Hindawi
Advances in Civil Engineering
Volume 2024, Article ID 5891852, 18 pages
https://doi.org/10.1155/2024/5891852

Research Article

Extraction of Feature Information from Point Cloud of Large Volume Steel Truss Members

Dong Liang, [1,2] Zeyu Zhang, [1,2] Bo Guo, [1,2] Bo Wang, [1] and Xiongjue Wang [1]

[1] *State Key Laboratory for Health and Safety of Bridge Structures, Wuhan 430034, China*
[2] *School of Civil and Transportation Engineering, Hebei University of Technology, Tianjin 300401, China*

Correspondence should be addressed to Zeyu Zhang; 202131603002@stu.hebut.edu.cn

Received 18 July 2023; Revised 8 October 2023; Accepted 2 December 2023; Published 4 January 2024

Academic Editor: José A. F. O. Correia

In order to address the issues pertaining to the subjective nature and limited precision associated with selecting feature points in the point cloud of a large steel truss structure, this study proposes a batch automatic extraction approach for identifying key feature information, including boundaries, corner points, and bolt holes of large steel truss components. This method relies on the nested application of established processing algorithms such as Euclidean clusters, regional growth clusters, and random sampling consensus. In addition, a novel approach is suggested for validating the precision of feature information extraction through the utilization of standard theoretical models. The results of the experimental and large-scale lower chord tests demonstrate that our approach is not dependent on specialized software, exhibits excellent efficiency, and possesses an acceptable degree of automation. The findings of this study can provide accurate data support for reverse modeling, virtual trial assembly, and dimensional inspection of steel truss components.

1. Introduction

The prevalence of steel structures in bridge construction can be attributed to several factors, including the abundant availability of steel as a material, the well-established industrial production methods for steel structures, the favorable mechanical characteristics exhibited by steel, the convenient transportability of steel components, and the efficient installation processes associated with steel bridges. Recently, there has been an increasing scholarly focus on the advancement of 3D point cloud processing technologies. This technology is a mechanism for establishing connections between physical structures and digitized information. The utilization of computer technology and 3D measurement technology has supplanted conventional approaches, such as outbound quality checking, overall deformation monitoring, and preassembly of steel structures. The execution of these duties has transitioned to a digital format facilitated by point clouds [1–4].

The structural dimensions of large steel truss members are mostly expressed through the borders, corner points, and centers of bolt holes. In the structural fabrication process, the geometries and dimensions of the members can be verified to meet the design requirements through accurate measurement. The extraction of boundary points, corner points, and bolt hole centers plays an essential role in the virtual trial assembly based on point clouds. These assembly points give vital geometric information about steel truss components in an intuitive manner. The closely related work with this study is the feature extraction on steel members with bolted connections. In the boundary detection of truss members, Lamas et al. [5] proposed an algorithm for the automatic extraction and segmentation of truss elements. The algorithm begins by sectioning the truss into horizontal and vertical planes, then uses PCA and clustering algorithms to measure and judge the neighborhood distribution of each point and finally completes the automatic extraction of truss elements. Fotsing et al. [6] proposed a novel region-growing-based method for fast extraction of planes of structure in point clouds. The method uses an iterative closest point algorithm to extract reliable seeds, and a voxel grid representation of the point cloud is used in the growth process. Based on the above method, Truong-Hong and Lindenbergh [7] coarsely extract the original point cloud, which effectively improves the accuracy of feature extraction for complex bridge structures. The process starts with the coarse extraction of candidate points by decomposing the point cloud of the bridge into 2D cells and analyzing the distribution of data points within the cells in a vertical direction. Kim et al. [8] used a hybrid algorithm with mean-shift clustering, nonconvex hull, and random sample and consensus (RANSAC) to extract geometric primitives with solid lines on edges from point clouds to digitalize reverse engineering in the engineering and construction domains. In the extraction of bolt hole feature information, Truong-Hong et al. [9] calculated the eigenvectors and eigenvalues of the correlation matrix formed by the kNN points and used the angle criterion to detect the boundary points of the circle. Zhou et al. [10] used principal component analysis (PCA) to convert the 3D point cloud into 2D data, combined with a clustering algorithm and kNN algorithm to extract the boundaries of bolt holes. Tabib et al. [11] proposed a deep learning framework to detect whether each point in the point cloud is a hole boundary point, and the detected hole boundary points are used for hole filling by fitting a surface and interpolating points on the surface. After extracting the boundary points for the bolt holes, fitting the data points using an algebraic method is relatively easy to solve. However, the circular objects are often partially hidden in practical situations as the boundary points on the occlusion boundary fall inside the circle, which behave as outliers and may strongly affect the estimate of the circle's parameters. To solve the aforementioned problem, a random sampling consistency algorithm [12], Hough transform [13], and robust least squares [14] have been adopted.

Current approaches for extracting features from point clouds are unable to completely automate the extraction of many features at the same time. To overcome the above problems, Xu et al. [15] used the spatial position relationships and geometric features of the structure in the point cloud model to set constraints and the Euclidean clustering algorithm to extract columns and beams in the frame structure synchronously. However, the algorithm has many extraction parameters, poor automation of different types of feature extraction, and is only applicable to typical assembled frame structures. Wang et al. [16] built and set the feature parameter thresholds by vector angle criterion and extracted the boundary feature points and sharp feature points in the point cloud synchronously by traversing each K-neighborhood point and discriminating the feature parameter. Cabaleiro et al. [17] proposed a method to extract the connection centroid, the contour of the bars, and the orientation of the connection centroid in the point cloud of the metal frames, which was based on the 2.5D density image. The Hough transform was used to determine the flange and web lines of the steel frame connections. Finally, the coordinates of the connection center points were selected based on the orthogonal geometric relationships of the bars. Wang et al. [18], for the dimensional inspection and building information modeling of the precast concrete bridge deck, first converted the 3D point cloud into a 2D image, then implemented multifeature extraction of precast concrete bridge decks using an edge line estimation algorithm, finally determined all the edge lines and corner points of the structure. Due to their high efficiency and accuracy, deep learning algorithms have been increasingly applied to multifeature point cloud extraction in recent years. Still, the existing algorithms focus on the coarse-scale semantic representation of the target rather than the detailed features. To address the above problems, Du et al. [19] designed a multiscale feature fusion segmentation network (MSSCN) to realize the extraction of detailed features in point clouds. However, the method has no advantage in terms of the accuracy of feature boundary segmentation.

A prevalent issue with these current techniques is their inability to automatically extract every feature straight from the point cloud of a steel truss component. This presents a challenge to automated point cloud analysis used in virtual trial assembly and reverses modeling based on point clouds of bolted connecting parts. When it comes to processing large structural point clouds, machine learning and deep learning methods are computationally complex and take a long time to process feature information. On the other hand, traditional methods neither process the information of all the feature points (such as corners, boundaries, and bolt holes) of large steel truss members at once nor can they be automatically extracted.

To address this issue, a novel approach is presented in this paper for the batch automatic extraction of key feature data, including bolt holes, corners, and boundaries, from large steel truss member point clouds. These extracted point clouds can subsequently be utilized in reverse modeling and the virtual trial assembly of said truss members. The method consists of two parts: part 1: extraction of boundary and corner points and part 2: extraction of bolt hole features. Moreover, a method to verify the feature extraction accuracy using a standard point cloud model is proposed.

The contributions to this work are as follows: (1) a feature extraction method for large bolted steel members is established, and the developed nested algorithm realizes the simultaneous automatic extraction of all feature information for large steel truss members. (2) A novel approach is provided for extracting boundary points from cross-section and side point clouds of large complex members. The method combines the region expanding algorithm, clustering algorithm, and RANSAC to achieve intelligent point extraction. Additionally, correct coordinates of corner points are obtained through the utilization of plane intersection. (3) A fast extraction method for bolt hole feature information is proposed based on classical algorithms such as alpha shapes and Euclidean clustering segmentation. (4) A new verification method for the extraction accuracy of the algorithm is proposed. The method verifies the extraction accuracy of the algorithm by establishing a standard point cloud model and comparing the differences between the design and feature extraction dimensions.

2. Feature Information Extraction Algorithm

The feature information of large-volume steel trusses mainly includes boundaries, corner points, and bolt hole groups. The main process of boundary and corner point extraction is as follows: first, the plane point cloud is segmented by

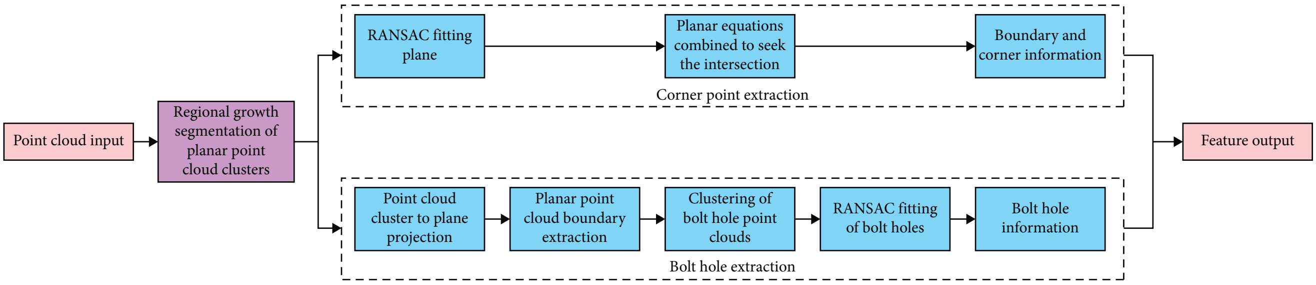

Point cloud input
Regional growth segmentation of planar point cloud clusters
RANSAC fitting plane
Planar equations combined to seek the intersection
Boundary and corner information
Corner point extraction
Point cloud cluster to plane projection
Planar point cloud boundary extraction
Clustering of bolt hole point clouds
RANSAC fitting of bolt holes
Bolt hole information
Bolt hole extraction
Feature output

Figure 1: Point cloud feature extraction flowchart.

using the region growth algorithm; then the plane equations are fitted by using the random sampling consistency algorithm; finally, the boundary and corner point feature information is obtained through the intersection of the planes; for the extraction of bolt hole clusters, the bolt hole point cloud is first segmented by using the Euclidean clustering based on the boundary extraction; then the bolt hole point cloud is fit by using the random sampling consistency algorithm; finally, the bolt hole feature information is obtained by using the random sampling consistency algorithm. The objective of batch processing numerous point cloud clusters is accomplished by nesting the fitting step within the clustering algorithm. The specific flow is shown in Figure 1.

