 1 + 2q(2)^2 & 2(q(2)q(3) + q(1)q(4)) & 2(q(2)q(4) - q(1)q(3)) \\ 2(q(2)q(3) - q(1)q(4)) & 2q(1)^2 - 1 + 2q(3)^2 & 2(q(3)q(4) + q(1)q(2)) \\ 2(q(2)q(4) + q(1)q(3)) & 2(q(3)q(4) - q(1)q(2)) & 2q(1)^2 - 1 + 2q(4)^2 \end{bmatrix} \quad (25)$$

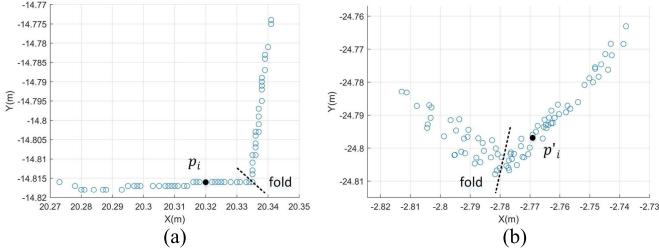


Fig. 9. Point p_i and point cloud inside the bounding sphere of point p_i before and after projection. (a) Point p_i and crease point cloud before projection. (b) Point p_i and crease point cloud after projection.

visible that the distance d'_i is very small when the point cloud in the bounding sphere is noncrease points. Conversely, the distance d'_i is larger when they are crease points. Therefore, we can obtain the first constraint for judging the crease points according to the standard deviation of the distance d'_i , as described in the following equation:

$$C_{f1} = \sqrt{\frac{\sum_{i=1}^R (d'_i - \bar{d}'_i)^2}{n-1}}. \quad (29)$$

From Fig. 8(b), it is obvious that the number of points on the two sides of line $p'_1 p'_2$ is very different. Therefore, we can construct the second constraint parameter for judging the crease points from the gap between the number of points on both sides of line $p'_1 p'_2$, as described in the following equation:

$$C_{f2} = \frac{n_R}{n_L}, \quad n_R > n_L \quad (30)$$

where n_R is the number of points on the right-hand side and n_L is the number of points on the left-hand side.

According to (29) and (30), the basic judgment model for extracting crease points is obtained, as shown in the following equation:

$$\begin{cases} C_{f1} \geq C'_{f1} \\ C_{f2} \geq C'_{f2} \end{cases} \quad (31)$$

According to (31), it can be judged whether the point cloud inside the bounding sphere of point p_i is crease points. However, there is no guarantee that the point p_i is the crease point, as shown in Fig. 9.

Fig. 9 shows that the point cloud p_s inside the bounding sphere of point p_i is crease point cloud. However, the point p_i is the noncrease point. From Fig. 9(b), it is obvious that crease points are recognized as points for which the value of d'_i is large. d'_i is sorted from the largest to the smallest. The points p_i corresponding to the first few d'_i are crease points. Finally, the extraction model of crease points is given, as shown in the following equation:

$$\begin{cases} C_{f1} \geq C'_{f1} \\ C_{f2} \geq C'_{f2} \\ p_i \in p_m, d'_1 > d'_2 > \dots > d'_m. \end{cases} \quad (32)$$

Generally, $m = 3$, and the other two parameters C'_{f1} and C'_{f2} are nonfixed values.

TABLE I
OPTIMAL NUMBER OF NEIGHBORING POINTS

Different type of point clouds	Different points				
	1	2	3	...	n
Block	15	18	21	...	29
storage1	17	22	24	...	25
"3S"	22	19	27	...	24
Bridge	19	21	23	...	22
haiyun building	20	23	24	...	26

TABLE II
RADIISES OF THE BOUNDING SPHERE

Different type of point clouds				
Block	storage1	"3S"	Bridge	haiyun building
2.0	0.05	0.13	0.5	0.3

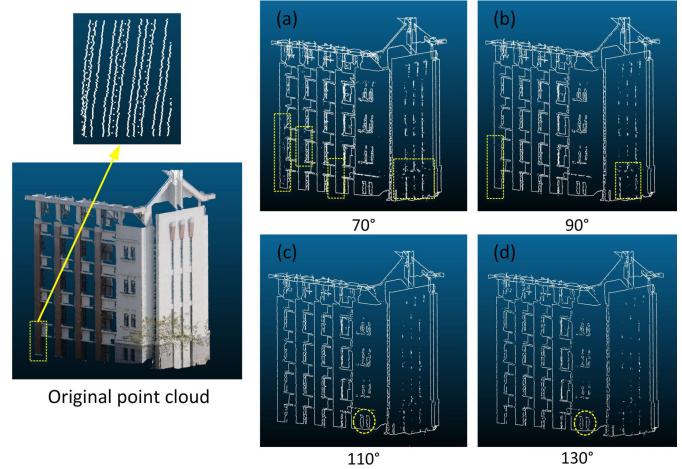


Fig. 10. Building boundary points with different values of α_{thr} .

VI. EXPERIMENTAL ANALYSIS

A. Determination of Nearest Neighbor

To extract boundary and crease points, it is necessary to determine the optimal number of neighboring points and the radius of the bounding sphere. First, the optimal number of neighboring points of each point of different types of point clouds is determined according to Section IV-A, as shown in Table I.

Then, the radiiuses of the bounding sphere of different types of point clouds are determined according to Section V-A, as shown in Table II.

B. Analysis of Hyperparameters

1) Parameter of Boundary Point Extraction: In order to determine α_{thr} , the boundary points of one building [6] are extracted according to different α_{thr} . We set α_{thr} to be 70° , 90° , 110° , and 130° , and the boundary points are obtained by the proposed method, as shown in Fig. 10.

From the dotted rectangle of Fig. 10(a), it is clearly visible that partial wall areas contain many redundant points when α_{thr} is 70° . The main reason for this phenomenon is that the point clouds on these wall areas are relatively sparse, as shown in the dotted rectangle of the original point clouds in Fig. 10. As α_{thr} increases to 90° , the redundant points in these wall areas are reduced a lot but still contain a small number of points, as shown in the dotted rectangle of Fig. 10(b). As α_{thr} increases to 110° , not only the boundary points are clearly retained

