

Research Article

Automatic Identification and Segmentation of Long-Span Rail-and-Road Cable-Stayed Bridges Using UAV LiDAR Point Cloud

Yueqian Shen ¹, Zili Deng ¹, Jinguo Wang ¹, Shihan Fu ¹ and Dong Chen ²

¹School of Earth Sciences and Engineering, Hohai University, Nanjing 210098, China

²College of Civil Engineering, Nanjing Forestry University, Nanjing 210037, China

Correspondence should be addressed to Dong Chen; chendong@njfu.edu.cn

Received 29 April 2024; Revised 4 September 2024; Accepted 16 October 2024

Academic Editor: Zhen Sun

Copyright © 2024 Yueqian Shen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Bridge information models are essential for bridge inspection, assessment, and management. LiDAR technology, particularly UAV LiDAR, offers a cost-effective means to capture dense and accurate 3D coordinates of a bridge's surface. However, the structure of large-scale bridges is complex, and existing commercial software still demands substantial manual effort to segment the components when constructing bridge information models for large-scale bridges. This study introduces a novel approach to automatically segment the components of a long-span rail-and-road cable-stayed bridge from the entire point cloud obtained through UAV LiDAR. In this proposed approach, the geometric and topological constraints of various bridge components are thoroughly examined, and a combination of the coarse-to-fine concept and top-down strategy is employed. The key structural elements, including piers, cable towers, wind fairing plate, stay-cable, main truss, railway surfaces, and deck surfaces, are identified and segmented. The proposed methodology achieves an average accuracy of over 96% at the point level validated using datasets acquired by UAV LiDAR.

Keywords: coarse-to-fine; long-span rail-and-road cable-stayed bridge; point cloud; segmentation; UAV LiDAR

1. Introduction

Bridges, being one of the most important types of infrastructures, play a vital role in urban traffic, particularly in the areas covered with vast watersheds. By the end of 2023, the number of large-scale bridges under construction and in service in China is over 10,000 [1]. To ensure the safety of in-service bridges, periodic inspection, assessment, and documentation are essential. Consequently, there is a significant demand for accurate segmentation of bridge components. This segmentation is crucial for detecting structural changes, inspecting local surface damage, and verifying the stability of bridges to assess their capacity for additional loads, such as maximum load-bearing capacity and maximum wind force it can withstand.

Light detection and range (LiDAR) technology has made significant advancements in recent years, possessing the capability to accurately and efficiently capture the geometric

properties of object surfaces. This technology offers a complementary and alternative approach to address these challenges.

Compared to traditional measurement methods such as total stations and leveling, LiDAR technology offers distinct advantages, including noncontact operation, high resolution, precise accuracy, and rapid data acquisition. The point cloud acquired by LiDAR has the capability of capturing geometric information of object surface, including component size or shape, and structural dimensions. With the above wealth of information, it becomes possible to construct bridge information modeling (BrIM) which provides a shared knowledge resource to support a reliable basis for decision-making during a bridge's life cycle. However, utilizing the point cloud for the automatic generation of BrIM presents a significant challenge, and several issues need to be addressed. Generating BrIM from point cloud data typically involves two steps: (1) object detection in the entire

bridge point cloud and (2) geometric model generation. In the construction industry, three types of LiDAR systems, i.e., terrestrial LiDAR, mobile LiDAR, and UAV-based LiDAR, are commonly used. The point cloud gathered by the LiDAR is generally defined by x -, y -, and z -coordinates, associated with attributions such as intensity or RGB values. This encoding captures the geometric properties of various objects which are widely used in three-dimensional modeling for elements like walls, trees [2], bridges [3, 4], doors [5], facades [6], roofs [7], and street lamps [8].

However, it is required to perform a series of processes before obtaining a useful 3D model. Among these processes, the recognition and segmentation of each structural component are essential intermediate steps. Point cloud segmentation, which involves grouping 3D points into multiple homogeneous groups with common characteristics, is typically necessary before recognition. Primitive features employed for segmentation often include Euclidean distance, density, normal vector, roughness, and others. Numerous studies have proposed methodologies and algorithms to detect and recognize components in various types of infrastructures, such as roads and railways [9] and urban environment [10].

Numerous research groups have attempted to develop automatic or semiautomatic methods for processing point clouds for various applications. However, most of these approaches predominantly focus on specific types of bridges, such as masonry bridges, while there is less emphasis on the diverse range of highway bridges, which often include various subtypes. In contrast, long-span rail-and-road cable-stayed bridges present a more intricate challenge due to the complex interactions between numerous structural elements, including their varied sizes, shapes, orientations, and materials, along with the additional demands of accommodating both rail and road traffic. These complexities significantly heighten the challenges associated with segmentation and subsequent processing. As a result, efficient methods for the straightforward processing of extensive point cloud data in such scenarios remain scarce.

Therefore, the primary motivation of this work is to develop a comprehensive and accurate algorithm for segmenting long-span rail-and-road cable-stayed bridges into individual components using UAV point cloud data. The segmentation results can be utilized to generate a BrIM. Additionally, this research aims to offer a method for efficiently handling large point clouds of large-scale bridges, which can further contribute to structural health monitoring. The remainder of this study is organized as follows: Section 2 summarizes related work on the current topic. We provide a detailed description of the proposed methodology in Section 3. Section 4 presents the experimental outcomes of the case study along with an analysis of their accuracy. Finally, Section 5 concludes this study along with a few suggestions for future research topics.

2. Literature Review

The existing methods can be categorized into four approaches: geometric features-based methods, machine

learning-based methods, deep learning-based methods, and hybrid methods.

2.1. Geometric Feature-Based Methods. As is well-known, geometric features-based methods utilize the geometric characteristics of points to achieve the extraction of components. Such methods can be categorized into two approaches: top-down and bottom-up.

The idea of the bottom-up approach is to perform segmentation and classification in sequence. More specifically, primitive features are utilized to generate a top-level system which can be surface normals, meshes, patches, etc. Notably, a prerequisite understanding of the overall scene context is not necessary. Instead, subsets of data sharing similar local features are grouped to define individual object instances. In contrast, data subsets displaying comparable local attributes are later clustered to delineate distinct object instances. A method based on range growing was proposed to group points by considering surface roughness and principal curvatures around each point. This approach is utilized to segment cylindrical pipelines from flat architectural surfaces, as described in the work by [11]. Another application of the range growing method, combined with quadratic surface fitting, is found in [12]. This method involves choosing a seed point and progressively adding neighboring points with similar features to the current group until reaching the edge. However, it is worth noting a limitation identified in the literature: This method may fail to detect the edge between a pier cap and a pier in a small portion of a bridge point cloud, and it is required to select the key points manually. Laefer and Linh [13] utilized kernel density estimation to establish the shape and dimensions of cross sections and automatically identify structural members from point clouds collected by a terrestrial LiDAR. Additionally, Xu et al. [14] employed an octree-based voxel structure to implement a probabilistic segmentation model for construction sites. It is important to note that the accuracy of segmentation in this model is highly sensitive to the voxel size.

In contrast to the bottom-up approach, the top-down approach relies on domain knowledge of scenes to identify elements within structures. This method typically employs mathematical models with various parameters to describe the geometrical and topological constraints of each component. For instance, Lafarge et al. [15] introduced a parametric model that encodes the geometries and planimetric arrangements of building roofs. The similarity of this model to the original point cloud is measured using a Bayesian model. Another approach involves a graph-based method, as presented by [4], which encodes the geometries and topologies of primitives from a point cloud. These graphs are then matched with a predefined object library to recognize objects. Lu et al. [16] proposed a sliding algorithm to separate beam and slab components from pier components. However, it is important to note that the segmentation of the scene point cloud around bridges is not explicitly considered in this algorithm.

2.2. Machine Learning–Based Methods. Machine learning techniques find applicability in diverse facets of point cloud processing, encompassing tasks such as classification, segmentation, and feature extraction. Among these methods, support vector machine (SVM) stands out as a prominent choice for point cloud processing, characterized by its absence of assumptions about pre-existing parameters and the model within the dataset. The SVM algorithm creates an optimized model using training data to predict classes in a test set, excelling at high-dimensional classification by finding a separating hyperplane in feature space. While efficient in unstructured datasets, SVM may face limitations in large and noisy datasets, making it a suitable choice for 3D classification tasks in construction and infrastructure point clouds. Chen et al. [17] utilized SVM for segmenting and extracting objects from virtual environments. Xia et al. [18] introduced an automated approach for the semantic segmentation of bridge point cloud data, incorporating local descriptors and machine learning methodologies into their framework.

2.3. Deep Learning–Based Methods. In the past decade, deep learning has undergone rapid development, with scholars exploring its applications in point cloud data processing. Deep learning–based methods in this context refer to models that operate directly on raw 3D point clouds, relying solely on learning without the need for transcendent knowledge of geometries or design standards. Notably, these models do not require a priori information about the geometric characteristics or design standards. Recent years have seen a surge in works focusing on segmentation tasks utilizing neural networks, especially deep architectures such as convolutional neural networks (CNNs) and graph neural networks (GNNs). These methods have gained considerable popularity for their capacity to learn intricate patterns and hierarchical representations directly from raw point cloud inputs, resulting in enhanced performance across various tasks including object detection, segmentation, and scene understanding. The key advantage lies in their ability to automatically extract meaningful information without explicit geometric or design specifications. Broadly, deep learning methods for point cloud processing can be categorized into three main types: 3D convolution on voxels [19], multiview projection onto images or planes followed by 2D convolution [20], and PointNet [21, 22]. In a study by Kim et al. [23], the PointNet architecture was successfully employed to detect piers and decks in full-scale reinforced concrete (RC) bridges from point cloud data. Several other deep learning models have emerged in the field, including PointCNN [24] and Dynamic Graph Convolutional Neural Network (DGCNN) [25]. In a study conducted by Kim [26], a performance comparison was undertaken for deep learning models, specifically PointCNN, DGCNN, and PointNet, focusing on the task of semantic segmentation of bridge point clouds. The comparison involved training and testing these deep learning models on the same set of bridges. Additionally, a graph-based hierarchical DGCNN model was proposed for the semantic segmentation of bridge

components [27]. However, the methodology employed the division of each bridge into two halves for training and testing. It is important to note that this approach raised concerns about potential controversy in the results due to the high similarity between the training and testing data, potentially leading to an overestimation of model performance.

Moreover, several researchers have attempted to develop end-to-end deep neural networks designed for the extraction of bridge components. These networks are adept at capturing critical geometric and topological information of bridge structures, enabling precise extraction of various bridge elements. This advancement opens new avenues for applications such as bridge inspection, maintenance, and structural health monitoring. For instance, Jing et al. [28] proposed a novel network model named BridgeNet, which incorporates channels for extracting geometric parameters of masonry bridges, thereby facilitating the segmentation of large-scale bridge point cloud components. Building upon this, Jing et al. [29] introduced an enhanced deep learning architecture in 2024, named BridgeNetv2. This updated model integrates transformer modules into the original BridgeNet framework and demonstrated superior performance in tests compared to its predecessor from 2022. Additionally, Yang et al. [30] proposed a weighted superpoint graph (WSPG) method for extracting highway bridge components. This method initially clusters point clouds into hundreds of semantically homogeneous superpoints and then classifies them into various bridge components using PointNet and GNNs. However, these studies primarily focus on specific types of bridges, such as masonry bridges, and research on the extraction of components from long-span rail-and-road cable-stayed bridge remains limited.