This work proposes to use the standard point cloud model for the test in order to verify the precision of the above feature extraction algorithm and prevent the influence of external factors on the size and shape of the steel structure throughout the construction process. Last but not least, the test depends on a large-scale lower chord to confirm the algorithm's suitability for real-world engineering applications.

2.1. Automatic Batch Extraction of Boundary and Corner Features. In this paper, three nonparallel planes are used to calculate the boundary and corner points, so it is especially critical to automatically obtain all the plane equations in a large point cloud, and the main steps are as follows.

2.1.1. Planar Point Cloud Clusters Segmented by Regional Growth Clustering. The point cloud segmentation algorithm based on region growth is a commonly used method for extracting point cloud clusters in the end plane of rods [20]. Each set of outputs using the region growth segmentation plane is recognized as part of the same smooth surface, making the algorithm optimal for planar extraction of point clouds at the ends of rods. In addition, the algorithm is robust to noise and local shape variations in the point cloud, enabling accurate segmentation in complex point cloud environments. The selection of seed points and the growth criterion are key factors affecting the quality of point cloud segmentation. This study used the points with the least curvature as seed points because starting growth in a smooth region reduces the total number of regions and avoids overlapping segments. The normal plane is solved by the data points with their neighboring points, so the setting of the number of neighboring points affects the result of solving the normal vector. Set the *NormalSearch* parameter of the normal vector neighborhood search value, which indicates the number of neighboring points involved in solving the normal vector. In the setting of the growth criterion, information such as geometric features, curvature, and vectors of the point cloud model are used as judgment conditions. *MinClusterSize* and *MaxClusterSize* represent the minimum and maximum number of clustered points, respectively, and values less than the minimum or greater than the maximum are ignored, so this parameter should be set according to the number of point clouds in the minimum and maximum planes of the member. *NumberOfNeighbors* represents the number of points in the test near the seed point as the region grows, indicating the number of point clouds needed to define a plane. Smoothness and curvature are the two most essential thresholds. Smoothness represents the value of the angle between the normal vector of the current seed point and each neighboring point, and if it is less than the set threshold, the neighboring point will be given priority. *Curvature* represents the curvature threshold. If a neighboring point is less than the set value, the point will be added to the set of seed points. The geometric characteristics of the trusses are thin-walled box sections, and the parameters selected in this study based on the density of the captured point cloud model are $\mathit{NormalSearch} = 10$, $\mathit{MinClusterSize} = 100$, $\mathit{MaxClusterSize} = 500{,}000$, $\mathit{NumberOfNeighbors} = 10$, $\mathit{Smoothness} = 5/180 \times \mathrm{M\_PI}$, $\mathit{Curvature} = 0.05$.

The planar extraction process of the region growing method is shown in Figure 2. The planar clustering algorithm based on region growth takes each extracted plane as a clustering condition to complete the segmentation of all planes. Figure 3 shows the effect of the point cloud region growing clustering segmentation of planes. Different colors are set for different point cloud clusters in the segmented point cloud to make the clustering segmentation effect more intuitive.

2.1.2. RANSAC Fitting Plane Equations. The random sampling consistency algorithm is a method for estimating the parameters of a mathematical model in a dataset containing various defects such as noise, outliers, and so forth. Steel trusses consist of regular flat planes and the most common and simplest way to fit the planes is least squares. However, the number of noise points in the point cloud is increased due to the large number of bolt holes distributed in the member. The noise points are composed of a point cloud of the hole wall and a point cloud of the hole's edge where the

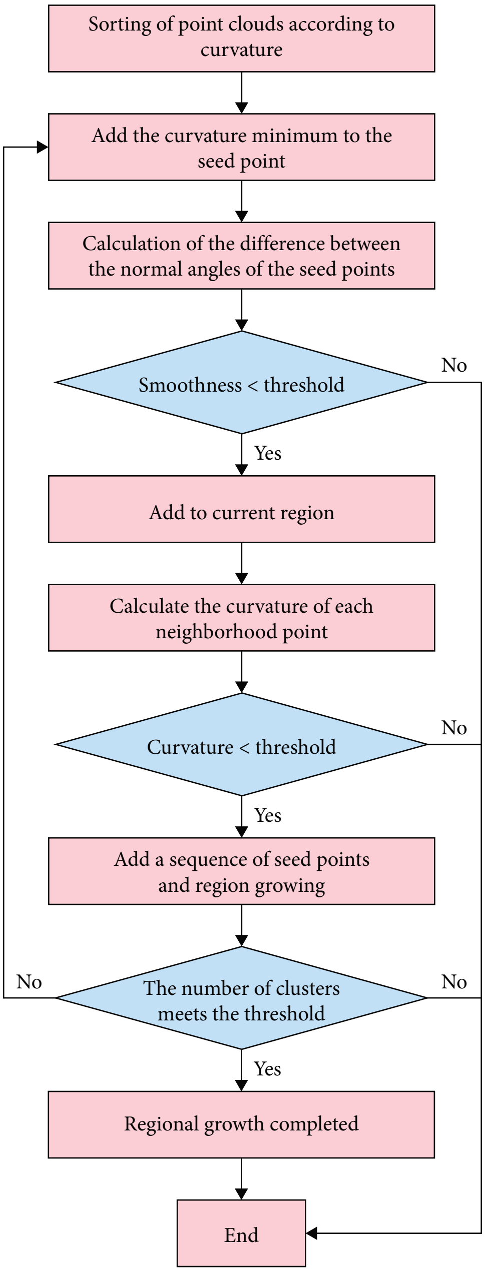

Sorting of point clouds according to curvature
Add the curvature minimum to the seed point
Calculation of the difference between the normal angles of the seed points
Smoothness < threshold
No
Yes
Add to current region
Calculate the curvature of each neighborhood point
Curvature < threshold
No
Yes
Add a sequence of seed points and region growing
No
The number of clusters meets the threshold
No
Yes
Regional growth completed
End

Figure 2: Region growth algorithm flowchart.

drift effect occurs. The fitting accuracy of the least squares method is easily affected by noise points, so RANSAC is used to improve the fitting accuracy by eliminating the influence of noise points through iterative fitting. The steps and parameter settings of the method are as follows:

Step 1: Three points are randomly selected from the acquired planar point cloud clusters, and then the planar model parameters A, B, C, and D are calculated according to the planar equations:

$$Ax + By + Cz + D = 0. \tag{1}$$

Step 2: Calculate the distance of all other points to the plane; if it is less than *DistanceThreshold*, these points are considered to be in the same plane as the plane. Therefore, *DistanceThreshold* should be set according to the distribution of noise points in the plane in the point cloud. In this study, the selected *DistanceThreshold* is 2.0.

Step 3: Save the plane and mark the points as matched if there are more than n points in the same plane.

Step 4: The termination condition is that three unlabeled points cannot be found after N iterations or the number of plane points is less than n. The setting of the iteration number N determines whether the final fitted plane is reliable, and this paper sets the size of the parameter N according to Equation (2), where p is the expected probability of selecting at least one sample from the outliers, which is set to 0.99, and e is the outlier rate. The maximum number of iterations chosen is $N = 500$:

$$\left(1 - (1 - e)^s\right)^N = 1 - p. \tag{2}$$

Repeat the above steps to continuously update the model parameters, saving the model parameters with the highest number of interior points. Finally, the internal points estimate the model parameters, as shown in Figure 4.

2.1.3. Intersecting Planes to Get a Boundary and Corner Information. According to all the plane equations in the end point cloud obtained above, as shown in Figure 5, the equations are combined sequentially, with each of the three equations as a group, and solved simultaneously. The positional relationship of the three planes in space is related to the rank of the augmentation and generalization matrices of the system of linear equations in which they are associated. If the nonzero row (rank) of the coefficient matrix of the system of linear equations $r(A) = 3$, the three planes intersect at one point, as shown in Figure 6. Therefore, this can be used as a decision condition to eliminate groups of planes that are not parallel and complete the subsequent processing.

Consider a system of equations. The equations of the three nonparallel planes are as follows:

$$\begin{cases} A_1x + B_1y + C_1z + D_1 = 0 \\ A_2x + B_2y + C_2z + D_2 = 0 \, . \\ A_3x + B_3y + C_3z + D_3 = 0 \end{cases} \tag{3}$$

Then, the normal vector of the intersecting line between planes 1 and 2 is $\vec{n} = (p,\ q,\ r)$, calculated as follows:

$$p = \begin{vmatrix} B_1 & C_1 \\ B_2 & C_2 \end{vmatrix}, q = \begin{vmatrix} C_1 & A_1 \\ C_2 & A_2 \end{vmatrix}, r = \begin{vmatrix} A_1 & C_1 \\ A_2 & C_2 \end{vmatrix}. \tag{4}$$

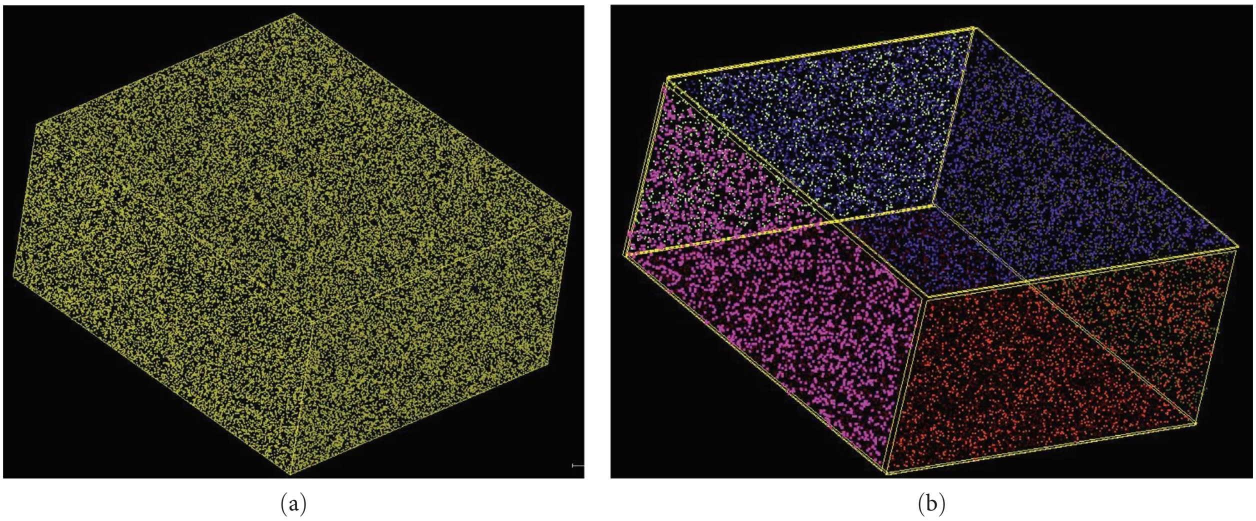


Figure 3: Plan view of region growth clustering: (a) original point cloud and (b) point cloud after plan segmentation.