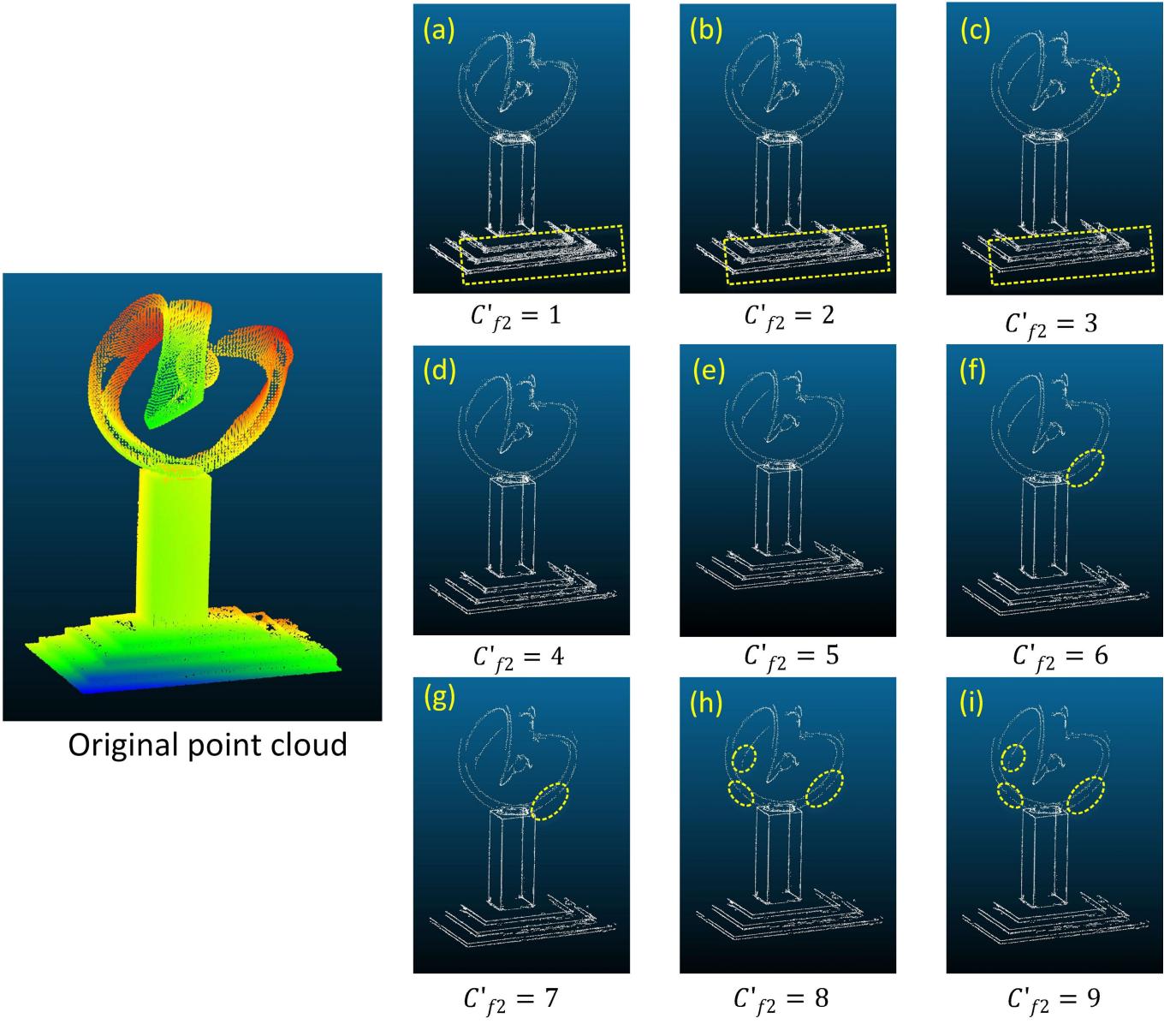


Fig. 11. Influence of different values of C'_{f2} on the extraction of crease points of the “3S” building. (a) Extraction of crease points when parameter $C'_{f2} = 1$. (b) Extraction of crease points when parameter $C'_{f2} = 2$. (c) Extraction of crease points when parameter $C'_{f2} = 3$. (d) Extraction of crease points when parameter $C'_{f2} = 4$. (e) Extraction of crease points when parameter $C'_{f2} = 5$. (f) Extraction of crease points when parameter $C'_{f2} = 6$. (g) Extraction of crease points when parameter $C'_{f2} = 7$. (h) Extraction of crease points when parameter $C'_{f2} = 8$. (i) Extraction of crease points when parameter $C'_{f2} = 9$.

but also there are almost no redundant points, as shown in Fig. 10(c). When the threshold α_{thr} is 130° , although most of the boundary points are clearly retained, the boundary points of some small areas cannot be retained, as shown in the dotted ellipse of Fig. 10(d). Ultimately, the threshold α_{thr} is determined to be approximately 110° . Generally, the parameter α_{thr} can be 110° .

2) *Parameter of Crease Point Extraction:* In order to determine the optimal values of the two parameters C'_{f1} and C'_{f2} , we analyze the influence of one of the parameters on the extraction of crease points when the other parameter is fixed. First, we analyze the influence of different values of C'_{f2} on the extraction of crease points. We set $C'_{f1} = 5$ mm and $m = 3$; C'_{f2} is the different value from 1 to 9 with the interval of 1. The proposed method is used to extract the crease points of the “3S” building point cloud, as shown in Fig. 11.

The stairway crease points at the bottom of the 3S building cannot be accurately extracted when C'_{f2} is small, and the lateral side of the stairway contains a large number of redundancy points, as shown in the dotted rectangle of Fig. 11(a) and (b). When C'_{f2} increased to 3, many points on the lateral sides of the stairway are eliminated, and the crease points of the stairway are clearly visible. However, the curved surface at the upper of the 3S building contains a few redundancy points, as shown in the dotted circle of Fig. 11(c). When $C'_{f2} = 4$, the crease points of the stairway can be accurately extracted, and the lateral sides of the stairway do not contain redundancy points. Meanwhile, the crease points in the upper curved surface of the “3S” building are more continuous. $C'_{f2} = 5$ is similar with $C'_{f2} = 4$; the crease points of “3S” building are accurately extracted; and the contour points of straight and curved lines on the lower upper part of the 3S building are