In summary, the study emphasizes that using end-to-end deep learning methods for large-scale bridge analysis is constrained primarily by two factors: the size of the training dataset and the available computing power. These limitations reveal the difficulties in applying deep learning to real-world, complex structures such as bridges.

2.4. Hybrid Methods. To achieve satisfactory results in scenarios with constraints like limited training data and computational resources, one strategy is to utilize a hybrid method that combines manually defined features and machine learning models to achieve classification. The hybrid method first involves extracting high-level manually defined features or generating geometric primitives from the input data. After the feature extraction, machine learning models—comprising logistic regression, Naive Bayes, decision trees, SVMs, random forests, and neural networks—can be trained on the extracted features for component classification. This hybrid approach facilitates a more efficient utilization of available data and computational resources, particularly in situations where the application of deep learning may be less feasible or practical. Xiong et al. [31] employed a region-growing algorithm to establish connections between points with similar surface normals, generating planar patches from voxelized point data. They

identified the four nearest neighbors for each patch using the Euclidean distance metric, allowing the calculation of pre-defined contextual information. Local features of each patch, including orientation, shape, size, point density, and height, were defined. This approach provided a comprehensive characterization of patches for subsequent analysis. A stacked learning model [32] was trained to classify planar surfaces into categories like walls, ceilings, floors, and clutters. While robust to occlusions in indoor environments, it could only generate planar surface models and could not identify structural components within curved surfaces. In a separate study, Koppula et al. [33] used image-based and point cloud features from RGB-D sensors to train learning algorithms for detecting elements like floors, walls, and nonstructural objects. Precision was relatively low in their evaluations. In bridge analysis, Zhang et al. [34, 35] extracted surface primitives (cuboids, cylinders, and sheets) from point cloud data and applied a decision tree method to detect structural components like columns, caps, decks, and beams. However, this method was more suitable for bridges with straightforward, well-defined elements. Deviations between designed and built structures in practical engineering scenarios could compromise the accuracy of as-built models, conflicting with the goal of creating accurate as-is building information models (BIM).

In summary, despite the potential demonstrated by the hybrid approach in handling limited training data, there has been no previous work that successfully achieved precise point-wise classification for real-world RC bridges. This precision is crucial, especially when characterizing non-parametric elements like curved decks and I-section beams within bridges. Attaining precise point-wise classification is essential for providing a more accurate representation of the intricate geometries of these elements.

2.5. Identified Knowledge Gaps. Bottom-up geometric features-based methods for point cloud segmentation are highly dependent on the precise partitioning of surface primitives and patches. This reliance makes these methods particularly sensitive to point cloud noise, which can lead to inaccuracies in segmentation. Additionally, after the initial clustering, extra steps are necessary to integrate and analyze various bridge components, adding complexity to the process.

Machine learning and deep learning approaches offer promising alternatives for segmenting bridge components, potentially overcoming some limitations of traditional methods. However, their effectiveness is constrained by two critical factors: the size of the training dataset and the available computational power. While hybrid approaches that combine traditional methods with machine learning techniques can partially address these challenges, no existing research has yet successfully achieved a comprehensive classification of large-scale bridge structural components using these methods. This gap highlights the need for further investigation into scalable solutions that can effectively handle the complexities of large-scale bridge analysis.

3. Materials and Method

With the emergence of numerous large-scale bridges, there is a growing focus on their documentation and safety measures. The research centered on a standard long-span rail-and-road cable-stayed bridge. Given the challenge of insufficient training data and the relatively small number of large-scale bridges, the necessity and feasibility of employing deep learning methods are limited. The proposed methodology integrates a strategy of coarse to fine-grained and a top-down approach, the procedures of which contain three parts: First, 2D principal component analysis (PCA) is used to transform the coordinates and establish an independent coordinate system for the bridge; second, the entire bridge is divided into several slices along the bridge direction for further processing; and lastly, the pier, cable tower, wind fairing plate, railway surface, deck surface, main truss, and stay-cable are segmented individually.

The framework of the proposed method is illustrated in Figure 1.

3.1. Scan Data Acquisition. The point cloud was acquired using the Genius system [36], which integrates with an R-Fans-16 LiDAR and MEMS mounted on a DJI M200 UAV platform. The LiDAR operates with a wavelength of 905 nm and features 16 scanning lines. The horizontal and vertical fields are 360° and 30°, respectively. The system's laser sampling frequency is 640 kHz, with a maximum range of 250 m and a precision of 2 cm. It supports dual returns and has a total weight of 1.2 kg, making it well-suited for integration with the DJI M200 UAV. Additionally, the system includes GPS L1/L2 and GLONASS L1/L2 positioning, with a postprocessed position accuracy of 0.02 m horizontally and 0.05 m vertically.

The large-scale bridge is located in Anhui Province, P.R. China. The length of the bridge is ~ 1.6 km, and the maximum height of the bridge is ~ 140 m. To acquire the entire data of the structures below and above the bridge deck, two flights were conducted on both sides of the bridge. On each route, the flight height is set as 50 m and 100 m relative to the take-off height, and the flight route on one side is shown by the red arrow in Figure 2. The obtained original point cloud data of the bridge are shown in Figure 3(a). Since this study focuses on the main span region of the bridge, the original point cloud was manually cropped using CloudCompare software [37] to remove noise. The denoised point cloud data, which retain only the main span region, are shown in Figure 3(b).

3.2. Noise Level Estimation. In this study, point cloud noise can negatively affect subsequent processes, particularly in threshold determination. To comprehensively assess the point cloud quality, three patches are selected, each located on the deck, bridge tower, and pier, respectively. The normal vectors of these patches are parallel to the Z, X, and Y axes, respectively. Each patch has a size of approximately 2 m × 2 m, and the corresponding points can be manually obtained in CloudCompare, as illustrated in Figures 4(a), 4(b), and 4(c). The noise in the point cloud is assumed to

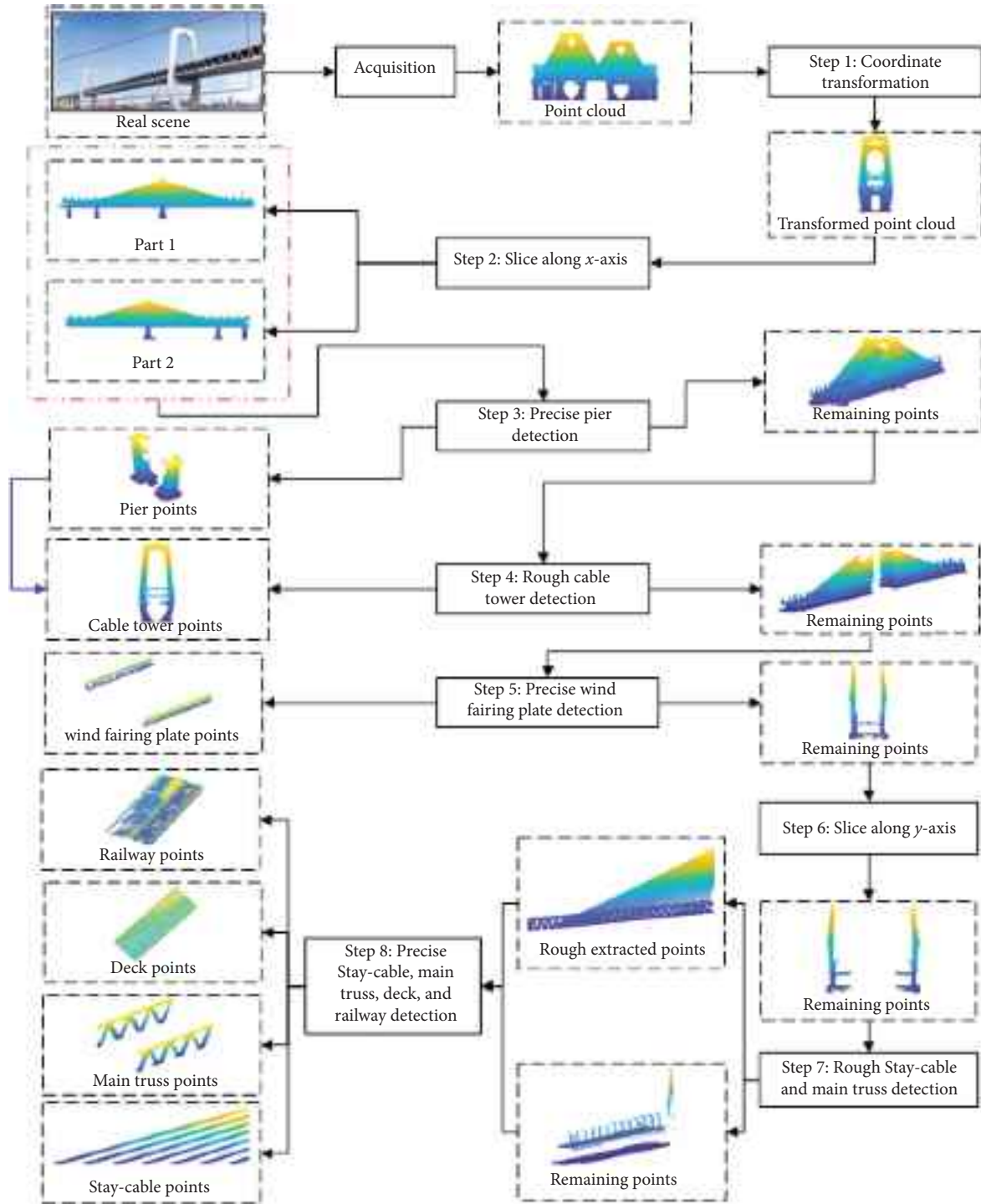


FIGURE 1: Workflow of automatic segmentation of bridge from UAV point cloud data.

follow a zero-mean Gaussian distribution [38], and its level can be evaluated by the standard deviation of the Gaussian distribution. For each patch, the standard deviation is estimated by employing PCA on the patch point cloud, as depicted in Figures 4(d), 4(e), and 4(f).

As demonstrated in Figures 4(d), 4(e), and 4(f), the standard deviations for the deck, bridge tower, and pier patches are 0.033 m, 0.082 m, and 0.038 m, respectively. It is evident that the standard deviation of the bridge tower is

significantly higher than the other two patches. This is because the side of the bridge tower is not a strictly flat surface but has a certain curvature. Therefore, the noise level of the point cloud is less than 0.04 m.

3.3. Preprocessing. The initial coordinates of the original point cloud collected by the UAV LiDAR are determined by the UAV's position, which is referenced in the WGS-84



FIGURE 2: Schematic diagram of the scanning route.



FIGURE 3: Bridge point cloud diagram. (a) Original bridge point cloud. (b) Point cloud after manual denoise.