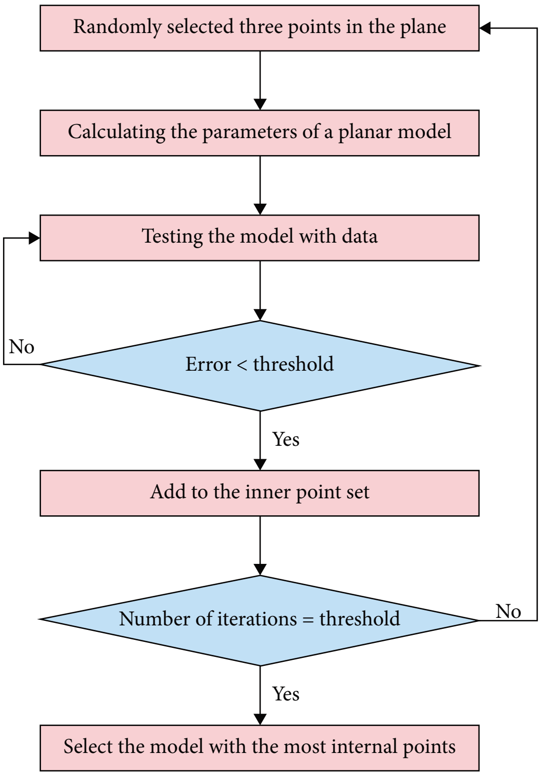

Randomly selected three points in the plane
Calculating the parameters of a planar model
Testing the model with data
No
Error < threshold
Yes
Add to the inner point set
Number of iterations = threshold
No
Yes
Select the model with the most internal points

Figure 4: Planar equation fitting flowchart.

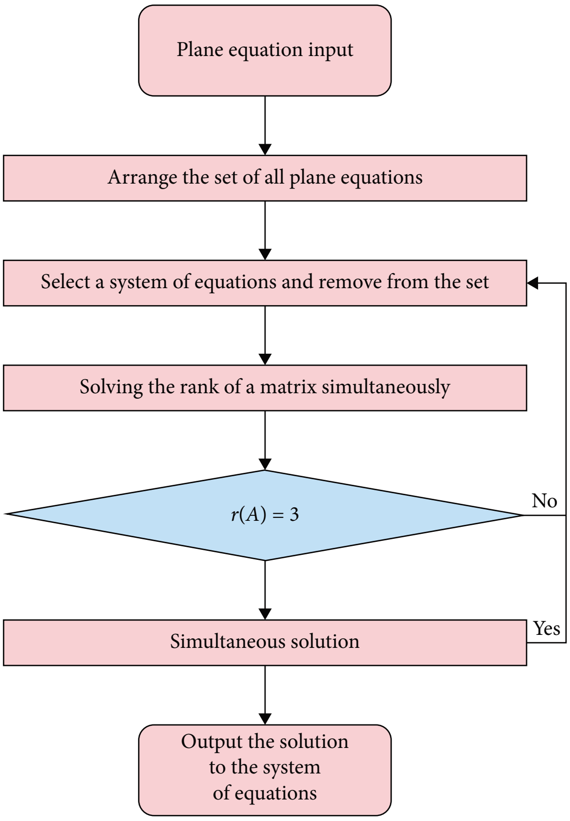

Plane equation input
Arrange the set of all plane equations
Select a system of equations and remove from the set
Solving the rank of a matrix simultaneously
r(A) = 3
No
Simultaneous solution
Yes
Output the solution to the system of equations

Figure 5: Corner point capture flowchart.

The line can be expressed as follows:

$$\begin{cases} x = x_0 + pt \\ y = y + 0 + qt(-\infty < t < +\infty), \\ z = z_0 + rt \end{cases} \tag{5}$$

Putting x, y, and z into $A_3x + B_3y + C_3z + D_3 = 0$, we can solve for t and thus find the coordinates of the intersection point (x_0, y_0, z_0), where:

$$x_0 = \frac{x_D}{D},\ y_0 = \frac{y_D}{D},\ z_0 = \frac{z_D}{D}, \tag{6}$$

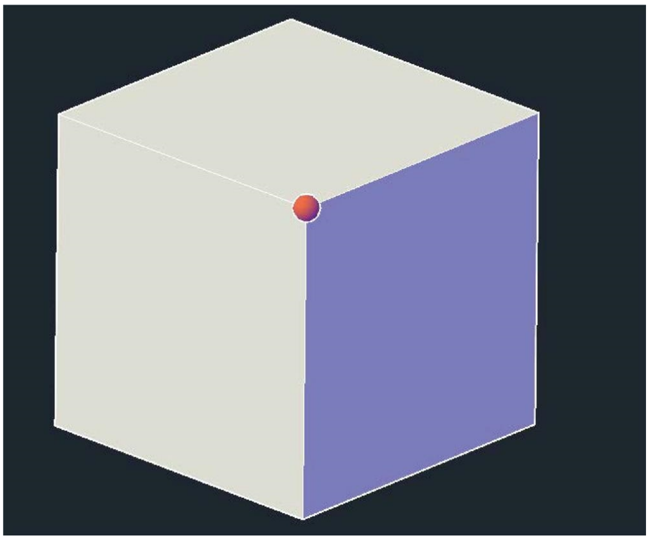


Figure 6: Schematic diagram of plane intersection.

where:

$$D=\begin{bmatrix}A_1 & B_1 & C_1\\ A_2 & B_2 & C_2\\ A_3 & B_3 & C_3\end{bmatrix};X_D=\begin{bmatrix}-D_1 & B_1 & C_1\\ -D_2 & B_2 & C_2\\ -D_3 & B_3 & C_3\end{bmatrix};$$
$$Y_D=\begin{bmatrix}A_1 & -D_1 & C_1\\ A_2 & -D_2 & C_2\\ A_3 & -D_3 & C_3\end{bmatrix};Z_D=\begin{bmatrix}A_1 & B_1 & -D_1\\ A_2 & B_2 & -D_2\\ A_3 & B_3 & -D_3\end{bmatrix}. \tag{7}$$

2.2. Automatic Batch Extraction of Bolt Hole Features. The process of extracting information from bolt holes generally consists of four key processes. These steps include projecting planar point cloud clusters, extracting boundaries from the planar point cloud, segmenting bolt hole point cloud clusters using Euclidean clustering, and fitting equations to the bolt holes. Each of these steps is described in detail below.

2.2.1. Planar Point Cloud Cluster Projection. In bolt hole point cloud segmentation, the plane point cloud cluster projection is performed first. For the steel truss members, the bolt holes at the end of the truss are all standard, and the error in the thickness direction of the holes is negligible, so the point clouds of the bolt holes on the same side are all projected onto the outer plane equation. According to the density distribution of the projected point cloud of the bolt hole surface, the hole center coordinates can be obtained more accurately, which increases the robustness of the bolt hole extraction algorithm. The general equation of the 3D space plane is shown in Equation (1).

Assume that the coordinates of the 3D space not in the plane are (x_0, y_0, z_0) and the coordinates of its projection point in the plane are (x_p, y_p, z_p). Since the current point to the projection point is perpendicular to the plane, according to the vertical constraint, it is known that y_p and z_p satisfy the following conditions:

$$y_p=\frac{B}{A}\left(x_p-x_0\right)+y_0, \tag{8}$$

$$z_p=\frac{C}{A}\left(x_p-x_0\right)-z_0. \tag{9}$$

Substituting Equations (8) and (9) into Equation (1):, we get the following equations:

$$x_p=\frac{(B^2+C^2)x_0-A\left(By_0+C_{z_0}+D\right)}{A^2+B^2+C^2}, \tag{10}$$

$$y_p=\frac{(A^2+C^2)x_0-B\left(By_0+C_{z_0}+D\right)}{A^2+B^2+C^2}, \tag{11}$$

$$z_p=\frac{(B^2+A^2)x_0-C\left(By_0+C_{z_0}+D\right)}{A^2+B^2+C^2}. \tag{12}$$

This gives the projection coordinates (x_p, y_p, z_p) of the 3D point in space onto the plane. Figure 7 displays the projection of the point cloud onto the plane. The original point cloud is represented by the white points, while the green points represent the projected point cloud.

2.2.2. Planar Point Cloud Boundary Extraction. The alpha shapes algorithm was first proposed by Edelsbrunne. The algorithm uses a circle of radius α to roll outside the point set P. If α is large enough, the circle will not roll inside the point set, as shown in Figure 8. Inside the point set P, a circle of radius α can be drawn over any two points p_1, p_2. If there is no other point inside the circle, p_1 and p_2 are considered boundary points, and the line p_1, p_2 between them is the boundary segment, as shown in Figure 9. The points $p_1(x_1,\ y_1)$ and $p_2(x_2,\ y_2)$ are known, and the center $p_3(x_3,\ y_3)$ of the circle over these two points can be obtained by the posterior distance rendezvous method. After obtaining the center of the circle, by judging the relationship between the distance of other points to the center of the circle and the radius α, it can be determined whether there are other points in the circle. The value of the parameter α determines the accuracy of the point cloud contour extraction. This implies that as α increases, fewer edge points may exist in a point set P, which blurs the outline details of P. Thus, the value of α must be set appropriately. In this study, the parameter α is set to 50.0 based on the density of the point cloud model.

$$\begin{cases} x_3=x_1+\dfrac{1}{2}(x_2-x_1)+H(y_2-y_1)\\ y_3=y_1+\dfrac{1}{2}(y_2-y_1)+H(x_1-x_2)\end{cases}, \tag{13}$$

where $H=\sqrt{\frac{a^2}{(x_1-x_2)^2-(y_1-y_2)^2}}-\frac{1}{4}$.