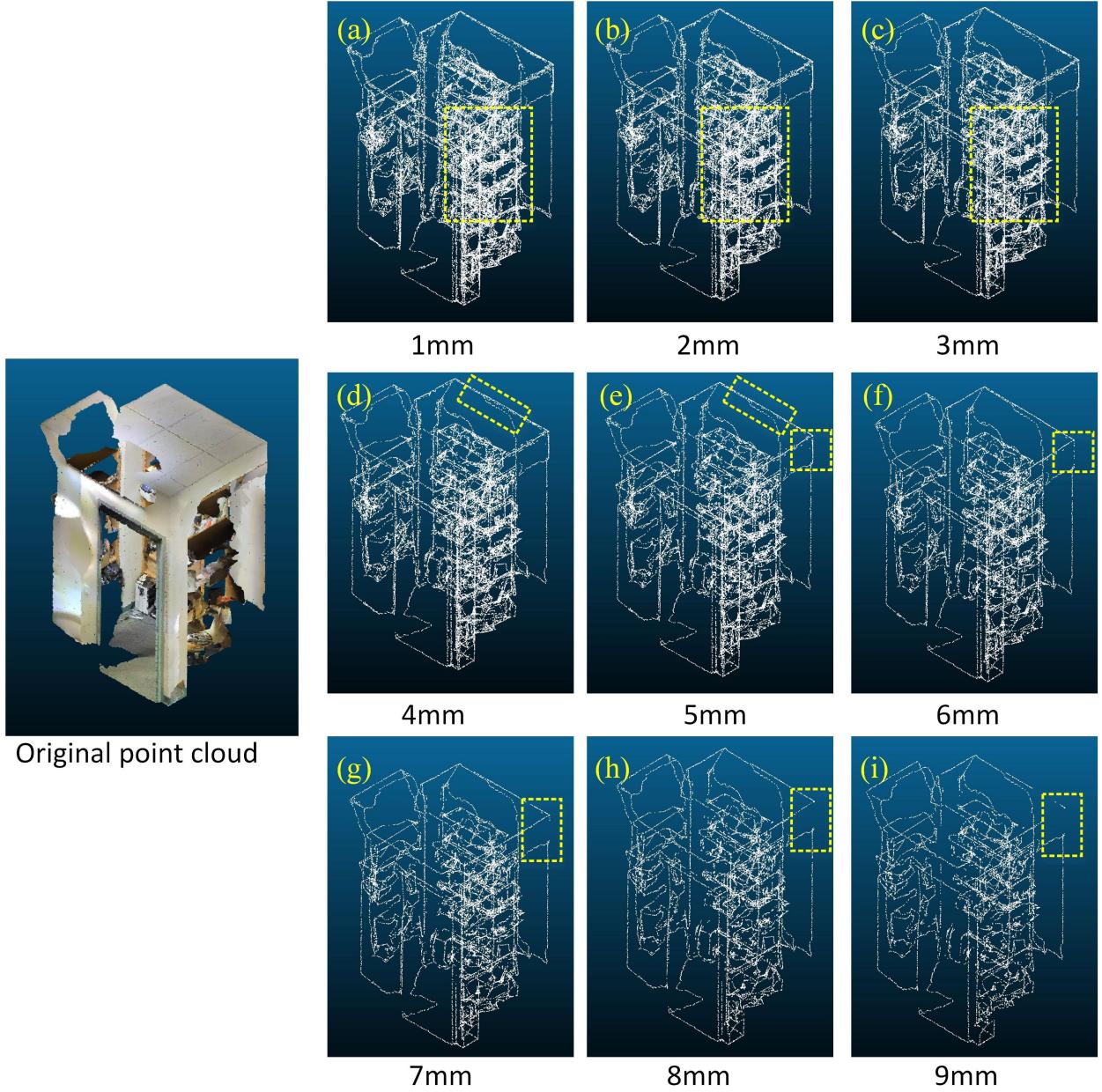


Fig. 12. Effect of different values of C'_{f1} on the extraction of crease points in a point cloud. (a) Extraction of crease points when parameter $C'_{f1} = 1$. (b) Extraction of crease points when parameter $C'_{f1} = 2$. (c) Extraction of crease points when parameter $C'_{f1} = 3$. (d) Extraction of crease points when parameter $C'_{f1} = 4$. (e) Extraction of crease points when parameter $C'_{f1} = 5$. (f) Extraction of crease points when parameter $C'_{f1} = 6$. (g) Extraction of crease points when parameter $C'_{f1} = 7$. (h) Extraction of crease points when parameter $C'_{f1} = 8$. (i) Extraction of crease points when parameter $C'_{f1} = 9$.

clearly visible. When $C'_{f2} = 6$, there are discontinuity points within the curved surface of the upper part of the 3S building, as shown in the ellipse of Fig. 11(f). With the increase in C'_{f2} , there are more and more discontinuity points within the upper curved surface of the “3S” building, as shown in the ellipse of Fig. 11(g)–(i). Therefore, $C'_{f2} = 4$ or 5 is regarded to be the optimal threshold for extracting crease points of a “3S” building. For different types of point clouds, the parameter $C'_{f2} = 4$ or 5 is useful.

Second, we set $C'_{f2} = 4$ and $m = 3$; C'_{f1} is the different value from 1 to 9 mm with an interval of 1mm. We use the proposed method to extract the building crease points, and the range resolution is 4 mm, as shown in Fig. 12.

Fig. 12 shows that the smaller the C'_{f1} value is, the more redundancy points are contained in the extracted crease points. For example, the crease points at $C'_{f1} = 1$ contain more redundancy points than $C'_{f1} = 3$. Especially, the internal contour of the building is not clear when C'_{f1} is small, as shown in the rectangle of Fig. 12(a)–(c). The reason for this phenomenon is that some noncrease points are recognized as crease points when C'_{f1} is relatively small. According to the comparison of Fig. 12(d) and (e), the crease points of $C'_{f1} = 4$ contain more redundancy points than $C'_{f1} = 5$ in the part of contours, as shown in the rectangle of Fig. 12(d) and (e). Simultaneously, a few crease points at $C'_{f1} = 5$ are more continuous than those at $C'_{f1} = 6$, as shown in the square of Fig. 12(e) and (f). As C'_{f1} increases, more crease points

TABLE III
OPTIMUM VALUE OF PARAMETER C'_{f1} AT DIFFERENT RANGE RESOLUTIONS OF POINT CLOUD

Different point cloud	C'_{f1}	Range resolution
Block	0.3	0.24
Block	0.22	0.25
Block	0.4	0.454
storage1	0.005	0.005
“3S”	0.018	0.022
Bridge	0.03	0.041
haiyun_building	0.012	0.026

cannot be accurately extracted, as shown in the rectangle of Fig. 12(g)–(i). Therefore, the optimal value of parameter C'_{f1} is determined as 5 mm.

However, we find that the optimum value of parameter C'_{f1} is not fixed for the point clouds with different range resolutions. In order to determine the optimum value of parameter C'_{f1} , we use the proposed method to conduct the extraction of crease points at different range resolutions. Therefore, we obtain the optimum value of the corresponding parameter C'_{f1} at different range resolutions, as shown in Table III.

Table III shows that different range resolutions correspond to different optimal values of parameter C'_{f1} . For the block point cloud, there are three different parameters C'_{f1} because of the three different range resolutions. For the “3S” and haiyun_building two different types of point clouds, they have similar parameters C'_{f1} because of the similar range resolution. Therefore, the different types of object point clouds have no influence on the optimal value of parameter C'_{f1} . The parameter C'_{f1} is mainly affected by the range resolution, and its value is approximately equal to the range resolution. Ultimately, the optimum value of parameter C'_{f1} can be obtained near the range resolution.

C. Comparative Studies

1) Extraction of Boundary Points: According to Table I, the optimal number of nearest neighbor points of each point is obtained for the extraction of boundary points. We use the proposed method to conduct the extraction of building boundary points and compare it with the method without considering the inconspicuous boundary points, as shown in Fig. 13.

According to the comparison of Fig. 13(a) and (b), it is clearly visible that the proposed method retains more boundary points than the method without considering the inconspicuous boundary points. The dashed rectangle of Fig. 13(a) shows that some boundary points of the window are not retained. The reason for this phenomenon is that the method does not take into account the inconspicuous points. According to (11), the proposed method considers the inconspicuous boundary points, and it preserves almost all boundary points of the window, as shown in the dashed rectangle of Fig. 13(a).

In addition, the proposed method is compared with the ordered point-aided Hough transform (OAHT) method [35], the facet segmentation-based (FSB) method [6], and the BI method [28], as shown in Fig. 14.