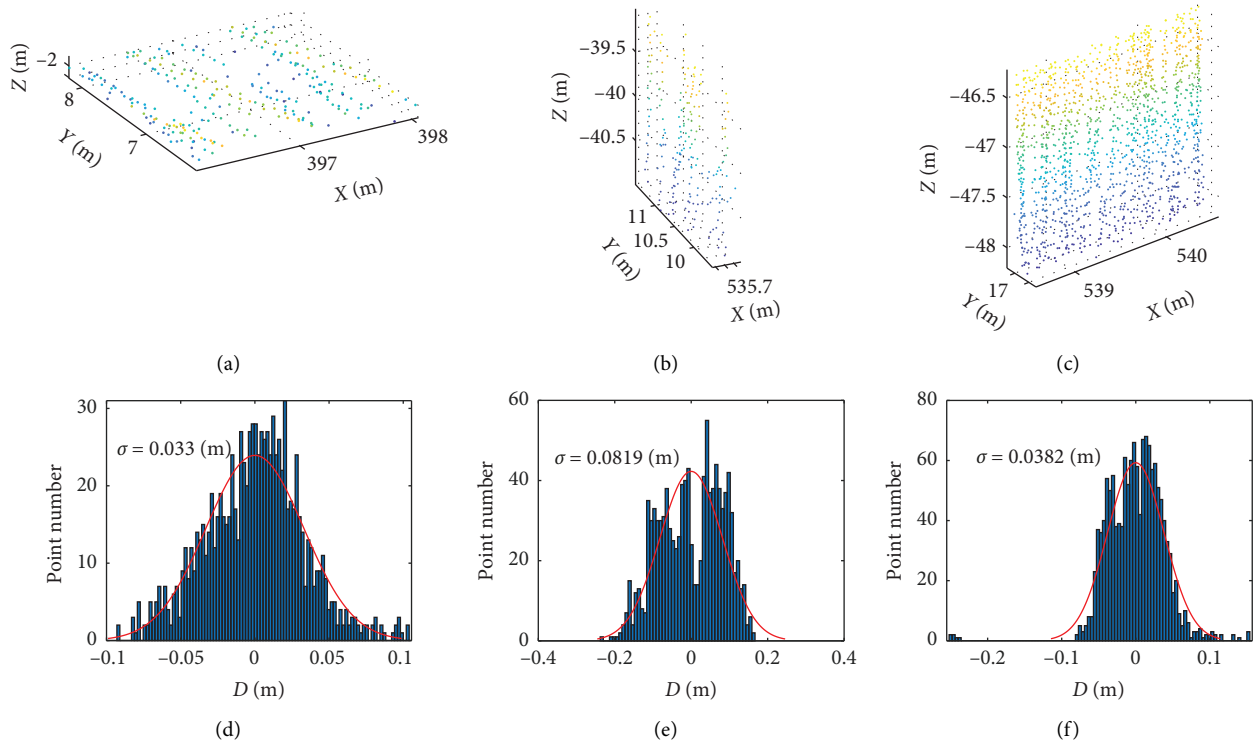


FIGURE 4: Comparison of noise level on three planar patches in three different directions. (a) Deck patch points. (b) Bridge tower patch points. (c) Pier patch points. (d) Noise level of deck patch. (e) Noise level of bridge tower patch. (f) Noise level of pier patch.

coordinate system. To enhance the comprehensibility of the structural details and facilitate subsequent processing, the original coordinates are converted into bridge coordinates. In this coordinate system, the X direction aligns with the bridge orientation, the Y direction is perpendicular to the bridge's alignment, and the Z direction points upward, perpendicular to the XOY plane.

The 2D PCA is employed to make the transformation from the UAV LiDAR coordinate system to the bridge coordinate system. Within PCA, the principal components are organized based on their variance magnitudes. In the context of 2D PCA, the objective is to identify two components: The primary component captures the direction along which the projected data exhibit the greatest variability, while the secondary component is perpendicular to the primary one. In this study, the Z -coordinates of all points are disregarded during the 2D PCA procedure, effectively aligning the first component with the bridge's orientation.

Large-scale bridge structures with two piers can typically be approximated as symmetric structures in the x -axis direction. To simplify the subsequent procedures of isolating and further segmenting the bridge piers, an initial segmentation of the bridge's point cloud is executed using a statistical method reliant on histograms. As depicted in Figure 5, this process entails the initial identification of the two most prominent peaks within the histogram, followed by pinpointing the valley position situated between these two peaks. This designated valley position serves as the separation point, effectively dividing the point cloud into two distinct segments, as visually represented in Figures 6(a) and 6(b).

After the segmentation procedure, the entire bridge's point cloud is divided into two distinct sections referred to as "Part 1" and "Part 2." These sections encompass a cable tower and all interconnected elements of the bridge associated with it. The same methodology can be applied to handle each of these parts individually. In the subsequent description of the proposed methodology, "Part 1" is utilized as a representative example for illustration purposes.

3.4. Pier Point Segmentation Precisely. Piers differ from other bridge structural components in that they are located at the foundation of the bridge. This peculiar location requires special attention when detecting the pier points. The elevation feature can be utilized to accurately extract the pier points through the following procedures:

- (1) Minimum elevation point set generation. To begin, the entire point cloud is projected onto the XOZ plane. Then, a slicing idea is introduced to divide the point cloud into multiple slices with a predefined width. Assume that the width is l_x , the entire could be divided into $n_x = \text{floor}((x_{\max} - x_{\min})/l_x)$ slices. Next, the minimum elevation for each slice, mh_i , is calculated. It is expected that the elevation at the pier location is significantly lower than in other areas. Therefore, the condition $mh_i < (\max(mh_i) + \min(mh_i))/2$ is utilized to extract the pier points in the minimum elevation set.
- (2) Pier minimum elevation point determination roughly. The k -means clustering method is introduced to identify points belonging to piers. Assume that $Ph_i = \{mh_m, mh_{m+1}, \dots, mh_n, i = 1, 2, 3\}$ is the pier point set, in which i represents the pier index and $m < n$. As illustrated in Figure 7, A represents the minimum elevation point set without the pier, while B , C , and D are different pier minimum elevation point sets, respectively.
- (3) Pier minimum elevation point determination precisely. Due to the minimum elevation point set being obtained through a given step size, a unique pier interval cannot guarantee that the starting and ending points are exactly positioned at the pier's starting point. As a result, the pier identified by the minimum elevation point set is rough which cannot meet the requirement of accurately extracting the pier's point cloud. However, the closest minimum elevation points to the corresponding pier location can be used to refine the pier points. The set of minimum elevation points within the pier location is denoted as $Ph_i = \{mh_m, mh_{m+1}, \dots, mh_n, i = 1, 2, 3\}$, as indicated by the red points in Figure 8(a). Consequently, the nearest minimum elevation points are adjacent to the pier interval mh_{m-1} and mh_{n+1} , as shown by the green points in Figure 8(a).
- (4) Boundary point extraction. The Alpha Shapes algorithm [39] is introduced to extract the boundary points from the rough set of extracted pier points. Figure 8(b) gives an example of the extracted result, in which the blue points are the boundary points, while the red points are the minimum elevation points. Here, we should demonstrate that the minimum elevation points are part of the boundary points.
- (5) Extraction of bridge pier points. The bridge piers are connected to the bridge deck through supports, and the width of the supports along the x -axis direction is smaller than the bridge pier, as illustrated in Figure 9(a). The boundary points are sorted from mh_{m-1} to mh_{n+1} according to spatial adjacency. The direction of the boundary points positioned at the supports appears to be changing. As a result, the problem becomes to find the direction change points in the boundary points. Assume that the boundary points for one pier are bp_i , in which $i = 1, 2, \dots, n$, and the vector induced by the adjacent points can be expressed by $v_j = bp_{i+1} - bp_i$, in which $j = 1, 2, \dots, n - 1$. If $v_j \cdot v_{j+1} > 0$, the boundary points bp_j, bp_{j+1} , and bp_{j+2} are in the same direction; otherwise, the point bp_{j+1} is considered a turning point. In such a way, the turning points in the boundary point set could be searched and the first and the final turning points are considered as the segmentation points, as illustrated in Figure 9(b), two purple points bp_t^1 and bp_t^2 are detected as turnback points. Finally, the pier points are extracted using the above segmentation points. The remaining point set is p_{1r} . The pseudocode is shown in Algorithm 1.

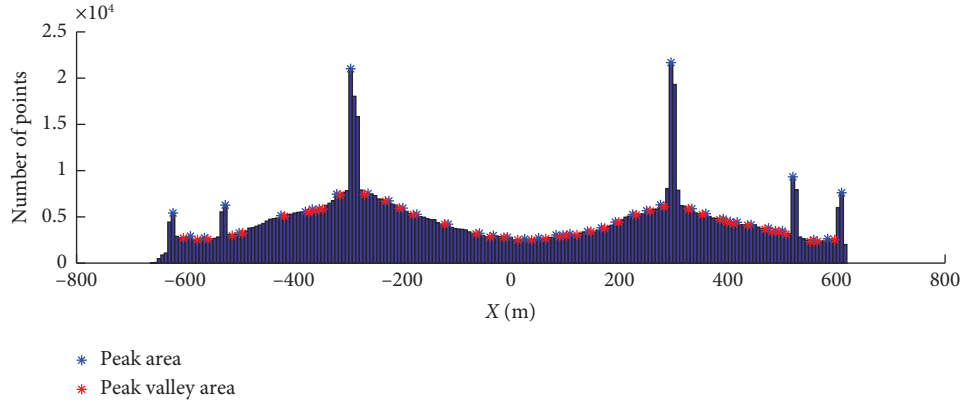
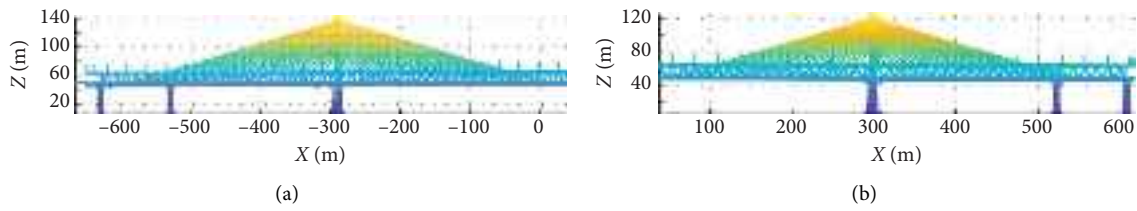
FIGURE 5: Histogram of x -coordinate.

FIGURE 6: Two segments of the entire bridge. (a) Part 1. (b) Part 2.

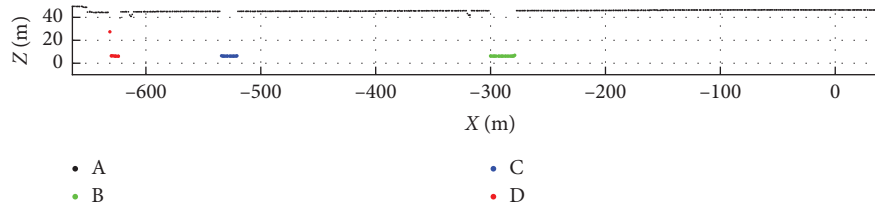


FIGURE 7: Rough pier point determination results.

3.5. Cable Tower Point Segmentation Precisely. We propose a rough extraction of cable tower points through the following procedures, with the detailed processing flow pseudocode presented in Algorithm 2.