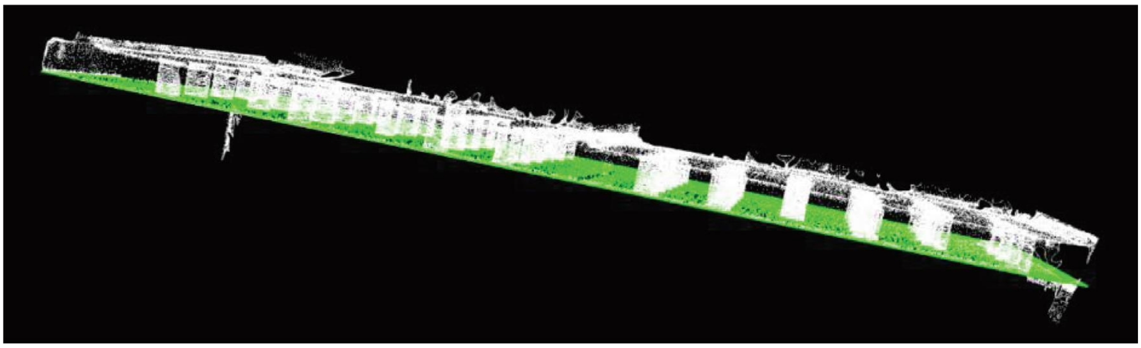


Figure 7: Point cloud to plane projection diagram.

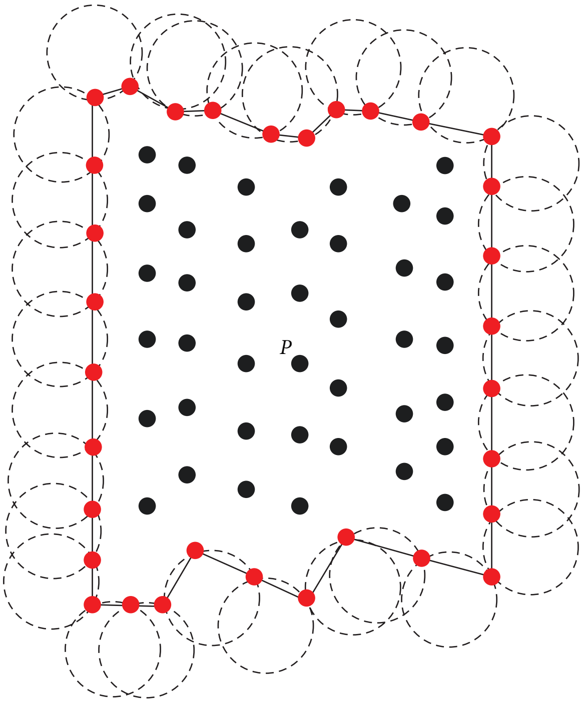


Figure 8: Boundary point extraction diagram.

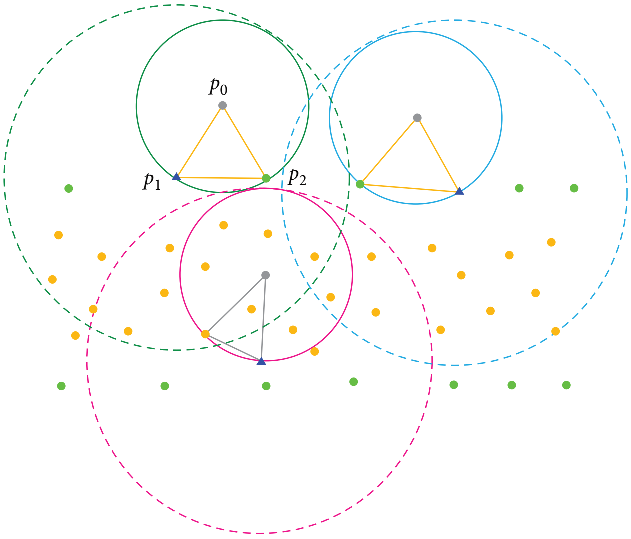


Figure 9: Boundary point judgment diagram.

2.2.3. Segmentation of Bolt Hole Point Cloud Clusters by Euclidean Clustering. After extracting the boundary point cloud using the alpha shape algorithm, Euclidean clustering was used to segment all the clusters in the bolt hole point cloud. In this study, a nearest neighbor search algorithm based on KD-tree is used to divide the space into a multidimensional data structure with multiple disjoint subspaces. Euclidean clustering has three main parameter settings: the cluster search radius *ClusterTolerance*, the minimum value of point cloud clusters *MinClusterSize*, and the maximum value *MaxClusterSize*. The search radius is used to define the extent to which a point searches for its neighboring points during the clustering process and is the most important parameter. An extensive search radius indicates that the neighboring points of each point are searched more extensively, resulting in excessive clustering. The opposite will cause points that originally belonged to the same cluster to be split into multiple clusters. Thus, the settings of the above parameters should be tuned according to the size of the point cloud raw data. The main parameters selected for this study are $ClusterTolerance = 200$, $MinClusterSize = 60$, and $MaxClusterSize = 380$. The main steps are as follows:

Step 1: Find the k nearest points to P by the KD-tree algorithm for any point P in the space.

Step 2: Put points with distances less than the *ClusterTolerance* into Q and remove them from the source point cloud.

Step 3: Repeat step 1 until the points in Q are no longer increasing.

Step 4: Save the point cloud cluster when the number of points in the point cloud cluster is $MinClusterSize < k < MaxClusterSize$. Otherwise, delete the cluster. The workflow is shown in Figure 10.

2.2.4. RANSAC Fitted Bolt Hole Equation. After the clustered segmentation of the bolt hole surfaces, the RANSAC algorithm is used to fit the information of the 3D spatial circle. When a sphere intersects a plane, the intersection line is a circle. Conversely, any circle in space can be represented as the intersection of a sphere with a plane. Therefore, the coordinate equation of a circle in the area is given as follows:

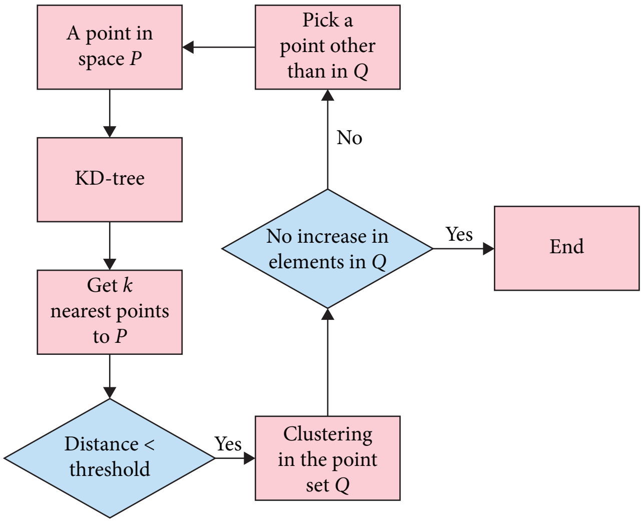

A point in space P
Pick a point other than in Q
No
KD-tree
No increase in elements in Q
Yes
End
Get k nearest points to P
Distance < threshold
Yes
Clustering in the point set Q

FIGURE 10: Euclidean clustering algorithm flowchart.

$$y = \begin{cases} (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2 \\ A_x + B_y + C_z + D = 0(A^2 + B^2 + C^2 \neq 0) \end{cases}. \tag{14}$$

The RANSAC algorithm is characterized by three primary parameters, namely the number of iterations (N), the number of subsets (S), and the threshold (T). N can be calculated by the Monte Carlo type probabilistic approach, as shown in Equation (2). e is the percentage of inliers taken as 50% according to the obtained data by terrestrial laser scanning (TLS), p is a coefficient usually set as 0.99, S is set as 20% of all boundary points, and T is taken as 5 mm in this study. Since the proposed algorithm is robust to selected parameters, N is taken as 1,000 in this study. The fitted bolt holes can be obtained using the RANSAC algorithm, as described above.

3. Accuracy Verification

In this paper, we propose a method to verify the accuracy of the feature points obtained by the above method using virtual standard parts. First, a point cloud model with accurate dimensions is created. Then, the point cloud model is measured using the above method; and finally, the results are compared to determine the accuracy of the feature points obtained by this method.

3.1. Standard Theoretical Point Cloud Model Creation

3.1.1. Solid Model Creation. To make the theoretical type as close as possible to the bars in the actual project, the dimensions of the bars in the existing project were used as a reference. The dimensions of the box model are determined as follows: the outer profile of the section is 1.5 × 2 m, the inner shape is 1.34 × 1.88 m, the total length of the bar is 12 m, the radius of the bolt hole is 0.1 m, the hole center distance is 0.5 m, and the coordinates of the hole center distance from the edge corner point are 0.71 m.

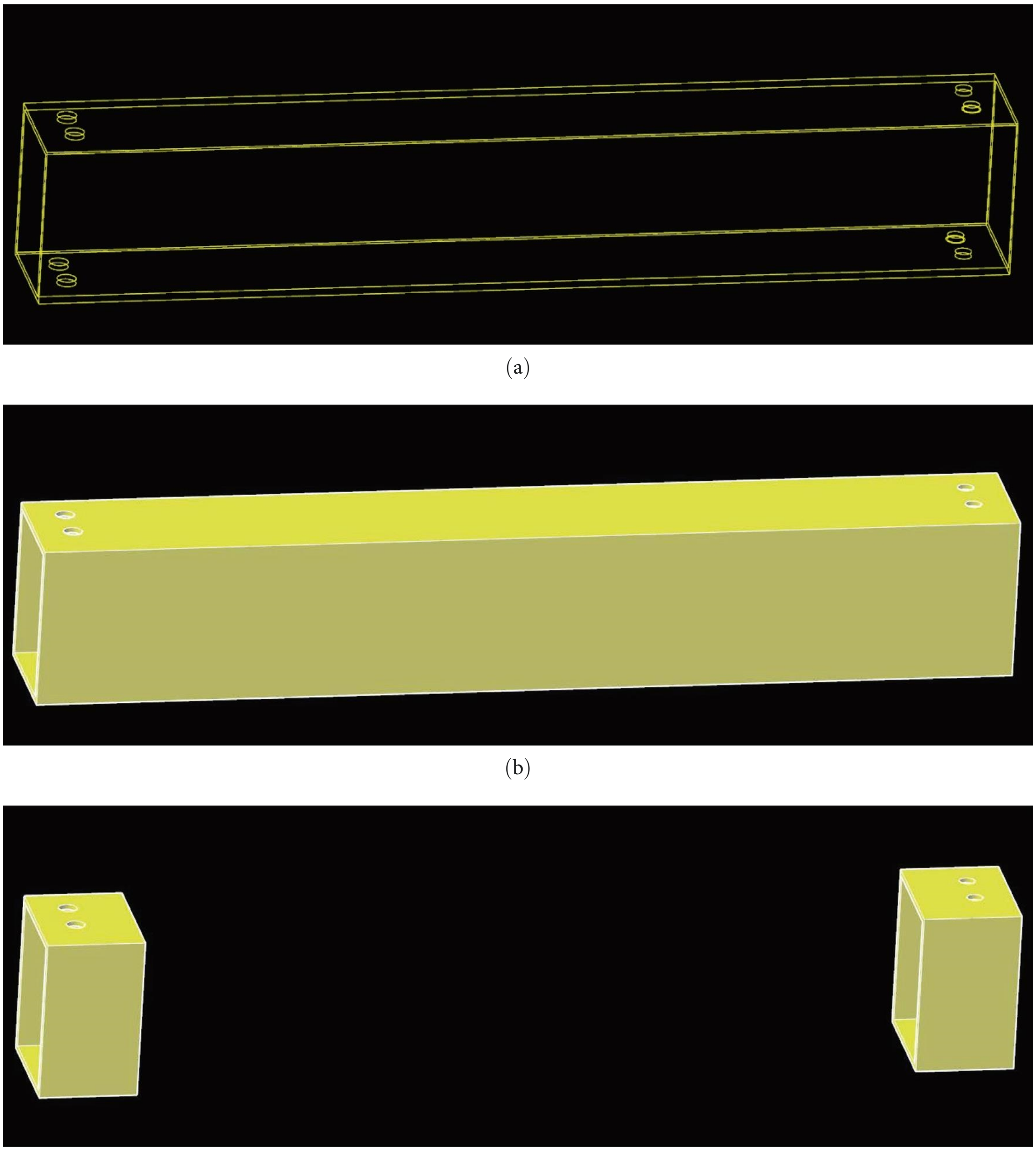
The specific solid model creation process is shown as follows: first, the 3D profile of the theoretical model is created by the designed dimensions, as shown in Figure 11(a); then, the solid model is constructed by lofting the edges and Boolean operation of the bolt holes in the 3D profile, as shown in Figure 11(b); finally, to reduce the computation time, the end point cloud of the bar is extracted by profiling, and the solid model is constructed, as shown in Figure 11(c).