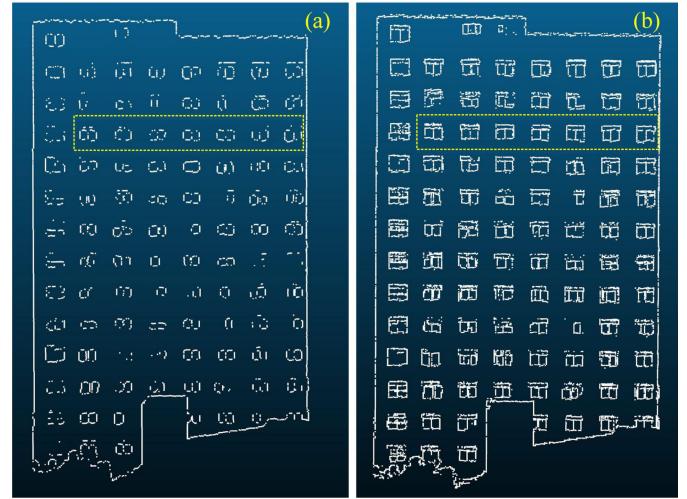


Fig. 13. Extraction results of boundary points. (a) Method without considering the inconspicuous boundary points. (b) Proposed method with consider conspicuous boundary points.

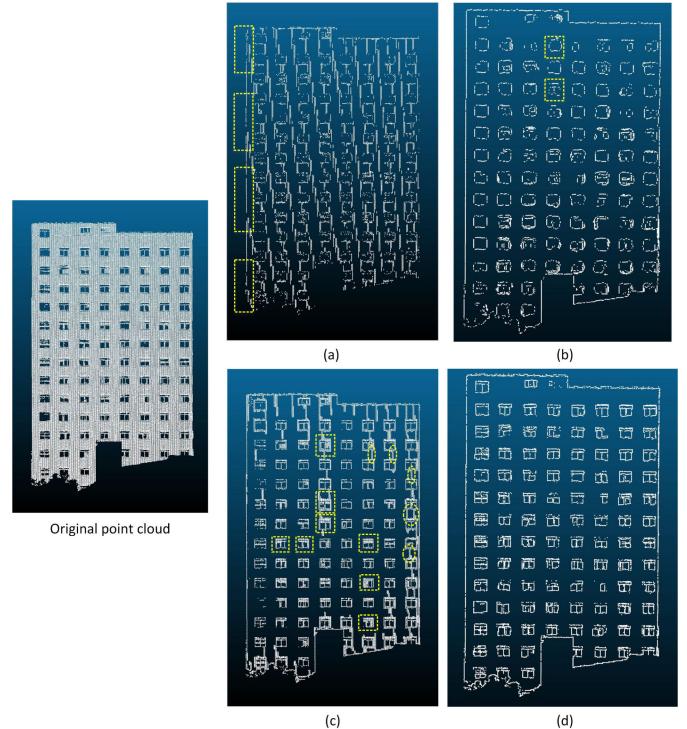


Fig. 14. Extraction of building boundary points by four different methods. (a) OAHT method [35]. (b) FSB method [6]. (c) BI method [28]. (d) Proposed method.

From Fig. 14, it is visible that the OAHT method does not clearly extract the wall boundary points. Although the boundary points of each window are obtained by the OAHT method, the extracted window boundary points are quite different from the real ones. There are many breakage boundary points on the right side of the entire wall, as shown in the rectangle of Fig. 14(a). Especially, the top, bottom, and right edge points of the entire wall are not extracted successfully. The reason for this phenomenon is that the Hough transform cannot accurately extract the boundary of the low-resolution image transformed by a point cloud. Fig. 14(b) shows that the

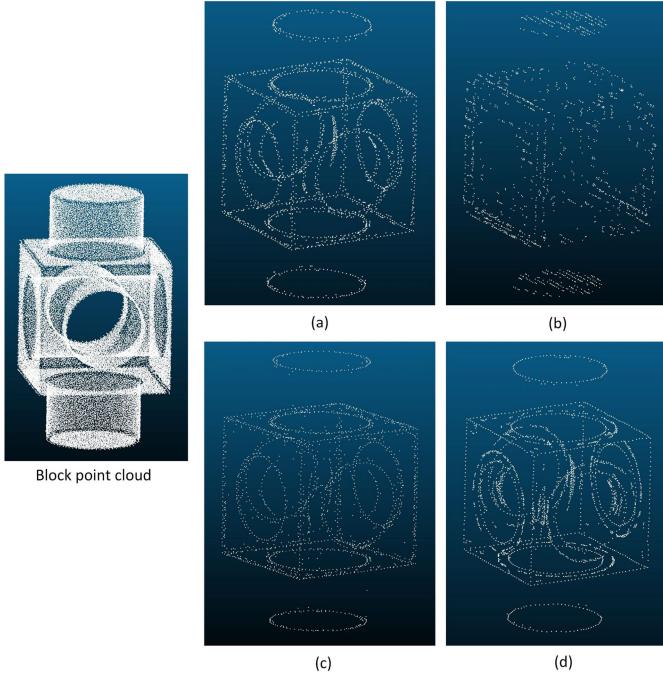


Fig. 15. Extraction of crease points by four different methods. (a) Region clustering curvature method [30]. (b) BI method. (c) Chen method. (d) Proposed method.

FSB method obtains a good result in the extraction of contour points. However, the window cross-beam points cannot be accurately extracted, as shown in the rectangle of Fig. 14(b). The possible reason is that this method cannot accurately obtain the small plane of the window cross-beam. The BI method not only extracts the four edges of the entire wall but also obtains most of the window outline. However, some window outlines contain a large number of nonboundary points, as shown in the rectangular of Fig. 14(c). Simultaneously, a large number of wall points between the upper and lower windows are misrecognized as boundary points, as shown in the ellipse of Fig. 14(c). It can be clearly seen from Fig. 14(d) that the proposed method not only successfully extracts the boundary points of the entire wall but also extracts the boundary points of windows and window cross-beams. Therefore, the extraction performance of boundary points by the proposed method is superior to the other three methods.

2) *Extraction of Crease Points:* According to the optimal radius of the bounding sphere provided in Table II, we use the proposed method to conduct the extraction of crease points. The extraction results by the proposed method are compared with the region clustering curvature method [30], the BI method [28], and the Chen [31] method, as shown in Fig. 15.