3.5.1. Rough Extraction of Cable Tower Seed Points. As the cable tower is approximately perpendicular to the bridge direction and the side that is vertical to the bridge direction is expressed as a plane, the normal vector of the cable tower side plane is almost parallel to the bridge direction. Once the point set p_{1r} is obtained, the next step is to estimate the three normal vectors, i.e., $\{e_1, e_2, e_3\}$, using PCA algorithm. For each point in the p_{1r} , the corresponding normal vector nv_i is estimated using PCA within its k -nearest neighborhoods. The next step is to calculate the included angle between nv_i and e_1 , i.e., ne_i . A threshold value c for the included angle is set to roughly extract cable tower points, i.e., if $ne_i \leq c$, it is added into the seed points set. As shown in Figure 10, it shows the cable tower point extraction results when $c = 3^\circ$.

As one cable tower has two sides along the bridge direction, it means that there are two planes in the roughly extracted seed points. To facilitate further processing, we

propose to split the rough extracted seed points into two parts by the median segmentation along the x -axis, one slice of which is shown in Figure 11.

3.5.2. Initial Segmentation Plane Fitting Using the Random Sample Consensus (RANSAC) Algorithm. Given the possibility that the roughly extracted seed points for the cable tower contain noncable tower points, the RANSAC algorithm is introduced to fit the cable tower plane.

The cable tower is approximately perpendicular to the bridge direction, and the vertical side is represented by a plane. Considering that the points belonging to one plane have the characteristic of consistent normal vectors, we introduce the normal vector clustering method to achieve the rough extraction of cable tower seed points. Afterward, the RANSAC algorithm is utilized to fit two side planes of the cable tower, which are regarded as the initial planes for segmentation.

3.5.3. Initial Plane Translation. It is expected to extract ideal cable tower points using two planes on both sides of the cable tower. However, the cable tower points are on both

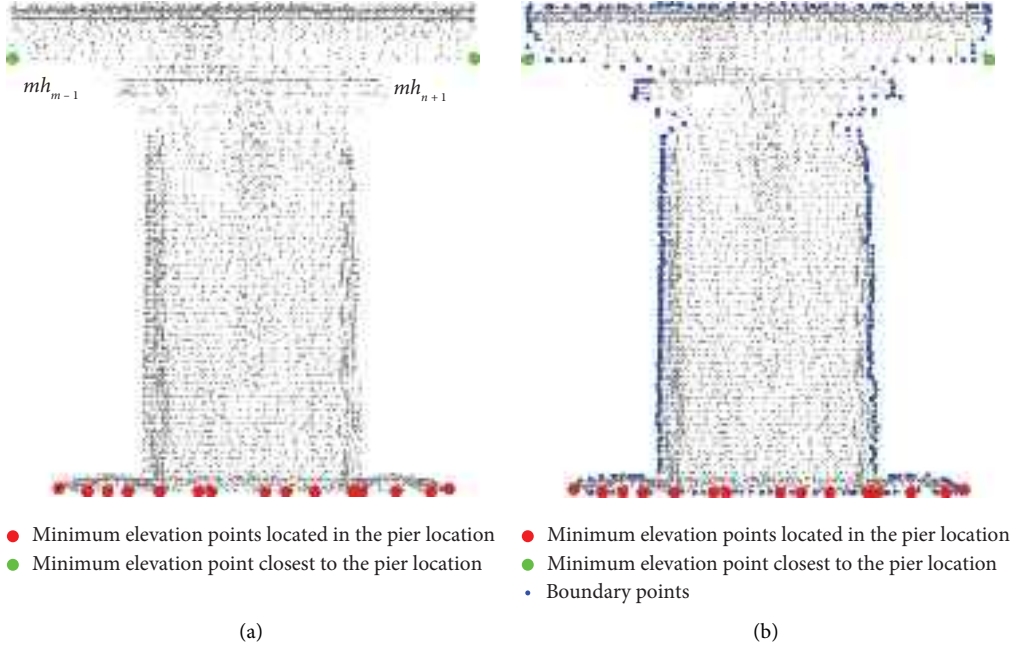


FIGURE 8: Graphical illustration for bridge pier extraction. (a) Minimum elevation point extractions. (b) Boundary point extractions.

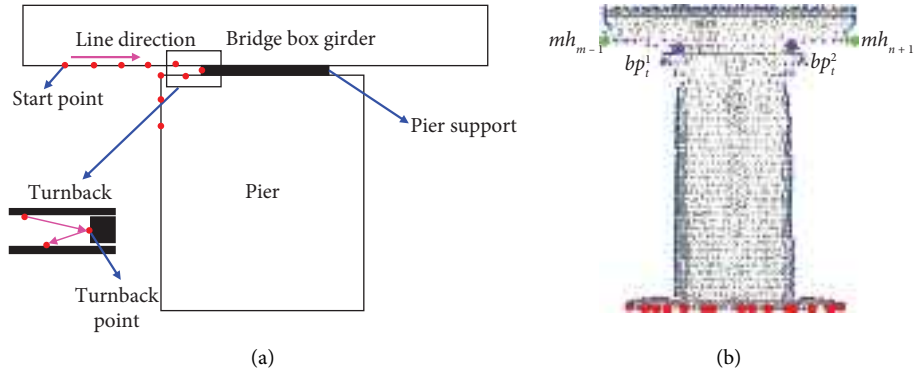


FIGURE 9: Graphical illustration of bridge pier turnback point. (a) Graphical illustration for extracting turnback point. (b) An example of turnback points extraction.

Input: Minimum elevation point set $Ph_i = \{mh_m, mh_{m+1}, \dots, mh_n\}$, boundary points bp_i with point number nbp_i , the closest minimum elevation points to the pier mh_{m-1} and mh_{n+1} .

Output: pier point set.

- 1 Sort boundary points bp_i from mh_{m-1} to mh_{n+1} ;
- 2 **for** $j = 1$ **to** $nbp_i - 1$ **do**
- 3 $v_j = bp_{j+1} - bp_j$;
- 4 $v_{j+1} = bp_{j+2} - bp_{j+1}$;
- 5 **if** $v_j \cdot v_{j+1} > 0$ **then**
- 6 Continue;
- 7 **else**
- 8 bp_{j+1} is the turning point;
- 9 **end if**
- 10 **end for**
- 11 The first and the final turning points are considered as the segmentation points bp_i^1 and bp_i^2 ;
- 12 Pier point set obtained by segmenting along the line between points bp_i^1 and bp_i^2 ;

ALGORITHM 1: Extraction of bridge pier points.

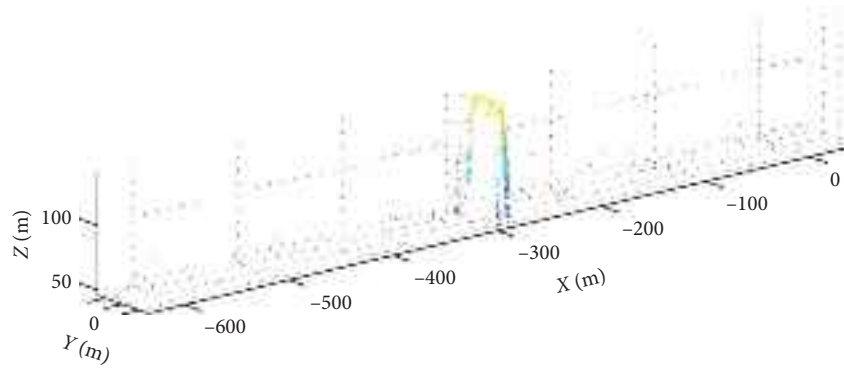


FIGURE 10: Rough extracted cable tower seed point set.

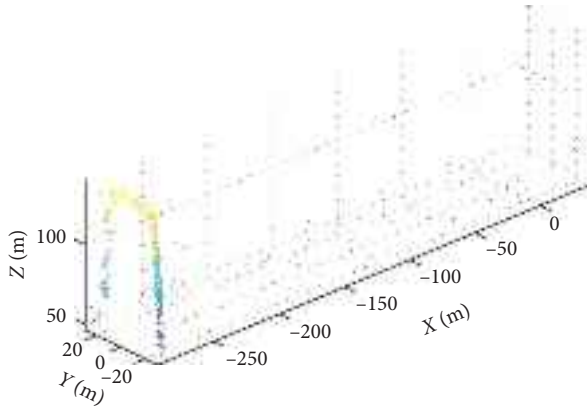


FIGURE 11: One slice of the rough extracted cable tower points.

sides of the planes caused by the fitting algorithm. This situation will result in the further extracted cable tower being incomplete. Therefore, we translate the fitted plane to ensure the inner points are on one side of the plane, as illustrated in Figure 12. The procedures are summarized as follows:

- (1) Translate distance estimation: Compared with cable tower points, the other points are relatively far away from the plane. We assume that the plane equation is expressed by $Ax + By + Cz + D = 0$, and the distance to the plane for each point is calculated. Next, the OSTU algorithm is introduced to acquire the distance threshold, representing the translate distance td .
- (2) Translate direction determination: To ensure that all cable tower points are located between the two planes, the translation directions of the two planes should be opposite. Assume point $P_A(x_{PA}, 0, 0)$ is on *Plane_A* and point $P_B(x_{PB}, 0, 0)$ is on *Plane_B*. The translation reference direction for *Plane_A* should be $\vec{N} = \overrightarrow{P_AP_B}$, and it should be $\vec{N} = \overrightarrow{P_BP_A}$ for *Plane_B*. Since the planes move along their respective normal vectors \vec{n} , to maintain consistency between the movement direction and the reference direction, if $\vec{N} \cdot \vec{n} < 0$, $\vec{n} = -\vec{n}$, otherwise, $\vec{n} = \vec{n}$.
- (3) Planar translation: We assume that the cosine of the normal vector \vec{n} with x , y , and z axes are $\cos(\alpha)$,

$\cos(\beta)$, and $\cos(\gamma)$, respectively. The equation after the translation could be expressed by the following equation:

$$A(x - td \times \cos(\alpha)) + B(y - td \times \cos(\beta)) + C(z - td \times \cos(\gamma)) + D = 0. \quad (1)$$

3.5.4. Cable Tower Point Extraction. Using the segmented two planes, the cable tower points could be determined precisely, as shown in Figure 13. The remaining point set is p_{2r} .