3.1.2. Point Cloud Model Creation. First, the solid model created by CAD (Figure 12) was exported to a file in STL format and imported into CloudCompare software; then, using the "point cloud" function in the software, a point cloud was added to the surface of the solid model, and by setting the density and sampling parameters of the point cloud, the solid model was transformed into a point cloud model with 1 million data points, as shown in Figure 13; finally, Gaussian noise was used to simulate the random noise of the point cloud, and the comparison before and after adding Gaussian noise is shown in Figure 14.

3.2. Boundary and Corner Point Information Extraction Test. To verify the accuracy of the algorithm proposed in this paper, the standard point cloud model is used to perform the edge point extraction test. First, the K-nearest neighbor search algorithm is used for noise reduction of the standard point cloud model; second, the planar cluster segmentation is performed by using the region growing algorithm, and the effect of the end-plane segmentation is shown in Figure 15(a); then, for the obtained planar point cloud clusters, the planar equations are fitted by using the RANSAC algorithm, as shown in Table 1; finally, the planar nonparallel planes are filtered out according to the rank of the matrix of the equation system, and the solution of the equation system is obtained by connective extraction. The coordinates of the boundary points are obtained, and the coordinates of the standard point cloud model boundary points are shown in Table 2. The schematic diagram of edge point extraction is shown in Figure 15(b).

The accuracy of corner point extraction has been evaluated, as indicated in Table 3. This evaluation includes the examination of the four sets of side lengths of the box section and the length of the bars. By comparing the error information of the bars, it can be inferred that the maximum error value is approximately 0.59%. Thus, the corner point extraction technique described in this study demonstrates its capability to meet the engineering requirements in terms of precision.

Three methods were used in this study to fit the data for the above boundaries and corner points. The first method was the one proposed in Section 2.1. The second method used HarrisCorner to extract corner points, and it was implemented directly using keypoints_harris_3d in Open 3D. This method works by sliding the square detection window in the point cloud and determining the corner points based on the number of point clouds in it. The third one is the method of ISS corner point extraction, which is solved using compute_iss_keypoints in Open 3D. This method is a key point detection method based on the curvature and normal vector of the point cloud.



(a)

(b)

(c)

Figure 11: Theoretical model creation: (a) theoretical model outline, (b) theoretical model entity, and (c) theoretical model tip.

Figure 12: Solid model.

Figure 13: Solid model into the point cloud.

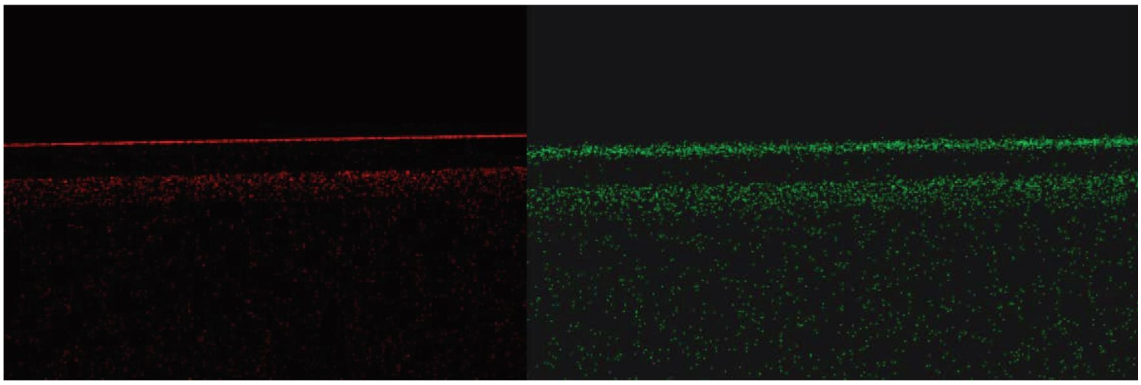
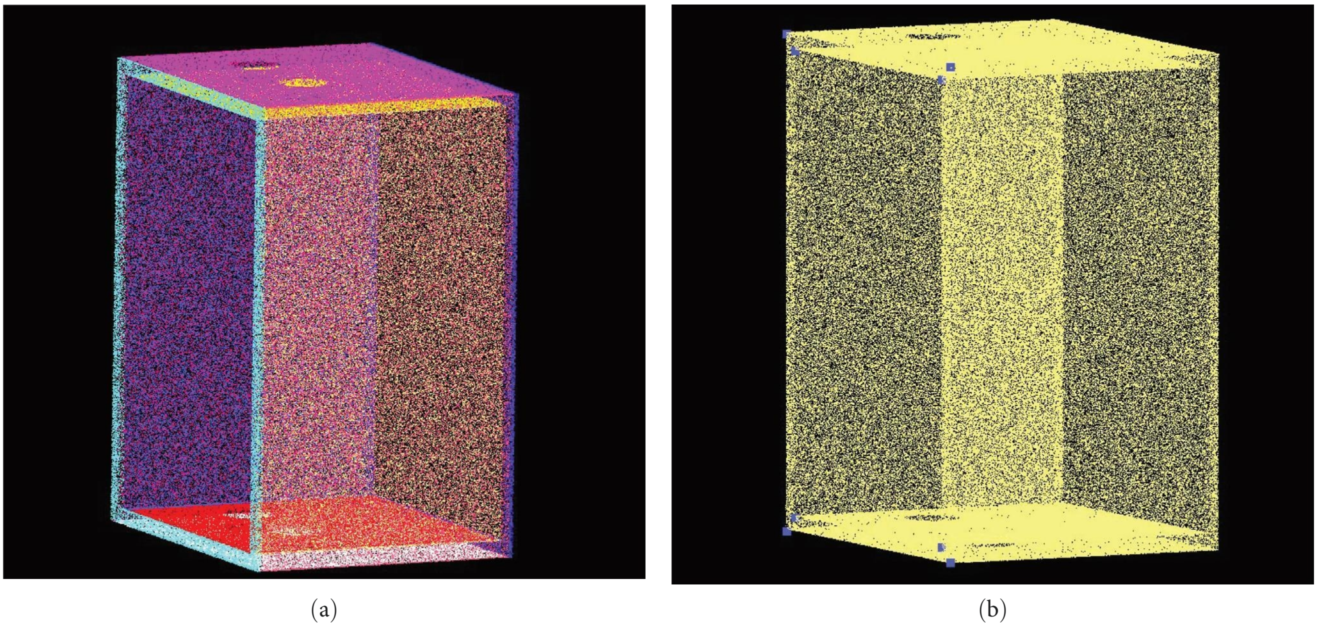


Figure 14: Gaussian noise before and after detail picture.



(a)

(b)

Figure 15: Theoretical model feature information extraction: (a) region growth clustering planar point cloud cluster and (b) corner point feature extraction effect.

Table 1: Plane coefficient parameters.

Plane factor	A	B	C	D
M1	0.000562	−0.000047	1.000004	−2,000.717219
M2	−0.002552	0.000051	1.000010	0.092759
M3	−1.002559	0.000050	0.000011	1,501.006984
M4	−1.002554	0.000058	0.000012	0.751837
M5	−0.000560	0.000051	−0.999994	1,941.318154
M6	0.997442	0.000050	0.000013	−79.668907
M7	−1.000555	0.000049	0.000003	1,420.197920
M8	−0.000557	0.000054	−0.999992	60.027501
M9	−0.000556	−0.999949	0.000010	0.491109

Table 2: Boundary and corner characterization information.

x	y	z	x	y	z
1,497.2	−0.3	1,999.9	1,419.4	−0.3	59.2
0.8	0.5	2,000.7	1,497.8	12,000.1	2,000.4
1,497.2	−0.3	3.7	1.5	12,000.9	2,001.3
0.8	0.5	−0.1	1,497.8	12,000.1	3.1
1,419.4	−0.3	1,940.5	1.4	12,000.9	−0.7
79.8	0.5	1,941.3	1,420.0	12,000.1	1,941.2
79.9	0.4	60.0	79.2	12,000.9	1,941.9
1,497.2	−0.3	1,999.9	79.3	12,000.9	60.6