Fig. 15(a) shows that the crease points extracted by the region clustering curvature method can display the outline of the block point cloud. However, the crease points contain a large number of redundancy points, resulting in a rough appearance of the contour. By the same token, the contour feature points extracted by the Chen method are rough, as shown in Fig. 15(c). The reason for this phenomenon is that these two methods primarily rely on curvature and normal vector clustering to extract contour feature points. Nevertheless, the

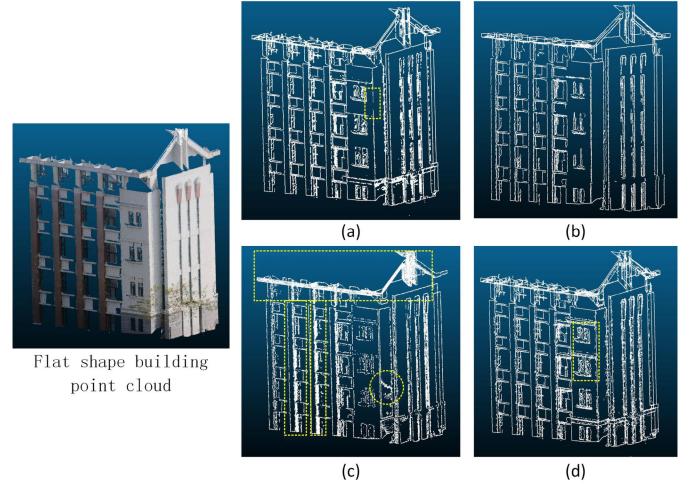


Fig. 16. Extraction of contour feature points of flat shape building point cloud by four different methods. (a) Region clustering curvature method. (b) FSB method. (c) OAHT method. (d) Proposed method.

curvature or normal vector clustering of the points on the edge of the contour is similar to the points on the contour. The contour points of the block point cloud extracted by the BI method are chaotic, as shown in Fig. 15(b). It illustrates that the BI method fails to extract the crease points. The proposed method can extract most crease points of the circular cavity. Simultaneously, the crease points of the cube and cylinder are successfully extracted by the proposed method, as shown in Fig. 15(d). Therefore, the performance of the proposed method is better than the other three methods in terms of the extraction of crease points.

3) *Extraction of Contour Feature Points:* The contour feature points mainly include boundary and crease points. We use the proposed method to extract boundary and crease points, and obtain contour feature points. In order to evaluate the performance of the proposed method, we use the flat shape building and curvilinear architecture point cloud to conduct the extraction of feature points and compare the results with the region clustering curvature method [30], the FSB method [6], and the OAHT method [35]. For the flat shape building, the contour feature points are extracted by the four methods, as shown in Fig. 16.

From Fig. 16, it is clearly visible that most of the contour feature points of the building have been extracted by these four methods. However, these four methods have different effects on the extraction of contour points. For example, the extraction results of the region clustering curvature method are superior to the FSB method. Fig. 16(a) shows that the extracted contour points of the building represent the whole building frame. However, the extracted partial feature points are discontinuous, as shown in the rectangle of Fig. 16(a). The FSB method obtains clear building contour feature points, as shown in Fig. 16(b), and its extraction effect is superior to that of region clustering curvature and OAHT methods. For the OAHT method, the contour feature points of the beams on the top of the building and the concealed column are not extracted, as shown in the rectangle of Fig. 16(c). In addition, the contour feature points extracted by the OAHT method contain

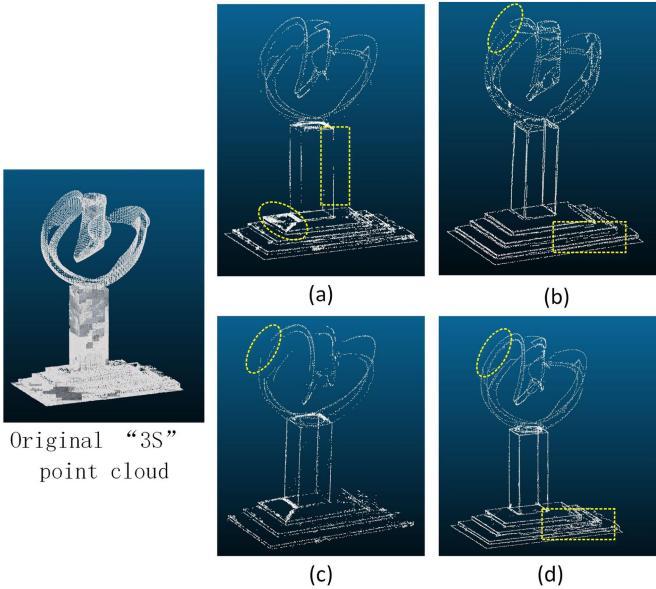


Fig. 17. Extraction of contour feature points of curvilinear architecture point cloud by four different methods. (a) Region clustering curvature method. (b) FSB method. (c) Chen method [35]. (d) Proposed method.

some redundant points, as shown in the circle of Fig. 16(c). The contour feature points extracted by the proposed method contain almost all contour points, and the outline of the building is clearly visible, as shown in Fig. 16(d). Although a few redundant points on the window are retained, the contour feature points of the window cross-beam are extracted by the proposed method. According to the comparison of these four methods, the performance of the proposed method is similar to the FSB method and better than the other two methods in terms of the extraction of feature points of flat shape building.

For the curvilinear architecture, the contour feature points of the “3S” building on the campus of Wuhan University are extracted by the region clustering curvature method, the FSB method, the Chen method, and the proposed method, as shown in Fig. 17.

From the rectangle of Fig. 17(a), it is visible that the contour points extracted by the region clustering curvature method are partially discontinuous. Furthermore, the contour points of the “3S” building extracted by the region clustering curvature method contain a large number of redundant points, as shown in the ellipse of Fig. 17(a). The reason for this result is that the curvature of some nonfeature points is similar to that of feature points. The contour points of the square prism and bottom base of the “3S” structure are successfully extracted by the FSB method, and the extraction effect is better than the other three methods, as shown in the rectangle of Fig. 17(b). However, some extra useless points are produced in the curve region after the extraction of crease points by the FSB method. Part edge of the surface is broken, as shown in the ellipse of Fig. 17(b). By the same token, some contour feature points of the surface in the upper of the “3S” structure cannot be kept by the Chen method, as shown in the ellipse of Fig. 17(c). Fig. 17(d) shows that the proposed method can accurately extract the contour feature points of the “3S” building. However, the extraction effect of the contour feature points of the bottom base of

the “3S” structure by the proposed method is slightly inferior to the FSB method, as shown in the rectangle of Fig. 17(d). The contour points of curvilinear architecture on the top of the “3S” building are successfully extracted by the proposed method, and it is superior to the FSB method, as shown in the ellipse of Fig. 17(d). In addition, the contour feature points do not contain redundant points. It illustrates that the proposed method is suitable for the contour feature point extraction of curvilinear architecture.

D. Robustness Studies

Fig. 18 shows that the extraction effect of crease points by the BI method is the worst, and the crease points extracted by this method with different noise point clouds are chaotic. From Fig. 18(c), it is clearly visible that the Chen method can accurately extract the contour feature points of the block point cloud when the noise is small. As Gaussian noise increases, the Chen method fails to extract the contour feature points. Fig. 18(d) shows that the region clustering curvature method can not only extract the crease points of the cube but also the crease points of the circular cavity in the middle part of the block. However, the circular cavity in the middle part of the block contains two creases, and this method can only extract a single crease of the circular cavity. In addition, with the increasing Gaussian noise, the extraction efficiency of the region clustering method on the crease points of the block becomes worse, and part of the noncrease points on the cylinder is retained, as shown in the ellipse of Fig. 18(d). Especially, it fails to extract crease points when the Gaussian noise increases to 0.2. It illustrates that the robustness of Chen and region clustering curvature methods is poor. For the block point clouds of different Gaussian noises, the proposed method not only extracts the crease points of the cube in the middle part of the block but also successfully extracts the two creases of the circular cavity in the middle part of the block. Simultaneously, the proposed method also successfully extracts the crease points of the two cylinders at the top and bottom of the block, as shown in Fig. 18(e). As the Gaussian noise increased to 0.2, although the crease points extracted by the proposed method contain a few redundant points, the overall contour of the block is obtained and clearly visible. From Fig. 18, it illustrates that the robustness of the proposed method is better than the other three methods.