3.6. Wind Fairing Plate Assembly Detection Precisely. The wind fairing plate is located between the railway deck and the pier, and the pier points have been previously segmented. Therefore, the wind fairing plate points are located at the bottom of each section. They are also linked to the railway deck and distributed on both sides of the bridge direction. Given the complexity and difficulty of extracting the wind fairing plate points, we propose first to fit the railway deck plane and obtain the seed points of the railway deck, which is called a pseudo railway deck. Following that, the pseudo railway deck and the wind fairing plate are segmented. Finally, the wind fairing plate is extracted by removing the pseudo railway deck points. The specific procedures are summarized as follows, with the detailed processing flow pseudocode presented in Algorithm 3:

- (1) Acquisition of railway deck seed points. Compared with other components of the bridge deck, the bottom surface is easier to obtain as they are lower in height within local areas. Therefore, we projected the points onto XOY and raster into a series of grids, with each grid's information specified by the minimum elevation within the grid. Assume that the grid size is a , and the corresponding minimum elevation for each grid may be considered as the railway deck's seed points. Due to factors such as uneven distribution of the point cloud, it is inevitable that there exists a small number of empty grid cells. This behavior, however, has no effect on the subsequent processing findings. Figure 14 provides an example

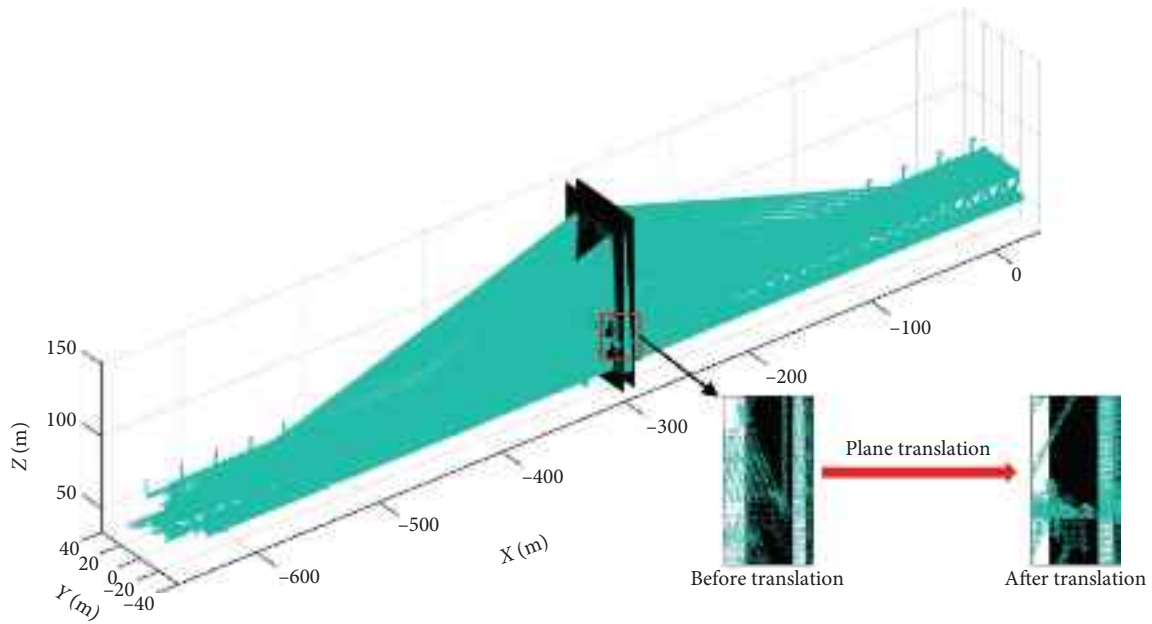


FIGURE 12: Rotated results for the segmented planes.

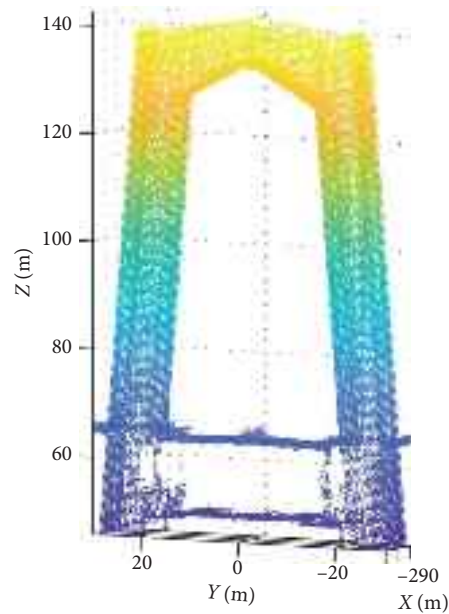


FIGURE 13: Extracted cable tower points.

of railway deck seed points extraction results when $a = 1m$.

- (2) Fitting the segmentation plane using the RANSAC algorithm. Due to the occlusion during the scanning, parts of the railway deck are not entirely recorded. Furthermore, in wind fairing plate sections, the aforementioned seed locations do not belong to the railway deck, which might be considered as noise. Given the large difference in height between the railway deck points and the noise, we proposed fitting the railway deck plane using the RANSAC algorithm, where the inlier points are the railway

deck points and the outlier points are the wind fairing plate points, as shown in Figure 15, with the red plane representing the fitted plane.

- (3) Extraction and refinement of wind fairing plate points. The left point cloud is divided into two parts along the segmentation plane, among which below the plane the point cloud contains wind fairing plate points and a small number of railway deck points, as illustrated in Figure 16(a). Next, M-estimator Least Squares with Adaptive Covariance Matrix (MLE-SAC) algorithm is introduced to segment the wind fairing plate points. In the process of this method, the

Input: The set of points remaining after removing the piers p_{1r} .

Output: Cable tower points.

```

1   $e_1 = PCA(p_{1r})$ ; //Calculate the bridge main direction  $e_1$  using PCA;
2  for each point  $t$  in  $p_{1r}$  do
3     $nv_i = PCA(t)$ ; //Calculate the normal vectors of point  $t$  using PCA
4     $ne_i = \text{AngleBetween}(nv_i, e_1)$ ; //Calculate the angle between the  $nv_i$  and  $e_1$ 
5  end for
6  Points where  $ne_i < c$  are designated as seed_points; //Extract seed points
7  seed_points are divided into two parts seed_pointsA and seed_pointsB along the X-axis based on the mean x-coordinate of the
  seed points;
8   $[Plane\_A, d\_A] = \text{RANSAC}(\text{seed\_pointsA})$ ; //Fit Plane_A using RANSAC algorithm and calculate the distance  $d\_A$  from the
  inner points to the Plane_A
9   $td\_A = \text{OSTU}(d\_A)$ ; //Obtain the plane's translation distance using the OSTU algorithm
10 Calculate Plane_A normal vector  $\vec{n}_A$  and plane point  $P_A(x_{PA}, 0, 0)$ ;
11 Perform operations as described in Steps 8, 9, and 10 on seed_pointsB to obtain  $Plane\_B$ ,  $d\_B$ ,  $td\_B$ ,  $\vec{n}_B$ , and  $P_B(x_{PB}, 0, 0)$ ;
12 Plane_A translation reference direction is  $\vec{N}_A = \vec{P}_A\vec{P}_B$  and Plane_B translation reference direction is  $\vec{N}_B = \vec{P}_B\vec{P}_A$ ;
13 if  $\vec{n}_A \cdot \vec{N}_A < 0$  then
14    $\vec{n}_A = -\vec{n}_A$ ;
15 end if
16 adjusted_planesA = AdjustPlanes(Plane_A,  $\vec{n}_A$ ,  $td\_A$ ); //Plane_A is translated in the direction of  $\vec{n}_A$  by a distance  $td\_A$ 
17 Perform operations as described in Steps 13, 14, 15, and 16 on Plane_B to obtain adjusted_planesB;
18 The set of points between adjusted_planesA and adjusted_planesB is tower points;

```

ALGORITHM 2: Precise Segmentation of Cable Tower Points.

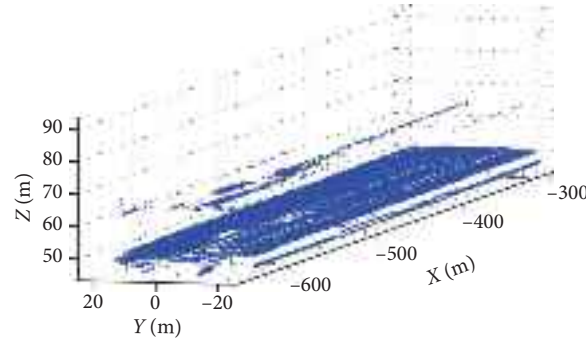


FIGURE 14: Railway deck seed point extraction results.

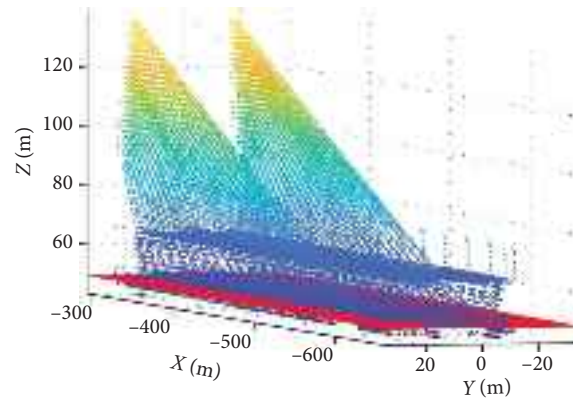


FIGURE 15: Railway surface fitting results.

referenced vector for the plane detection is $[0, 0, 1]$ which represents the normal vector of the XOY plane. A threshold of angle deviation is set to 1° and

the maximum distance from the points to the fitted plane is set to m_d . The results are shown in Figure 16(b), in which the blue points are railway

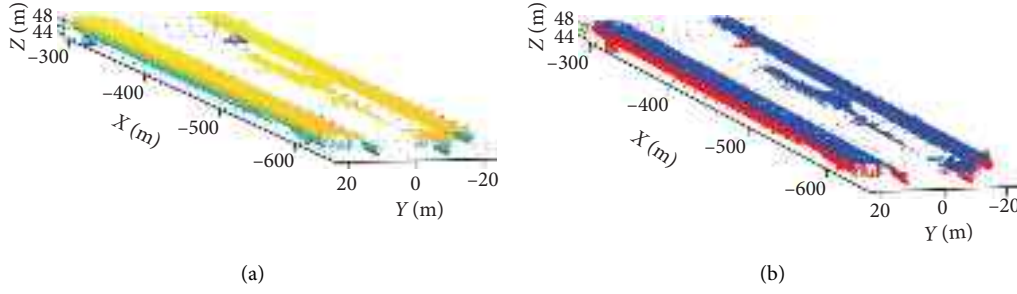


FIGURE 16: Extraction of wind fairing plate points. (a) Graphical illustration for plane segmentation. (b) Wind fairing plate points extraction results.

Input: Remaining point set p_{2r} after processing of Section 3.5, grid side length a .

Output: Wind fairing plate points.

- 1 Grid with side length a established on the XOY plane of point set p_{2r} , selecting the lowest elevation point in each grid as seed_points;
- 2 Plane = RANSAC(seed_points); // RANSAC algorithm fitting plane
- 3 p_{2r} is divided into two parts along the Plane, The part below the plane is p_{below} ;
- 4 railway_surface = MLESAC(p_{below}); // Railway surface removal using MLESAC algorithm
- 5 The remaining of p_{below} is wind fairing plate points;

ALGORITHM 3: Precise segmentation of wind fairing plate.

deck points while the red points are wind fairing plate points. The remaining point set is p_{3r} .

3.7. Stay-Cable and Main Truss Points Segmentation Roughly. The bridge components below the deck surface, such as railway surface, stay-cable, and main truss, are accessible. This property limits the accuracy and possibility of extracting the stay-cable and main truss. However, the bridge components above the deck surface are expected to be geometrically independent of each other. With these considerations, we propose a dynamic line growing method from top to bottom to realize the stay-cable detection. The specific procedures are described as follows, with the detailed processing flow pseudocode presented in Algorithm 4.