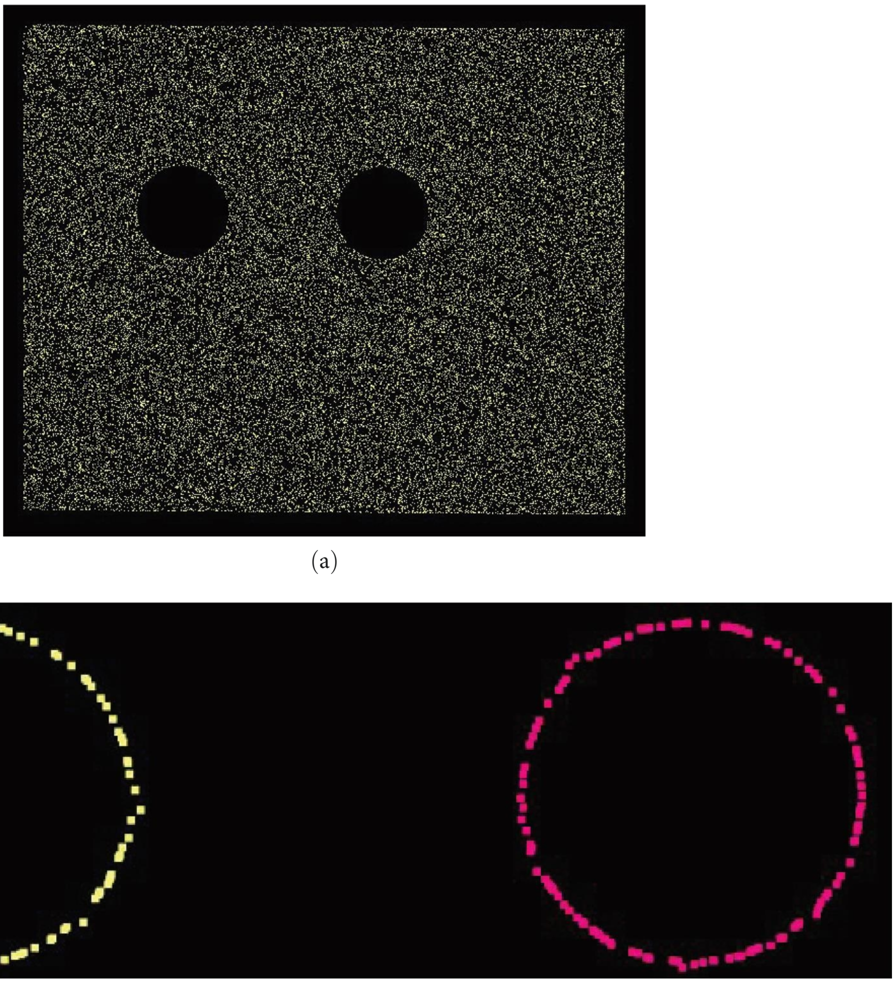
Table 3: Error detection of boundary and corner point feature information.

Projects	Theoretical value (mm)	Measured value (mm)	Error (mm)	Percentage of error (%)
Section side length 1	1,500/2,000	1,496/1,996	4/4	0.27/0.20
Section side length 2	1,340/1,880	1,348/1,883	8/3	0.59/0.16
Section side length 3	1,500/2,000	1,493/1,989	7/11	0.47/0.55
Section side length 4	1,340/1,880	1,340/1,881	0/1	0/0.05
Rod length				
a	12,000	12,000.6	0.6	0.005
b	12,000	12,009.4	0.6	0.005
c	12,000	12,001.0	1.0	0.008
d	12,000	12,000.0	0	0

Table 4: Corner point extraction results of the three algorithms.

Gaussian noise (σ^2)	Our method			Harris corner			ISS		
	Time (s)	Error (mm)	Correctness (%)	Time (s)	Error (mm)	Correctness (%)	Time (s)	Error (mm)	Correctness (%)
0.1	104.46	2.5	100	159.61	2.5	84	219.38	3.2	76
0.2	109.59	2.8	100	168.67	3.6	64	237.81	3.4	73
0.3	121.69	2.9	100	177.42	4.7	53	244.57	5.3	57

It can be seen in Table 4 that the error of our method is the smallest regardless of the data noise, so it is the optimal solution for noisy data. However, for the data containing obvious outliers, the results extracted by Harris Corner and ISS deviate from the actual values. Harris Corner identifies the noise points as corner points and is less robust to noise disturbances. The ISS algorithm requires establishing a local coordinate system and modeling within this coordinate system, which results in a longer execution time. Our method achieves better results for data with the aforementioned three types of noise and saves 44.0% in extraction speed compared to traditional algorithms when there are outliers in the data.



(a)

(b)

Figure 16: Bolt hole clustering: (a) plane projection effect and (b) TLS bolt hole clustering.

Table 5: Bolt hole extraction accuracy.

Bolt hole radius (mm)		Error	Hole center distance (mm)		Error
Theoretical value	Measured value		Theoretical value	Measured value	
100	100.6	0.60%	300	497.8	0.7%

Compare the feature point index extracted by each algorithm with the reference feature point index. If the indexes are consistent, the feature point is extracted correctly. Use the proportion of correctly extracted indexes to the total number of feature points as the correctness. As shown in Table 4, our method extracts fewer feature points with higher correctness compared to other feature point extraction algorithms. This indicates that the corner points extracted by our method have stronger feature representations. The correctness of the Harris and ISS algorithms decreases significantly with the addition of Gaussian noise as they do not have the step of removing the boundary response of corner points, resulting in a large number of false corner points. Our method obtains corner points by associating three plane equations, which provides an accurate and fast method of extracting corner points for steel truss point clouds.

3.3. Bolt Hole Information Extraction Test. To increase the robustness of the bolt hole extraction algorithm, the same side plane clusters containing bolt holes are projected onto the plane equations on the outside of the end, and a set of bolt hole information is analyzed in detail below. The algorithm parameters are set according to the characteristics of the structure, the radius of the α circle in the boundary extraction is set to 150 mm, and the radius of the Euclidean clusters is set to 400 mm. Figure 16(a) shows the effect of planar projection of the point cloud data. The results of fitting the bolt hole feature information using the RANSAC algorithm are shown in Table 5. The mean value represents the radius of the bolt holes.

From the above results, it can be seen that for the obtained point cloud, the Euclidean clustering algorithm based on

Table 6: The results of the three algorithms for bolt hole feature extraction.

	Time (s)	Radius error (mm)	Hole center distance error (mm)
Our method	23.61	0.60	2.17
DBSCAN	—	—	—
Angle criterion	31.85	1.53	3.96

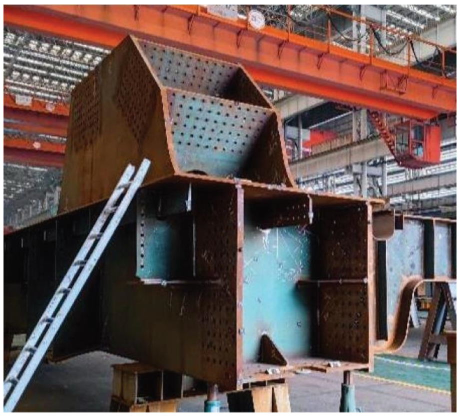


Figure 17: Site view of the lower chord bar.

boundary extraction has an error of 0.73% after being adjusted by the random sampling consistency algorithm; it meets the requirements in terms of accuracy.

The accuracy of boundary extraction determines the precision of bolt hole fitting. Three methods were employed to extract the point cloud of the bolt hole boundary. The first method was the one proposed in Section 2.2. The second method used the DBSCAN point cloud clustering segmentation algorithm. This method finds clusters of any shape in a noisy point cloud, and it was implemented directly using the pcb.cluster dbscan function in Open 3D. The third method is to detect circular hole boundary points based on the distribution of kNN points for the selected points, where angle, half disk, and shape criteria are commonly used. The pcd compute boundary points function in Open 3D is used and solved based on the angle criterion.

It can be seen from Table 6 that the Euclidean clustering algorithm based on boundary extraction is the best in both computational efficiency and extraction accuracy. However, the density clustering algorithm fails to extract the bolt holes due to the uniform distribution of the point cloud obtained by TLS near the bolt holes. For data containing significant outliers, the relative error of extracting bolt hole edges using only the angle criterion is significant. The efficiency of bolt hole feature extraction is enhanced by 25.9% compared to the traditional method. From the aforementioned results, it can be concluded that our method can quickly compute the parameters of bolt holes while being insensitive to noise.

4. Examples of Projects

The appearance of a single lower chord bar of a steel truss structure is shown in Figure 17, with the following dimensions: total length of 12.18 m, cross-section outer contour length of 1.3 m, width of 1.42 m, and inner profile length of 1.20 m, and width of 1.35 m. The distance between the centers of the bolt holes is 100 mm, the minimum value of the bolt holes from the edges and corners is 160 mm, and the bolt holes have a radius of 16.7 mm. There are two groups of holes on a single surface, and each group of holes on the side surface contains 42 bolt holes, and each group of holes on the top surface contains 32 bolt holes, for a total of 464 bolt holes. There are two groups of holes on one side, and each group includes 42 side bolt holes, and each group of holes on the top side contains 32 bolt holes, for a total of 464 bolt holes.

4.1. Point Cloud Data Preprocessing.

The bar point cloud contains some noise, and the outlier noise points can be directly removed manually, and then the K-nearest neighbor search algorithm is used to remove the interfering points. As shown in Figures 18 and 19, the original point cloud data volume is 1,228,019, and the point cloud after preprocessing is 1,225,025.

4.2. Boundary and Corner Point Information Extraction.

After the preprocessing of the point cloud, the data volume is still over 1 million. Since the final data have a large Euclidean distance between them, to increase the processing speed of the algorithm, the last data of the point cloud are first segmented by Euclidean clustering. The clustering radius is set to 5 m according to the design size of the lower chord rod, and the point cloud cluster threshold is set to 10,000–40,000. The clustering results are shown in Figure 20(a), and different colors indicate the different point cloud clusters.

Based on the Euclidean clustering segmentation, a planar clustering segmentation based on region growth is

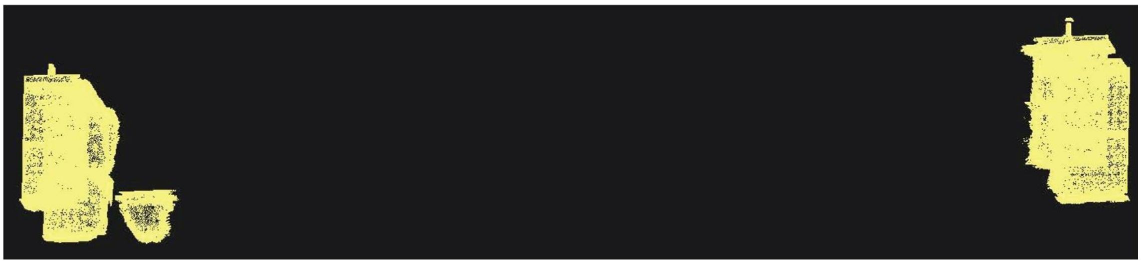


Figure 18: Rod raw point cloud.