E. Quantitative Comparison

In order to evaluate the performance of the proposed method, we conduct the quantitative comparison by means of three retrieval measurements: precision, recall, and F1-score [41], as shown in the following equations:

$$\text{Precision} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FP}|} \quad (33)$$

$$\text{Recall} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}|} \quad (34)$$

$$\text{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (35)$$



Fig. 18. Extraction of crease points by different methods in the presence of different levels of Gaussian noise. From (top) to (bottom): 0.03-, 0.05-, 0.1-, and 0.2-m Gaussian noises. (a) Block point cloud with Gaussian noise. (b) BI method. (c) Chen method. (d) Region clustering curvature. (e) Proposed method.

where TP represents true positive, which means that the true contour feature points are extracted successfully. FP represents false positive, which means that the extracted points are not true contour feature points. FN represents false negative, which means that the contour feature points cannot be extracted.

We use the original point cloud of the “Computer World” commercial building [31] next to Wuhan University to conduct the quantitative analysis, as shown in Fig. 19.

We extract the true contour feature points manually, and the number of the total true contour feature points is 4868.

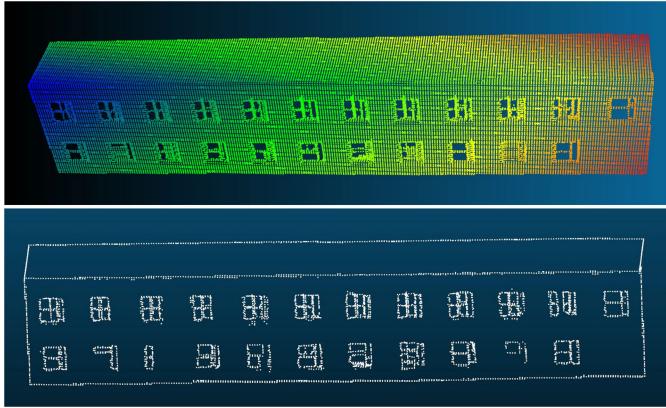


Fig. 19. True contour feature points of the “Computer World” building.

TABLE IV
TP, FP, AND FN OF THE FOUR METHODS

	Region clustering curvature	OAHT	Chen method	Proposed method
True contour feature points		4868		
Extracted contour feature points	4842	4269	5727	4975
TP	4298	3168	4588	4741
FP	544	1101	1139	234
FN	570	1700	280	127

TABLE V
PRECISION, RECALL, AND F1-SCORE OF THE FOUR METHODS

	Region clustering curvature	OAHT	Chen	Proposed method
Precision	0.89	0.74	0.80	0.95
Recall	0.88	0.65	0.94	0.97
F1-score	0.89	0.69	0.87	0.96

For the contour feature points extracted by different methods, a point is considered a TP if the distance between it and the true contour feature point is within the range resolution.

According to the optimal number of neighboring points provided in Table I and the radiiuses of the bounding sphere provided in Table IV, we use the proposed method to extract the contour feature points of the building and compare them with the region clustering curvature, OAHT, and Chen methods [31]. The TP, FP, and FN are obtained by these four methods, as shown in Table IV.

According to Table IV, the precision, recall, and F1-score of the four methods are obtained, as shown in Table V.

From Table V, it is clearly visible that the precision, recall, and F1-score of the OAHT method are all lower than the other three methods. The precision of the Chen method is slightly lower than the region clustering curvature and proposed methods. The reason for this phenomenon is that the precision is only related to TP and FP. Some points extracted by the Chen method are not true contour points. Nevertheless, the recall of the Chen method is similar to the proposed method and higher than the other two methods. The reason is that the recall is mainly related to FN, and only a few true contour points cannot be extracted by the Chen method. The F1-score of the region clustering curvature is similar to the Chen method and higher than the OAHT method. Table V shows that the F1-score of

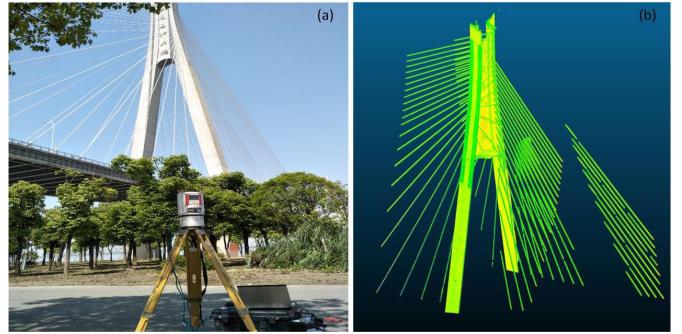


Fig. 20. Monitoring of the “Wuhan xx Yangtze River Bridge.” (a) Position of the RIEGL VZ400 scanner. (b) Bridge point cloud.

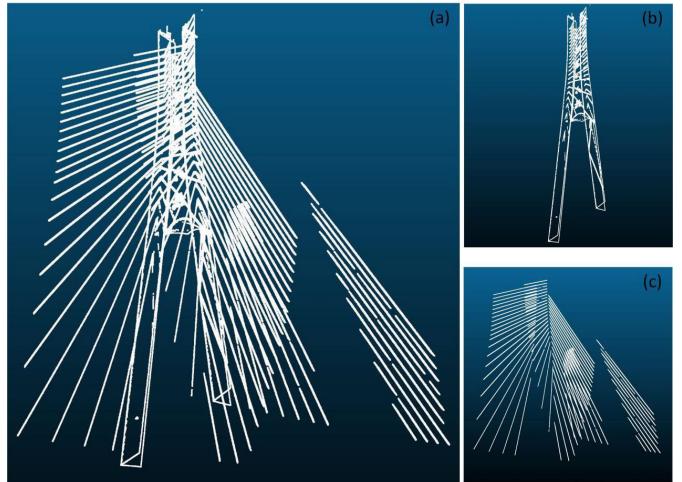


Fig. 21. Extraction of the “Wuhan xx Yangtze River Bridge” contour feature points. (a) Cable bent tower feature points. (b) Stay cable feature points.

the proposed method is higher than the other three methods. It illustrates that the proposed method can not only extract most of the true contour feature points but also does not cause redundancy of data points.

VII. APPLICATION

In order to evaluate the application performance of the proposed method, we use the extracted contour feature points to conduct the deformation analysis. We use the RIEGL VZ400 scanner to scan the “Wuhan xx Yangtze River Bridge” five times, and the scanning interval is set to 30 min according to the changes in traffic flow. The bridge mainly includes the road of the bridge, cable bent tower, and stay cable, as shown in Fig. 20(a). The distance between the scanner and cable bent tower is approximately 150 m; the height of the cable-bent tower is approximately 125 m. The scanner has a scanning distance of 600 m and a positional accuracy of 3 mm within 100 m. We set the range resolution of the scan to 0.05 m, and the scanning point cloud is obtained, as shown in Fig. 20(b).