- (1) Point cloud segmenting: The point cloud is segmented with an equal interval along the z-axis, from top to bottom, into a total of t segments, in which $t = \text{floor}((z_{\text{max}} - z_{\text{min}})/\Delta z) + 1$.
- (2) Seed points acquisition:
 1. Due to the gravitational force, the stay-cable is subject to its own weight, causing it to bend or sag downward. To reduce the impact of curvature on the fitted line, points of each segment are projected onto the YOZ plane.
 2. For each layer, the Euclidean clustering algorithm is introduced to classify the points into two categories, i.e., p_i^A and p_i^B represent the inner and outer stay-cable points, respectively.
 3. The central points of each layer for two categories are estimated by calculating the mean coordinates,

and two clustering central point sets are generated, i.e., cp_i^A and cp_i^B , in which $i = 1, \dots, t$.

(3) Dynamic line growing

1. k layers' points are selected from top to down from the entire t segments.
2. RANSAC algorithm is introduced to fit the lines for two categories using the corresponding clustering central points, i.e., cp_i^A and cp_i^B , in which $i = 1, \dots, k$.
3. Line growing process. Iterate layers from $k + 1$ to t , the clustering central points from first to current layer are utilized to refit the lines. For the points in the current layer, the distance from points to the fitted lines is calculated and the points belonging to the stay-cable are determined based on a distance threshold m_d , as described by the following equation:

$$\begin{aligned} d_A \leq m_d \& d_A \leq d_B \quad p_i^A, \\ d_B \leq m_d \& d_A > d_B \quad p_i^B, \\ d_A > m_d \& d_A > m_d \quad p_{4r}, \end{aligned} \quad (2)$$

where p_i^A is for the inner stay-cable points, p_i^B is for the outer stay-cable points, and p_{4r} is for the remaining points.

The steps described are implemented and the p_i^A , p_i^B , and p_{4r} point sets are obtained, as illustrated in Figure 17(a). To facilitate subsequent processing, p_i^A and p_i^B are merged to form a new point set p_{r-sv} , as shown in Figure 17(b). The point set p_{r-sv} primarily consists of stay-cable points and

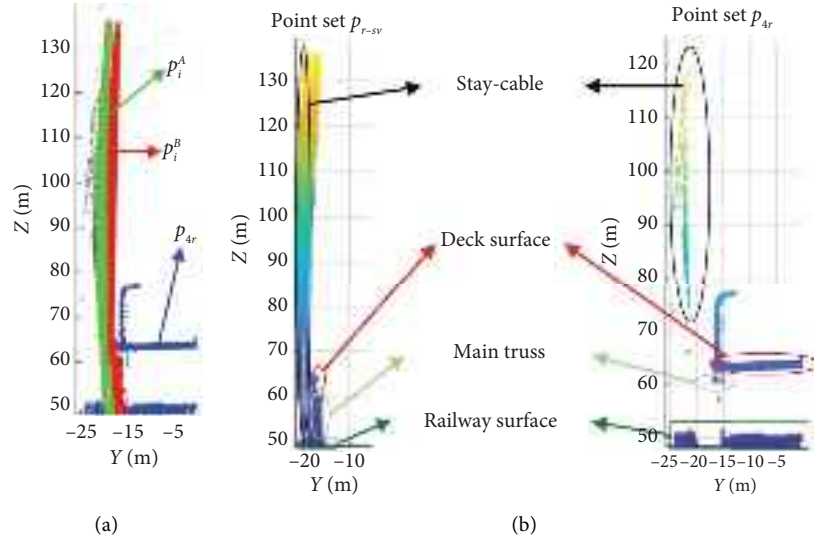


FIGURE 17: Graphical illustration for stay-cable and main truss points detection. (a) Segmentation results from the XOY plane. (b) Point sets p_{r-sv} and p_{4r} .

Input: Remaining point set p_{3r} after processing of Section 3.6, distance threshold m_d , the number of slices t , and center points k used for line fitting.

Output: Point set p_{r-sv} and p_{4r} , as shown in Figure 17(b).

```

1  Segmented into  $t$  sections along the  $z$ -axis, each slice being slice $i$ ;
2  Project all slices to the YOZ plane;
3  for  $i = 1$  to  $k$  do
4     $[p_i^A, p_i^B] = \text{EuclideanClustering}(\text{slice}_i, 2); // \text{Hierarchical clustering algorithm}$ 
5     $cp_i^A = \{cp_i^A\} \cup \{\text{mean}(p_i^A)\};$ 
6     $cp_i^B = \{cp_i^B\} \cup \{\text{mean}(p_i^B)\};$ 
7  end for
8  for  $i = k + 1$  to  $t$  do
9     $m = \{cp_i^A, cp_{i-1}^A, \dots, cp_{i-k}^A\};$ 
10    $n = \{cp_i^B, cp_{i-1}^B, \dots, cp_{i-k}^B\};$ 
11    $\text{lineA} = \text{RANSAC}(m); // \text{RANSAC algorithm fitting plane lineA}$ 
12    $\text{lineB} = \text{RANSAC}(n); // \text{RANSAC algorithm fitting plane lineB}$ 
13   for each point  $p$  in  $\text{slice}_i$  do
14     Calculate the distance from point  $P$  to the  $\text{lineA}$  and  $\text{lineB}$  as  $d_A$  and  $d_B$  respectively;
15     if  $d_A \leq m_d$  &  $d_A \leq d_B$  then
16        $p \in p_i^A$ ;
17     elseif  $d_B \leq m_d$  &  $d_A > d_B$  then
18        $p \in p_i^B$ ;
19     else
20        $p \in p_{4r}$ ;
21     end if
22      $cp_i^A = \{cp_i^A\} \cup \{\text{mean}(p_i^A)\};$ 
23      $cp_i^B = \{cp_i^B\} \cup \{\text{mean}(p_i^B)\};$ 
24   end for
25  Combine all points in the slice that are judged to be  $p_i^A$  and  $p_i^B$  into  $p_{r-sv}$ , the remaining points is  $p_{4r}$ ;

```

ALGORITHM 4: Stay-cable and main truss points segmentation roughly.

main truss points, with a minor inclusion of deck surface points and railway surface points. In contrast, point set p_{4r} mainly comprises deck surface points and railway surface points, with a small number of stay-cable points and main truss points.

3.8. Precise Separation of Rough Segmentation Results. After the preceding processing steps, we achieved precise segmentation of the bridge pier points, cable tower points, and wind fairing plate points, while the stay-cable points, main truss points, deck surface points, and railway surface

points were roughly segmented. Among them, the rough segmentation sets of stay-cable and main truss points are labeled as p_{r-sv} , the point set after the rough extraction of stay-cable and main truss is denoted as p_{4r} .

3.8.1. Segmentation of Point Set p_{4r} . As previously mentioned, the point set p_{4r} consists of deck surface points and railway surface points, with a small number of stay-cable points and main truss points. As described in Figure 17(b), the railway surface is located at the lowest part of the point set p_{4r} and is separated from other component points in the Z-axis direction. Therefore, it can be separated based on elevation values. We utilize the mean Z value to separate the railway surface points (p_{rail-1}) from the entire point set p_{4r} , and the results are shown in Figure 18(a), while the remaining points p_{5r} are shown in Figure 18(b).

As shown in Figure 18(b), stay-cable points are located at the left of the entire point set p_{5r} . It is easy to separate by combining density-based method [39] with contextual spatial relationships [40], as shown in Figure 18(b), and the red points p_{sc-1} represent the stay-cable points while the blue points p_{deck-1} are the deck points within the furniture.

3.8.2. Segmentation of Point Set p_{r-sv} . The point set p_{r-sv} still contains a small number of railway surface points. To extract these railway points, we introduce MLESAC. MLESAC estimates parameters by minimizing a cost function that incorporates the robust M-estimator approach. In comparison with RANSAC, MLESAC assigns weights to data points based on their residuals, with higher weights given to points that agree well with the model. This weight assignment helps in distinguishing inliers from outliers more effectively. The extracting results are presented in Figure 19, in which Figure 19(a) presents the extracted railway surface points (p_{rail-2}), while Figure 19(b) is the remaining point set p_{6r} .

After extracting the railway surface, the remaining points p_{6r} could be divided into two parts along the z-axis: Points with z-coordinates higher than the top of the deck surface only contain stay-cable points, while the points with z-coordinates lower than the bottom of the deck surface consist of stay-cable points, main truss points, and a small number of deck points. To achieve this segmentation, we use the point density histogram method. We generate a histogram of z-coordinates to identify the peak and peak valleys, as illustrated in Figure 20(a). In this figure, the red points and the green points represent the positions of peak and peak valleys, respectively. These positions h_c serve as the segmentation points, and Figures 20(b) and 20(c) represent the results of the point set above and below the deck surface, respectively.

Given that the point set below the deck surface p_{below} has a relatively close distance between the stay-cable, main truss, and road surface points. This increases the difficulty of extracting the stay-cable. To address this issue, we introduce the concept of stretching and stretch the point cloud along the y-axis direction, and the specific procedures are summarized as follows:

- (1) The minimum y coordinate is estimated, the value of which is utilized to determine a plane parallel to the XOZ plane.
- (2) The distances from all points to the above plane are calculated, which are then stretched by a factor of k according to the following equation:

$$H_i^y = k \times (H_i^y - \min(H_i^y)) + \min(H_i^y). \quad (3)$$

Based on this, we perform histogram statistics along the y-axis to obtain peak and valley positions, which allows us to separate the stay-cable from the main truss and the deck surface. Figure 21(a) depicts the schematic diagram of the point set p_{below} after being stretched 50 times along y-axis viewed from the XOY perspective, and Figure 21(b) is the corresponding point histogram along y-axis. Figure 21(c) is the segmented stay-cable points p_{sc-2} while Figure 21(d) is the main truss points, as viewed from the XOZ perspective.

4. Experimental Results and Discussion

This section provides the experimental results of the case study to validate the effectiveness of the proposed method.

4.1. Parameter Setting and Illustration. Before implementing the proposed methodology and evaluating its performance, we analyzed all relevant parameter settings and explained their sensitivity. Table 1 provides a comprehensive list of all the parameters used in the methodology and their recommended values. The specific setting for each parameter is summarized as follows.

During the precise segmentation of pier points, a slicing width l_x is employed to generate a minimum elevation set, the value of which is preferably smaller than the width of the pier along the x-axis. The width of the bridge piers and pier bases along the x-axis is approximately 6 m and 12 m, respectively. Therefore, we conducted experiments using various slicing widths, ranging from 0.5 m to 5 m, with a step increment of 0.5 m. The experiment results showed that the coarse results were not significantly influenced by the choice of the slicing width. This is crucial as subsequent refined steps for the extraction of bridge pier points were performed to eliminate the effect of the slicing width.

During the precise segmentation of the cable tower points, a threshold c for the included angle is utilized to coarsely extract cable tower seed points. This threshold is strongly related to the smoothness of the side surface of the cable tower. If the value is too large, there is a risk of incorrectly identifying some nontower points as cable tower points. The careful selection of this threshold is essential to ensure accurate identification of cable tower seed points and prevent misclassification of nontower points.