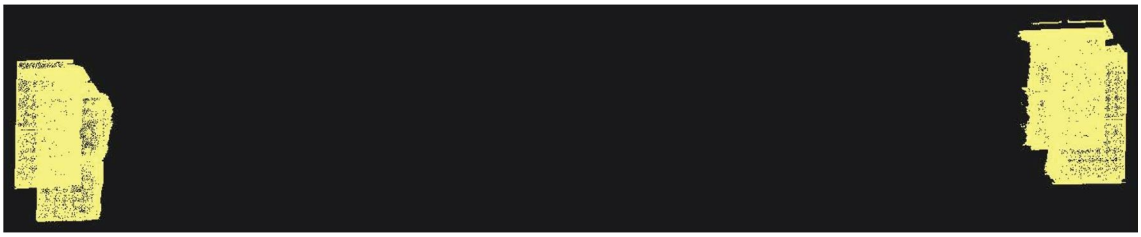
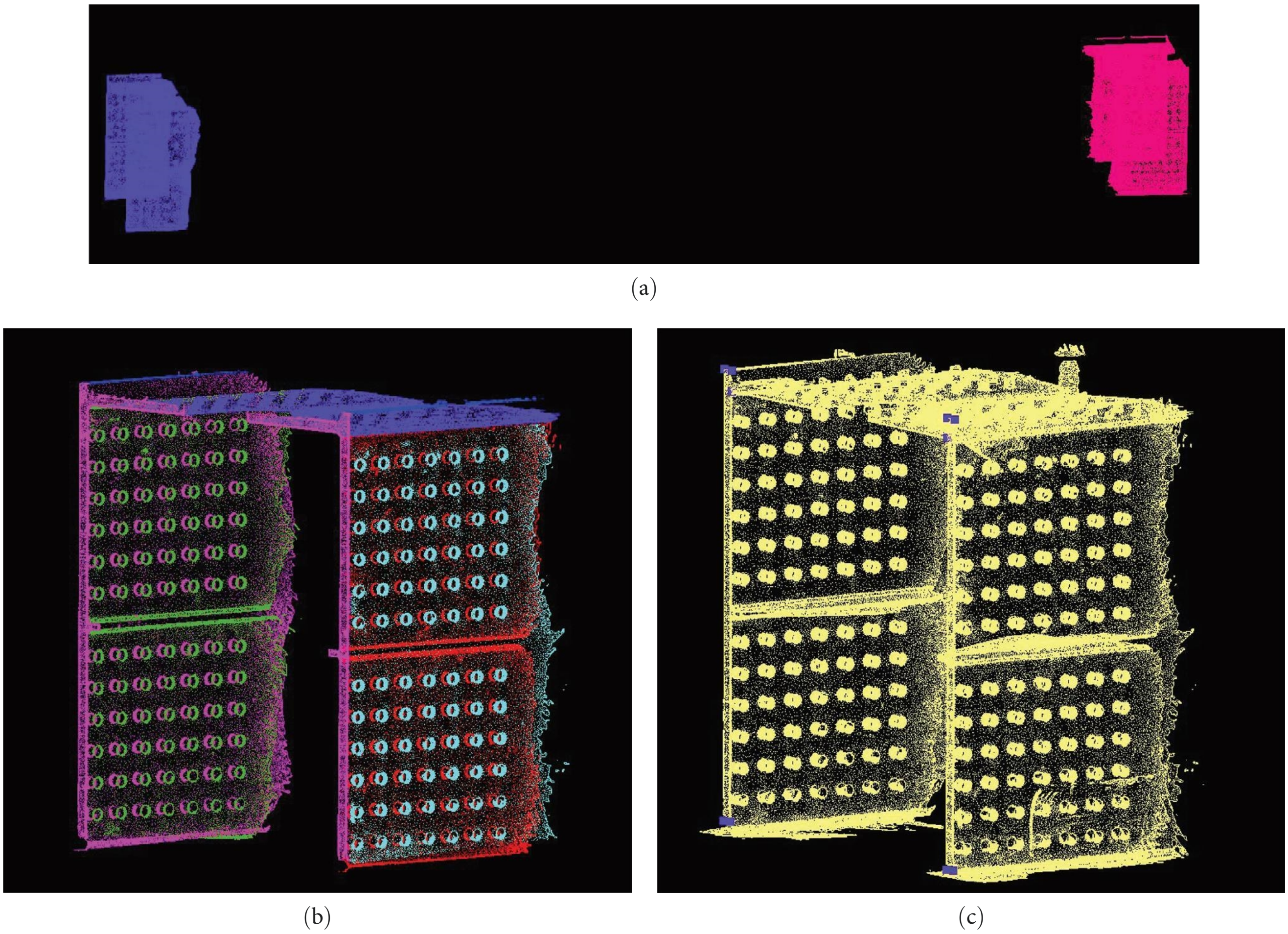


Figure 19: Point cloud data after rod preprocessing.



(a)

(b)

(c)

Figure 20: Boundary and corner point feature extraction: (a) rod end data segmentation, (b) T1 point cloud plane segmentation, and (c) T1 corner point extraction.

Table 7: T1 plane equation coefficient.

Point cloud	Plane name	A	B	C	D
T1	M1	−0.9886	0.1450	−0.0156	8.3954
	M2	−0.0028	0.0110	0.9999	−162.5580
	M3	−0.4100	0.9097	−0.0654	18.0125
	M4	0.9983	0.0579	0.0005	−4.7387
	M5	−0.0054	0.0107	−0.9999	162.6840
	M6	−0.9978	−0.0667	−0.0009	4.0148
	M7	0.9978	0.0668	0.0022	−4.2687
	M8	−0.0005	−0.0099	−0.9995	161.4950

Table 8: Coordinates of plane intersection points.

Point cloud	Corner point	x	y	z
T1	P1	5.0442	−5.8329	162.6460
	P2	5.0100	−5.8484	162.6460
	P3	5.0101	−5.8512	162.6060
	P4	4.2961	−6.1700	162.6480
	P5	4.3261	−6.1565	162.6480
	P6	4.3264	−6.1593	162.6070
	P7	5.0498	−5.9076	161.5730
	P8	5.0149	−5.9243	161.5590
	P9	5.0442	−5.8329	162.6460
	P10	5.0010	−5.8484	162.6460

performed for each point cloud cluster to adjust all the planar clusters. One of the final point cloud data, T1, is used for detailed illustration, as shown in Figure 20(b). The number of last point clouds is 612,544, eight planar point cloud clusters are coclustered, and different colors distinguish different planar point cloud clusters.

The plane equations are fitted to the extracted planar point cloud clusters using the RANSAC algorithm, and the coefficients of the plane equations are shown in Table 7. The plane equations are arranged and combined in groups of three. The rank $r(A) = 3$ of the linear equation system is used as the screening condition to obtain the plane equation system that can intersect at one point. The simultaneous solution obtains the characteristic information of the corner points. The coordinates of the corner points are shown in Table 8, and Figure 20(c) shows the extraction effect of the T1 corner points.

4.3. Bolt Hole Information Extraction. For the planar point cloud clusters segmented by regional growth clustering, all planar point clouds containing bolt holes are first extracted. The height error is negligible since the bolt holes are all made of standard parts. To improve the accuracy of the feature information extraction, as shown in Figure 21(a), the neighboring planar clusters containing bolt holes are all projected to the outer planar equations, and then the projected point clouds are merged to form a point cloud cluster. Then, using the density-based DBSCAN clustering segmentation, the neighborhood radius is set to 3 mm, the minimum number of points is set to 10, and the threshold of the number of point cloud clusters is set to 50–1,000 to remove the boundary point cloud. The effect of the bolt hole surface extraction is shown in Figure 21(b). The feature information of the bolt hole is fitted to each point cloud cluster using the RANSAC algorithm. This includes the location of the bolt hole center and the corresponding bolt hole radius. Figure 21(c) shows the spatial location of the bolt hole center, and Figure 21(d) shows its relative position with respect to the original point cloud.

4.4. Dimensional Accuracy Test. The results of all the feature information extraction for the bar are shown in Figure 22, where the yellow point cloud is the original point cloud, the red point cloud is the center of the bolt hole, and the blue point cloud is the corner point. The feature information of the bolt is the distance calculated to compare the theoretical size results of the bolt, as shown in Table 9, and the average value is used to show the measured values of the same type. By comparing the dimensional information of the actual measured data of the bars with the design data, the processing error of the structure is given visual feedback. According to the relevant contents of the technical specification for highway bridge construction (JTG/T 3650-2020), the manufacturing error of the bar is within the permissible range of the specification.