We use the proposed method to extract the bridge contour points, which mainly includes the feature points of cable bent tower and stay cable, as shown in Fig. 21.

In order to analyze the deformation of the bridge, we mainly analyze the skewing of two stay cables on both sides of the cable-stayed tower and the inclination of the cable-stayed tower on the upper side of the tower. The linear equation

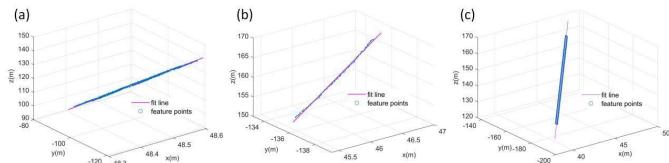


Fig. 22. Spatial linear fitting of feature points. (a) Left stay cables. (b) Cable bent tower. (c) Right stay cables.

TABLE VI
FITTING PARAMETERS OF DIFFERENT SPATIAL LINES AT DIFFERENT TIMES

Different time	1	2	3	4	5
cable bent tower	a	0.085	0.085	0.085	0.085
	b	32.364	32.362	32.369	32.335
	c	0.005	0.005	0.003	0.005
	d	-137.032	-137.108	-136.690	-137.041
left stay cables	a	0.005	0.005	0.005	0.005
	b	47.836	47.826	47.846	47.838
	c	-0.757	-0.757	-0.757	-0.757
	d	-13.545	-13.551	-13.541	-13.542
right stay cables	a	0.216	0.216	0.215	0.215
	b	13.034	13.040	13.135	13.174
	c	1.225	1.225	1.223	1.222
	d	-338.793	-338.775	-338.495	-338.409

TABLE VII
INCLUDED ANGLE OF THE FIT LINE AT DIFFERENT TIMES

	First and second	First and third	First and fourth	First and fifth
Cable bent tower	0.0276	0.0162	0.0171	0.0602
Left stay cables	0.0048	0.0018	0.0046	0.0085
Right stay cables	0.0027	0.0473	0.0614	0.0033

of 3-D space points is expressed, as shown in the following equation:

$$\begin{cases} x = az + b \\ y = cz + d. \end{cases} \quad (36)$$

According to (36), we use least squares to conduct the spatial linear fitting of feature points of two stay cables and cable bent tower, as shown in Fig. 22.

Simultaneously, the fitting parameters of different feature lines at different times are obtained, as shown in Table VI.

The first scan is considered as the baseline, the other four scans are compared to it, and the included angles between two fit spatial lines at different times are obtained according to the fitting parameters, as shown in Table VII.

The horizontal and vertical scan angle step sizes are $0.0024^\circ - 0.5^\circ$ and $0.0024^\circ - 0.288^\circ$, respectively. For the cable-stayed tower, the included angle between the first and fifth fitting lines is the largest; it is only 0.0602° . The included angle between the fitting lines of the other two scans is approximately 0.02° . By the same token, for the left and right stay cables, the included angles between the fitting lines of the other scan and the first scan are all very small, and they are all within the range of the scan angle step size. It illustrates

that the cable bent tower is not inclined, and the left and right stay cables have no skewing.

VIII. CONCLUSION

In this article, we propose a novel method to extract the contour feature points, which mainly includes the extraction of boundary and crease points. For the boundary points, the optimal number of neighboring points is determined according to the information entropy of the neighborhood of each point. Then, the vectors are constructed between the object and domain points, and the conspicuous boundary points are determined according to the distribution characteristic of the include angle between the two vectors. Simultaneously, the inconspicuous boundary points are obtained according to the span of the azimuth after the distance of neighboring points is sorted. For the crease points, the three eigenvectors are obtained by performing the principal component analysis (PCA) on the point cloud in a 2-D view inside the bounding sphere. We find that the eigenvector e_1 corresponding to the maximum eigenvalue λ_1 is always along the crease direction. According to the eigenvector, the projection plane is constructed. Then, the crease points are determined according to the distribution mechanism of neighboring points in the bounding sphere on the projection plane. The proposed method can be applied not only to ground LiDAR scanning data but also to airborne LiDAR scanning data. Simultaneously, it is not influenced by the type of point clouds.

The two nonfixed parameters have a great influence on the extraction effect of crease points. We give the judgment method of the optimal value of two nonfixed parameters. Experiment results show that the proposed method is superior to the BI, OAHT, Chen, FSB, and region clustering curvature methods. Furthermore, the proposed method is not only suitable for flat-shaped buildings but also for curvilinear architectural objects. Besides, the proposed method in this article involves fewer parameters to be tuned than the other methods, and the robustness of the proposed method is better than that of the latest contour feature point extraction method. The proposed method successfully extracts the contour feature points of the “Wuhan xx Yangtze River Bridge,” and then, the fit feature lines are used to analyze the skewing of two stay cables and the inclination of the cable bent tower.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

DATA AVAILABILITY

Data will be made available on request.

REFERENCES

- [1] X. T. Guo and T. Huang, “Rock-based landslide safety monitoring using terrestrial laser scanning technology,” *Lasers Eng.*, vol. 37, nos. 1–3, pp. 17–34, 2017.
- [2] E. Tait, R. Laing, A. Grinnall, S. Burnett, and J. Isaacs, “(Re)presenting heritage: Laser scanning and 3D visualisations for cultural resilience and community engagement,” *J. Inf. Sci.*, vol. 42, no. 3, pp. 420–433, Jun. 2016.