During the process of precise segmentation of wind fairing plate points, three parameters are employed, among which grid width a represents the resolution of raster grids, angle deviation m_{angle} represents the maximum angle difference between the scatter points with the referenced vector, and m_d is the maximum distance from the scatter

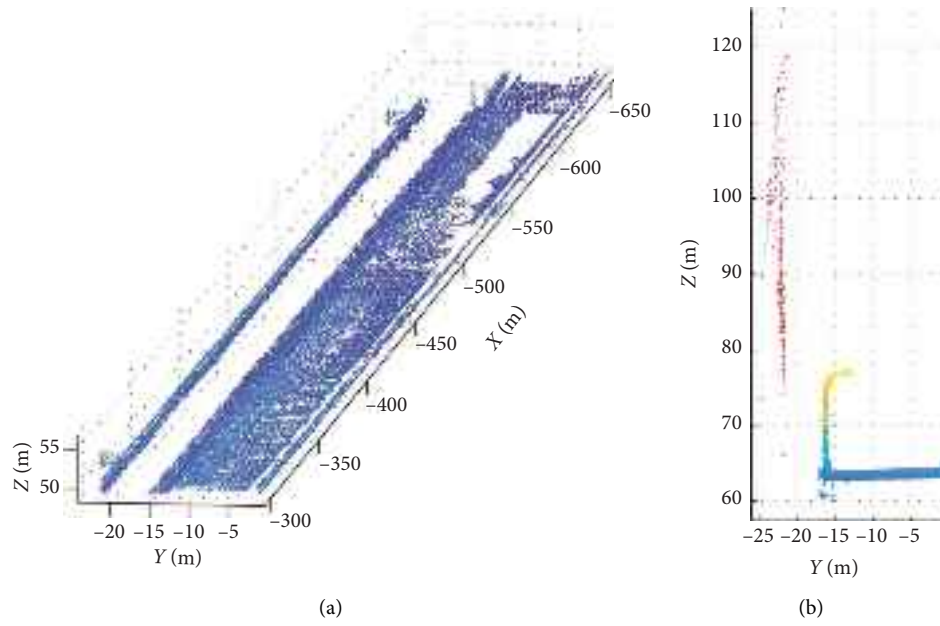


FIGURE 18: Part I of railway point segmentation diagram. (a) Railway surface points $p_{\text{rail}-1}$. (b) Remaining points p_{5r} .

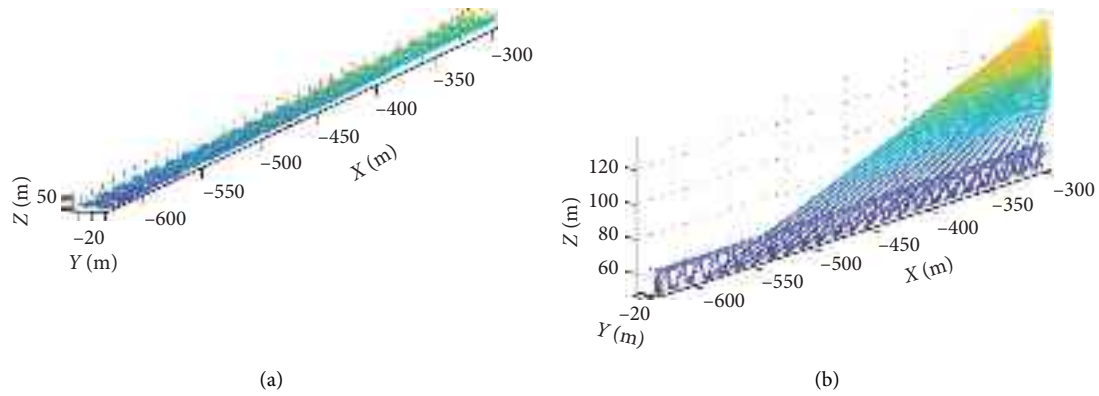
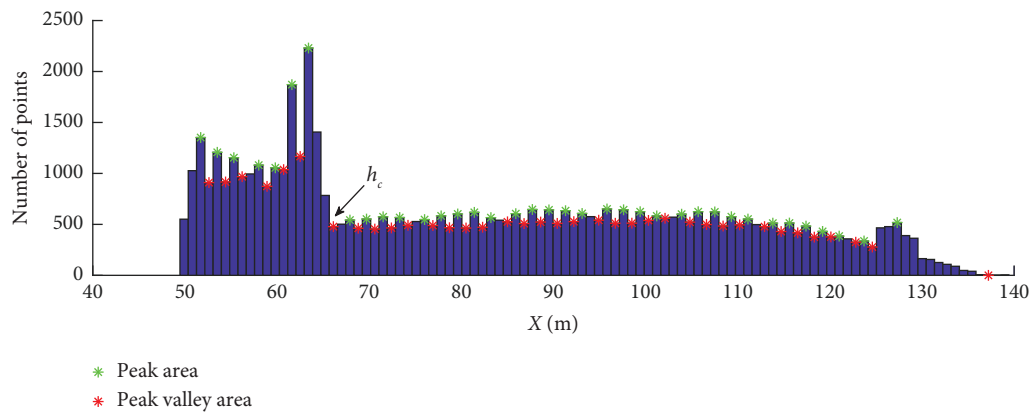


FIGURE 19: Railway surface point extraction illustration. (a) Railway surface points $p_{\text{rail}-2}$. (b) Remaining points p_{6r} .



(a)

FIGURE 20: Continued.

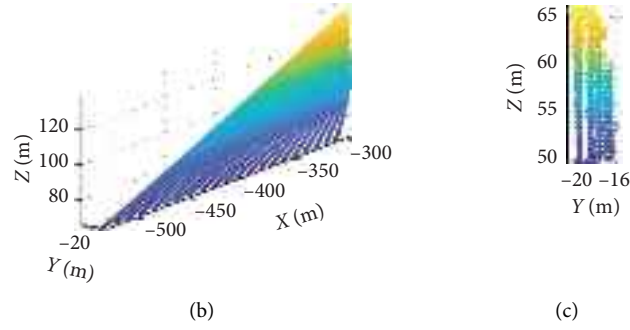


FIGURE 20: Stay-cable segmenting results. (a) Histogram of the z -axis of point set p_{6r} . (b) Stay-cable points p_{sc-2} . (c) Point set P_{below} .

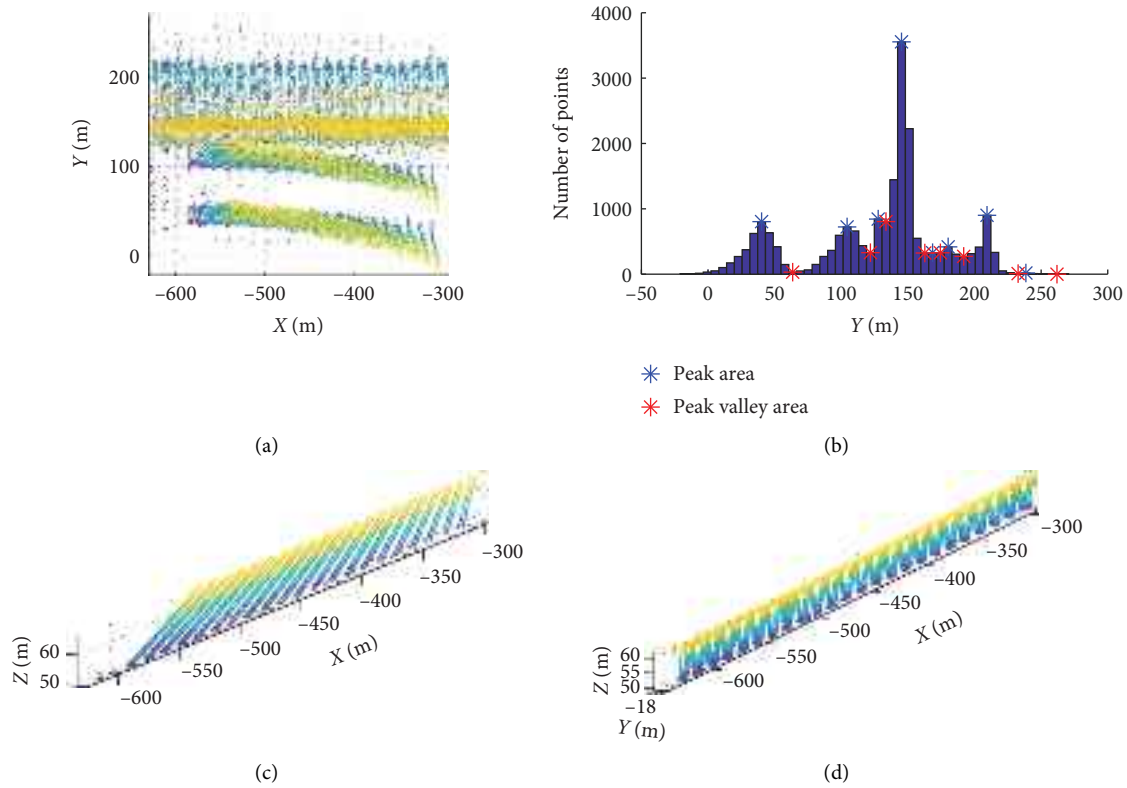


FIGURE 21: Precise separation of rough extracted results. (a) One slice in point set p_{below} . (b) Histogram along y -axis corresponds to the slice. (c) Stay-cable points p_{sc-3} . (d) Main truss points.

points to the fitted plane. Grid width represents the resolution of the raster grids. To ensure at least 2 grids in each column, it is advisable to set the grid width to be less than half of the railway width. In this case study, the railway width is approximately 40 m. Therefore, it is recommended to set the grid width within the range of 0.5 m to 20 m. m_{angle} and m_d are related to the flatness and roughness of the point cloud, respectively. It is crucial to note that these three parameters should be demonstrated to be insensitive over a wide range of values. This ensures that the segmentation method remains robust and effective regardless of variations in the point cloud.

During the rough segmentation of stay-cable and main truss points, the point cloud is segmented into t slices along the z -axis with an interval value of Δz . As mentioned above, the top k slices are utilized to generate the dynamic line

growing model. Experiments were conducted using various t , ranging from 5 to 25. The experimental results indicated that the rough extracted stay-cable points were not influenced by the different t values. This suggests robustness in the segmentation process with respect to variations in the interval value along the z -axis.

4.2. Results. For the precise segmentation of pier points, the method described in Section 3.4 is implemented. Initially, the bridge is divided into two sections, Part 1 and Part 2, along the X -axis using the approach described in Section 3.3, as shown in Figure 22. Each section undergoes the following five-step process: extraction of the minimum elevation points set, segmentation based on the mean elevation

TABLE 1: Parameter illustration for processing the point cloud.