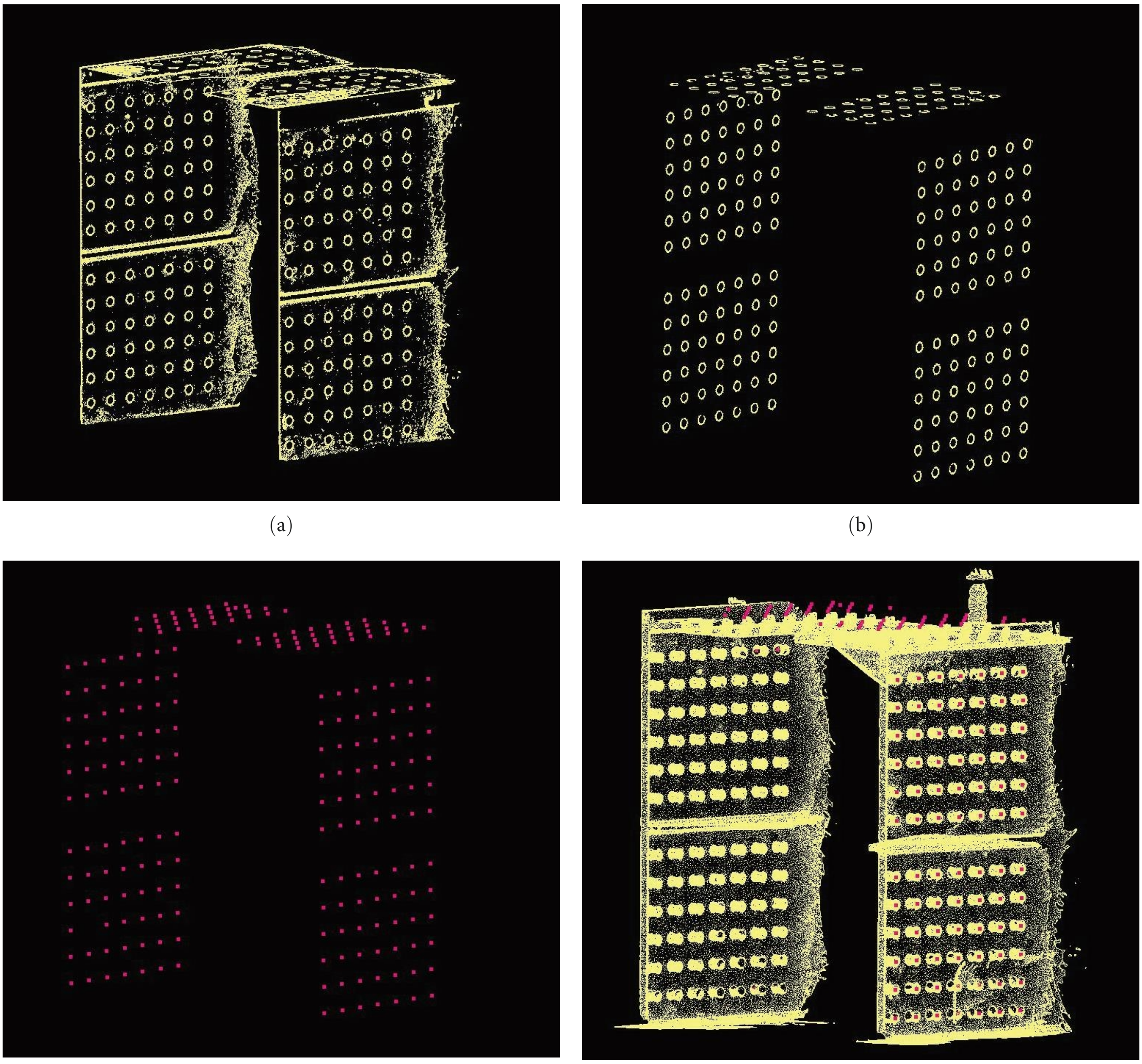


Figure 21: Bolt hole feature extraction: (a) T1 plane projection, (b) T1 bolt hole clustering, (c) T1 bolt hole center, and (d) T1 hole center relative position.

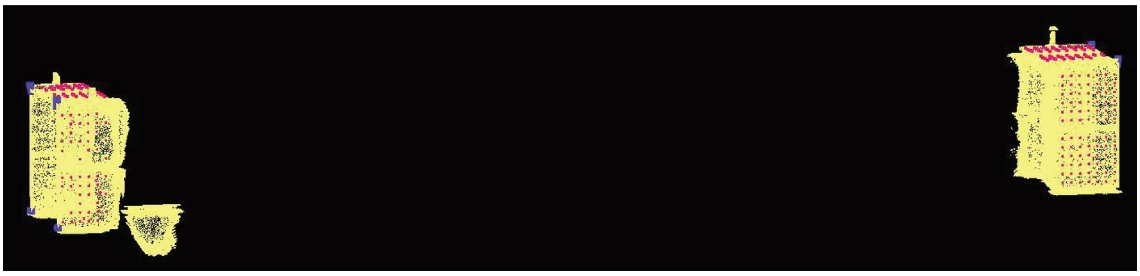


Figure 22: The effect of rod feature information extraction.

Table 9: Comparison of rod size information.

Projects	Theoretical size (m)	Measured size (m)	Error (mm)	Error (%)
Rod length	12.18	12.176	0.02	0.164
Outer profile size	1.3 × 1.42	1.298 × 1.415	2 × 5	0.505
Inner profile size	1.2 × 1.35	1.205 × 1.337	5 × 1.3	0.550
Hole center distance	0.1	0.099	1	1
Distance between hole center and corner point	0.16	0.157	3	1.875
Bolt hole radius	0.0167	0.01675	0.05	0.299

5. Conclusion

To address the problems of poor feature information extraction accuracy, large subjective factors, and low automation degree of large steel joist structures, this paper proposed an automatic extraction method of key feature information such as boundary points, corner points, and bolt holes by nested combination of classical algorithms for the engineering needs of reverse modeling and virtual trial assembly. Based on this study, the following findings were obtained:

(1) The feature point extraction algorithm based on the region growing algorithm, the clustering algorithm, and RANSAC can accurately extract the boundary and corner points of large bolted steel members. The experimental results show that the proposed method is robust to noise interference, the algorithm has a high correct extraction rate, and it saves 44.0% of the computation time compared with the traditional way.

(2) The bolt hole parameter calculation method based on classical algorithms such as alpha shapes and Euclidean clustering segmentation can accurately fit the circle center coordinates and radius parameters. The average errors of the calculated circular hole radius and hole center distance are 0.6 and 2.17 mm, respectively, which improves the extraction efficiency by 25.9% compared to the traditional algorithms.

(3) The new method of verifying the extraction accuracy of the algorithm by establishing a standard point cloud model effectively avoids the influence of factors such as manufacturing error and observation error on the accuracy verification. In addition, this method can add Gaussian noise to the standard point cloud model to verify the robustness of the algorithm against noise interference.

(4) The proposed automatic feature batch extraction method for bolted steel trusses shows good accuracy in practical engineering, which can provide accurate data support for virtual trial assembly, reverse modeling, and dimensional inspection.

Data Availability

Data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest for the publication of this research.

Acknowledgments

This research was supported by the Open Projects Foundation (no. BHSKL21-07-GF) of the State Key Laboratory for Health and Safety of Bridge Structures.

References

[1] G. Cheng, J. Liu, N. Cui, H. Hu, C. Xu, and J. Tang, "Virtual trial assembly of large steel members with bolted connections based on point cloud data," *Automation in Construction*, vol. 151, Article ID 104866, 2023.

[2] L. Ma, C. Zhang, Y. Fu, and D. Ma, "Application of point cloud segmentation algorithm in high-precision virtual assembly," in *2019 Chinese Control And Decision Conference (CCDC)*, pp. 1003–1007, IEEE, Nanchang, China, 2019.

[3] X. Zhu, H. Manamasa, J. L. J. Sánchez, A. Maki, and L. Hanson, "Automatic assembly quality inspection based on an unsupervised point cloud domain adaptation model," *Procedia CIRP*, vol. 104, pp. 1801–1806, 2021.

[4] H. Zou, R. Xia, Y. Zeng et al., "Ovality measurement based on scanning point cloud for tube bend deformation analysis," *The International Journal of Advanced Manufacturing Technology*, vol. 125, pp. 3743–3760, 2023.

[5] D. Lamas, A. Justo, M. Soilán, M. Cabaleiro, and B. Riveiro, "Instance and semantic segmentation of point clouds of large metallic truss bridges," *Automation in Construction*, vol. 151, Article ID 104865, 2023.

[6] C. Fotsing, N. Menadjou, and C. Bobda, "Iterative closest point for accurate plane detection in unorganized point clouds," *Automation in Construction*, vol. 125, Article ID 103610, 2021.

[7] L. Truong-Hong and R. Lindenbergh, "Automatically extracting surfaces of reinforced concrete bridges from terrestrial laser scanning point clouds," *Automation in Construction*, vol. 135, Article ID 104127, 2022.

[8] M. Kim, D. Lee, T. Kim, S. Oh, and H. Cho, "Automated extraction of geometric primitives with solid lines from unstructured point clouds for creating digital buildings models," *Automation in Construction*, vol. 145, Article ID 104642, 2023.

[9] L. Truong-Hong, D. F. Laefer, T. Hinks, and H. Carr, "Combining an angle criterion with voxelization and the flying voxel method in reconstructing building models from LiDAR data," *Computer-Aided Civil and Infrastructure Engineering*, vol. 28, no. 2, pp. 112–129, 2013.

[10] X. Zhou, J. Liu, G. Cheng, D. Li, and Y. F. Chen, "Automated locating of replaceable coupling steel beam using terrestrial laser scanning," *Automation in Construction*, vol. 122, Article ID 103468, 2021.

[11] R. A. Tabib, Y. V. Jadhav, S. Tegginkeri et al., “Learning-based hole detection in 3D point cloud towards hole filling,” *Procedia Computer Science*, vol. 171, pp. 475–482, 2020.

[12] Z. Cao, D. Chen, Y. Shi et al., “A flexible architecture for extracting metro tunnel cross sections from terrestrial laser scanning point clouds,” *Remote Sensing*, vol. 11, no. 3, Article ID 297, 2019.

[13] W. Wang, K. Lu, R. Hong, P. Chen, J. Zhang, and B. Wang, “A machine vision method for automatic circular parts detection based on optimization algorithm,” in *Intelligent Computing Theories and Application*, D.-S. Huang, V. Bevilacqua, P. Premaratne, and P. Gupta, Eds., vol. 10361 of *Lecture Notes in Computer Science*, pp. 600–611, Springer, Cham, 2017.

[14] M. Michałowska, J. Rapiński, and J. Janicka, “Tree position estimation from TLS data using hough transform and robust least-squares circle fitting,” *Remote Sensing Applications: Society and Environment*, vol. 29, Article ID 100863, 2022.

[15] Z. Xu, Y. Liang, H. Lu, W. Kong, and G. Wu, “An approach for monitoring prefabricated building construction based on feature extraction and point cloud segmentation,” *Engineering, Construction and Architectural Management*, vol. 30, no. 10, pp. 5302–5332, 2022.

[16] Q. Wang, R. Huang, X. Yan, and T. Cheng, “Feature point extraction of scattered point cloud based on multiple criterions,” *Application Research of Computers*, vol. 36, no. 5, pp. 1585–1588, 2019.

[17] M. Cabaleiro, B. Riveiro, P. Arias, J. C. Caamaño, and J. A. Vilán, “Automatic 3D modelling of metal frame connections from LiDAR data for structural engineering purposes,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 96, no. 1, pp. 47–56, 2014.

[18] Q. Wang, H. Sohn, and J. C. P. Cheng, “Automatic as-built BIM creation of precast concrete bridge deck panels using laser scan data,” *Journal of Computing in Civil Engineering*, vol. 32, no. 3, 2018.

[19] J. Du, Z. Jiang, S. Huang et al., “Point cloud semantic segmentation network based on multi-scale feature fusion,” *Sensors*, vol. 21, no. 5, Article ID 1625, 2021.

[20] Y. Yang, H. Fang, Y. Fang, and S. Shi, “Three-dimensional point cloud data subtle feature extraction algorithm for laser scanning measurement of large-scale irregular surface in reverse engineering,” *Measurement*, vol. 151, Article ID 107220, 2020.