- [3] B.-X. Zhou, J.-P. Yue, G.-Y. Xi, and J. Li, "Rigid-body deformation monitoring of buildings based on terrestrial laser scanning (TLS) technology," *Lasers Eng.*, vol. 27, nos. 5–6, pp. 277–288, 2014.
- [4] M. S. O'Banion, M. J. Olsen, J. P. Hollenbeck, and W. C. Wright, "Data gap classification for terrestrial laser scanning-derived digital elevation models," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 12, p. 749, Dec. 2020.
- [5] H. Ni, X. Lin, X. Ning, and J. Zhang, "Edge detection and feature line tracing in 3D-point clouds by analyzing geometric properties of neighborhoods," *Remote Sens.*, vol. 8, no. 9, pp. 710–718, 2016.
- [6] Y. Lin, C. Wang, B. Chen, D. Zai, and J. Li, "Facet segmentation-based line segment extraction for large-scale point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 4839–4854, Sep. 2017.
- [7] M. Pepe, D. Costantino, V. S. Alfio, G. Vozza, and E. Cartellino, "A novel method based on deep learning, GIS and geomatics software for building a 3D city model from VHR satellite stereo imagery," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 10, p. 697, Oct. 2021.
- [8] B. Yang, L. Fang, and J. Li, "Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 79, pp. 80–93, May 2013.
- [9] A. Habib, M. Ghamma, M. Morgan, and R. Al-Ruzouq, "Photogrammetric and LiDAR data registration using linear features," *Photogram. Eng. Remote Sens.*, vol. 71, no. 6, pp. 699–707, Jun. 2005.
- [10] E. J. Almazàn, R. Tal, Y. Qian, and J. H. Elder, "MCMLSD: A dynamic programming approach to line segment detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5854–5862.
- [11] K. Hildebrandt, K. Polthier, and M. Wardetzky, "Smooth feature lines on surface meshes," in *Proc. Symp. Geometry Process.*, 2005, pp. 85–90.
- [12] Y. Ohtake, A. Belyaev, and H.-P. Seidel, "Ridge-valley lines on meshes via implicit surface fitting," in *Proc. ACM SIGGRAPH Papers*, Aug. 2004, pp. 609–612.
- [13] H. T. Ho and D. Gibbons, "Curvature-based approach for multi-scale feature extraction from 3D meshes and unstructured point clouds," *IET Comput. Vis.*, vol. 3, no. 4, pp. 201–212, Dec. 2009.
- [14] Q. Gao and Y. Yamaguchi, "Extraction of coherent and smooth feature lines from meshes with fine details," *Comput. Graph.*, vol. 82, pp. 222–231, Aug. 2019.
- [15] N. Salman, M. Yvinec, and Q. Merigot, "Feature preserving mesh generation from 3D point clouds," *Comput. Graph. Forum*, vol. 29, no. 5, pp. 1623–1632, Sep. 2010.
- [16] S. Gumhold, X. Wang, and R. S. MacLeod, "Feature extraction from point clouds," in *Proc. 10th Int. Meshing Roundtable*, 2001, pp. 293–305.
- [17] K. Demarsin, D. Vanderstraeten, T. Volodine, and D. Roose, "Detection of closed sharp edges in point clouds using normal estimation and graph theory," *Comput.-Aided Des.*, vol. 39, no. 4, pp. 276–283, Apr. 2007.
- [18] M. Pauly, R. Keiser, and M. Gross, "Multi-scale feature extraction on point-sampled surfaces," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 281–289, Sep. 2003.
- [19] X. Liu and C. Jin, "Feature line extraction from unorganized noisy point clouds," *J. Comput. Inf. Syst.*, vol. 10, no. 8, pp. 3503–3510, 2014.
- [20] T. Hackel, J. D. Wegner, and K. Schindler, "Contour detection in unstructured 3D point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1610–1618.
- [21] I. Elkhrachy, "Feature extraction of laser scan data based on geometric properties," *J. Indian Soc. Remote Sens.*, vol. 45, no. 1, pp. 1–10, Feb. 2017.
- [22] E. Altantsetseg, Y. Muraki, K. Matsuyama, and K. Konno, "Feature line extraction from unorganized noisy point clouds using truncated Fourier series," *Vis. Comput.*, vol. 29, nos. 6–8, pp. 617–626, Jun. 2013.
- [23] J. Prochazkova, D. Procházka, and J. Landa, "Sharp feature detection as a useful tool in smart manufacturing," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 7, 2020, Art. no. 422.
- [24] D. Bazazian, J. R. Casas, and J. Ruiz-Hidalgo, "Fast and robust edge extraction in unorganized point clouds," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, New York, NY, USA, Nov. 2015, pp. 1–8.
- [25] S.-K. Kim, "Extraction of ridge and valley lines from unorganized points," *Multimedia Tools Appl.*, vol. 63, no. 1, pp. 265–279, Mar. 2013.
- [26] M. K. Park, S. J. Lee, and K. H. Lee, "Multi-scale tensor voting for feature extraction from unstructured point clouds," *Graph. Models.*, vol. 74, no. 4, pp. 197–208, Jul. 2012.
- [27] H. Lin, W. Wang, Y. Shao, and D. Lei, "Feature extraction from unorganized point cloud based on analytical tensor voting," *J. Graph.*, vol. 38, no. 2, pp. 137–143, 2017.
- [28] X. Lu, Y. Liu, and K. Li, "Fast 3D line segment detection from unorganized point cloud," 2019, *arXiv:1901.02532*.
- [29] D. Bazazian, J. R. Casas, and J. Ruiz-Hidalgo, "Segmentation-based multi-scale edge extraction to measure the persistence of features in unorganized point clouds," in *Proc. 12th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, Porto, Portugal, 2017, pp. 317–325.
- [30] X. Wang, H. Chen, and L. Wu, "Feature extraction of point clouds based on region clustering segmentation," *Multimedia Tools Appl.*, vol. 79, nos. 17–18, pp. 11861–11889, May 2020.
- [31] X. Chen and K. Yu, "Feature line generation and regularization from point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9779–9790, Dec. 2019.
- [32] A. Manno-Kovacs and T. Sziranyi, "Orientation-selective building detection in aerial images," *ISPRS J. Photogramm. Remote Sens.*, vol. 108, pp. 94–112, Oct. 2015.
- [33] M. Awrangjeb, "Using point cloud data to identify, trace, and regularize the outlines of buildings," *Int. J. Remote Sens.*, vol. 37, no. 3, pp. 551–579, Feb. 2016.
- [34] D. O. Richard and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Artif. Intell. Center*, vol. 15, no. 1, pp. 11–15, 1972.
- [35] E. Widyaningrum, B. Gorte, and R. Lindenbergh, "Automatic building outline extraction from ALS point clouds by ordered points aided Hough transform," *Remote Sens.*, vol. 11, no. 14, pp. 1727–1736, 2019.
- [36] E. Widyaningrum, B. Gorte, and R. Lindenbergh, "Automatic building outline extraction from ALS point clouds by ordered points aided Hough transform," *Remote Sens.*, vol. 11, no. 14, pp. 16–29, 2019.
- [37] E. Widyaningrum, R. Y. Peters, and R. C. Lindenbergh, "Building outline extraction from ALS point clouds using medial axis transform descriptors," *Pattern Recognit.*, vol. 106, Oct. 2020, Art. no. 107447.
- [38] J. Kustra, A. Jalba, and A. Telea, "Computing refined skeletal features from medial point clouds," *Pattern Recognit. Lett.*, vol. 76, pp. 13–21, Jun. 2016.
- [39] X. Chen, X. Hua, G. Zhang, H. Wu, W. Xuan, and M. Li, "Evaluating point cloud accuracy of static three-dimensional laser scanning based on point cloud error ellipsoid model," *J. Appl. Remote Sens.*, vol. 9, no. 1, Nov. 2015, Art. no. 095991.
- [40] X. Chen, K. Yu, and H. Wu, "Determination of minimum detectable deformation of terrestrial laser scanning based on error entropy model," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 105–116, Jan. 2018.
- [41] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," in *Proc. DARPA Broadcast News Workshop*, 1999, pp. 249–252.



Xijiang Chen was born in Anhui, China, in December 1984. He received the Ph.D. degree from the School of Geodesy and Geomatics, Wuhan University, Wuhan, China, in 2014.

He is currently an Associate Professor with the School of Safety Science and Emergency Management, Wuhan University of Technology, Wuhan. His research interests include laser scanning and data processing.



Bufan Zhao was born in Guilin, Guangxi, China, in 1994. He received the B.S., M.S., and Ph.D. degrees from the School of Geodesy and Geomatics, Wuhan University, Wuhan, China, in 2016, 2019, and 2022, respectively.

He is currently a Lecturer with the School of Safety Science and Emergency Management, Wuhan University of Technology, Wuhan. His research interests include 3-D laser scanning data processing and 3-D modeling.