Modules	Parameters	Descriptions and recommended values	Adopted values
Precise pier point extraction	Slice width l_x	It represents the slice width along the x-axis and its value is determined based on the pier length along the x-axis, the value of which in our case study is approximately 15 m. It is insensitive over a wide range of values and typically recommended to be set less than the pier length along the x-axis, i.e., $l_x < 15$ m	1 m
	Threshold c	The threshold c is utilized to roughly extract cable tower seed points, the value of which is related to the roughness of the point cloud itself. This parameter is recommended to be set to less than 5°	3°
Precise wind fairing plate point extraction	Grid width a	Gird width represents the resolution of raster grids. It is insensitive over a wide range of values and typically recommended to be less than half of the railway surface width, which guarantees that the number of grids along the y-axis is more than 2. In our case study, the railway surface width is ~ 40 m. Therefore, we recommended setting its value within the range of 0.5 to 20 m	1 m
	Angle deviation m_{angle}	The threshold of angle deviation is also related to the roughness of the point cloud. We recommended setting its value within the range of 0.5° to 5°	1°
	Maximum distance to the fitted plane m_d	The maximum distance to the fitted plane is also related to the roughness of the point cloud. Assume that the average value of the distances from the scatter points to the fitted plane is d_{mean} and the corresponding standard deviation is d_{std} . Considering the principle limiting error, it is recommended to set it to a value less than $d_{\text{mean}} + 3d_{\text{std}}$	0.17 ± 0.29 m
Coarse stay-cable and main truss point extraction	Slice number along z-axis t	The slice number along the z-axis is determined by Δz . Since some slices are used to generate the dynamic line growing model, at least three slices are required. It is insensitive over a wide range of values and typically recommended to be more than 3	10



FIGURE 22: Part 1 and Part 2 schematic diagrams.

threshold, k -means clustering, Alpha Shapes algorithm boundary point detection, and turning point extraction. Through this process, the segmentation lines for each pier are determined, enabling the precise extraction of individual piers. The segmentation results for the two sections are presented in Figures 23(a) and 23(b).

For the precise segmentation of cable tower points, the method described in Section 3.5 is applied. After removing the piers from Part 1 and Part 2, points whose normal vectors form an angle of less than 3° with the bridge direction are identified as rough cable tower seed points. Next, initial cable tower segmentation planes are fitted using the RANSAC algorithm. These planes are then translated along opposite directions to obtain the precise cable tower segmentation planes. The results are illustrated in Figure 24, where the red points in Figure 24(a) and the green points in Figure 24(b) represent the cable tower points in Part 1 and Part 2, respectively.

After the piers and cable towers are removed, Part 1 and Part 2 are each divided into two sections at the locations of the cable towers. Specifically, Part 1 is divided into Part 1_1 and Part 1_2, while Part 2 is divided into Part 2_1 and Part 2_2. The segmentation results are shown in Figure 25.

For the precise segmentation of wind fairing plate points, the method described in Section 3.6 is implemented. For each segment in Figure 26, a 2D grid with a 1-meter side length is constructed on the XOY plane, and the minimum elevation point in each grid cell is selected as the seed point for the railway surface. Next, the railway segmentation plane is obtained using the RANSAC algorithm, enabling the rough extraction of the fairing plate. Finally, precise extraction of the fairing plate is achieved through post-processing with the MLESAC algorithm, and the results are shown in Figures 26(a), 26(b), 26(c), 26(d).

To facilitate subsequent processing, the point clouds remaining after the removal of the wind fairing plates in Part 1_1, Part 1_2, Part 2_1, and Part 2_2 are further divided along the Y-axis based on the mean Y value. The slice results are shown in Figure 27.

For the precise segmentation of railway surface points, stay-cable points, main truss points, and deck surface points, the methods described in Sections 3.7 and 3.8 are employed. For each slice in Figure 27, a dynamic line growing method, as introduced in Section 3.7, is used to roughly segment the stay-cable and main truss points. Subsequently, the approach described in Section 3.8 is utilized to accurately separate the railway surface points, stay-cable points, main

truss points, and deck surface points. The extracted components from different slices are merged and presented in Figures 28(a), 28(b), 28(c), 28(d).

To better demonstrate the segmentation results, we have assembled all segmented components together, as shown in Figure 29, in which points of different components are colored with random colors.

4.3. Validation Results. The evaluation of segmentation accuracy typically proceeds from two perspectives, namely, point-wise and element-wise. Element-wise accuracy refers to the proportion of instances correctly retrieved by the proposed method. Considering that the method proposed in this study is a top-down approach, starting from the geometric relationships and characteristics of each component, all structural components are effectively recognized. That means that the proposed method has the high capability of identifying the presence of the structural elements in the bridge. The results described in Section 4.2 also indicate that all the components are detected successfully, which verifies the previous hypothesis. For further analysis of the accuracy, the point-wise accuracy for each structural component is then adopted to evaluate the performance of the proposed methodology.

The proposed approach for segmenting the bridge components is assessed by comparing the obtained results with the ground truth. The ground truth of each component was manually labeled using open-source CloudCompare software¹. To assess the segmentation results, the precision, recall, and F1-score can be calculated as follows:

$$\text{precision} = \frac{TP}{TP + FP}, \quad (4)$$

$$\text{recall} = \frac{TP}{TP + FN}, \quad (5)$$

$$F1 - \text{score} = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \quad (6)$$

In equations (4) and (5), TP, FP, and FN refer to the number of true positive, false positive, and false negative, respectively. TP and TN represent the number of component points that are segmented correctly into the corresponding component in the ground truth. FP represents the number of component points that are predicted as the corresponding

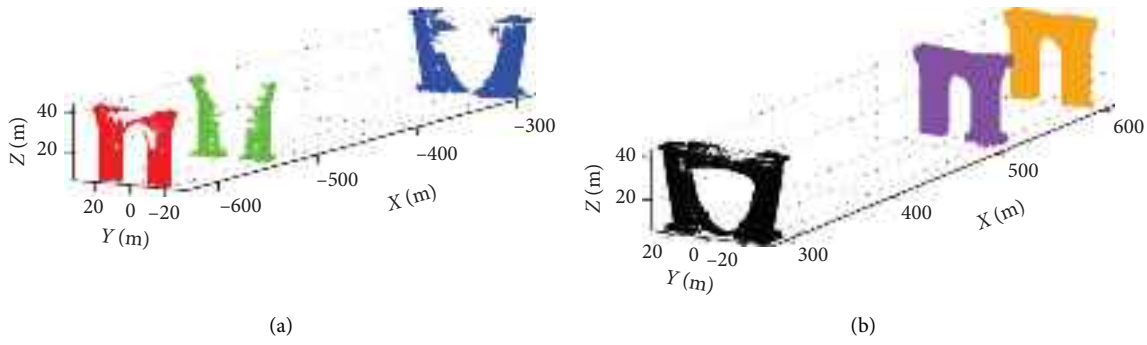


FIGURE 23: Bridge pier segmentation results. (a) Pier point segmentation results in Part 1. (b) Pier point segmentation results in Part 2.

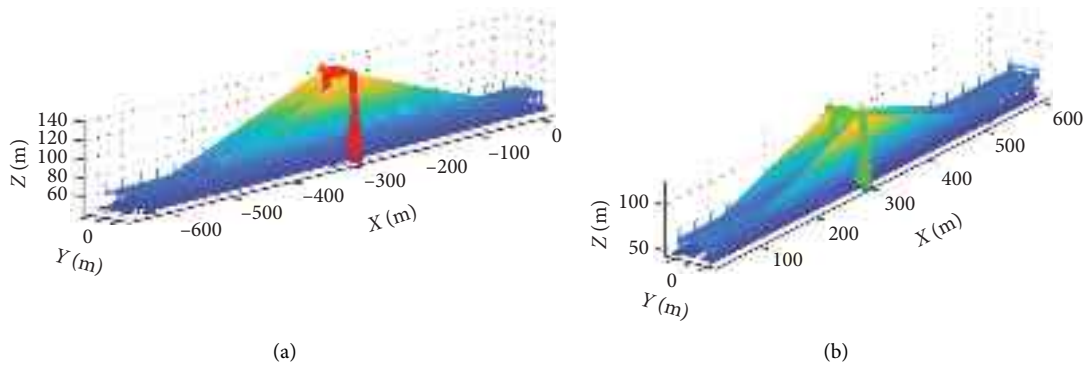


FIGURE 24: Cable tower point segmentation results after pier removal. (a) Cable tower segmentation results in Part 1. (b) Cable tower segmentation results in Part 2.

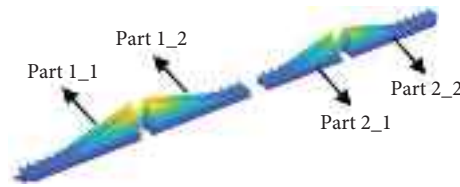


FIGURE 25: Part 1 and Part 2 segmentation results.

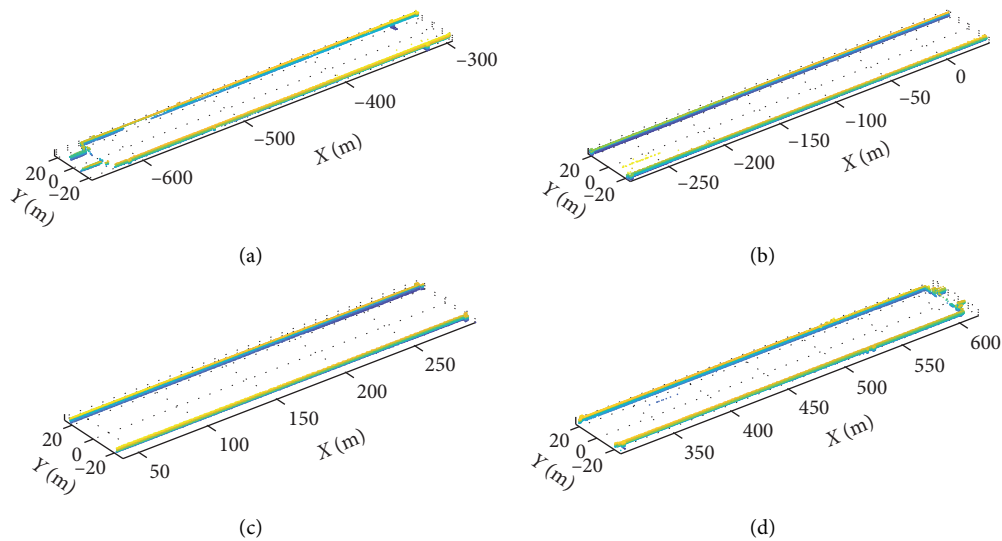


FIGURE 26: Wind fairing plate point segmentation results. (a) Part 1_1 wind fairing plate segmentation results. (b) Part 1_2 wind fairing plate segmentation results. (c) Part 2_1 wind fairing plate segmentation results. (d) Part 2_2 wind fairing plate segmentation results.

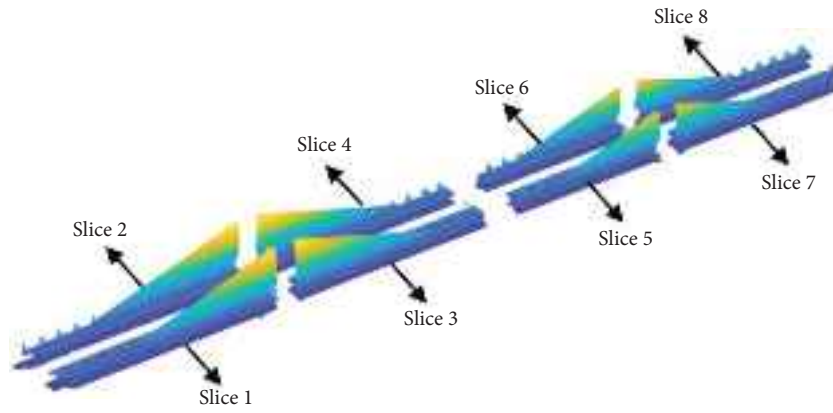


FIGURE 27: Slicing results.

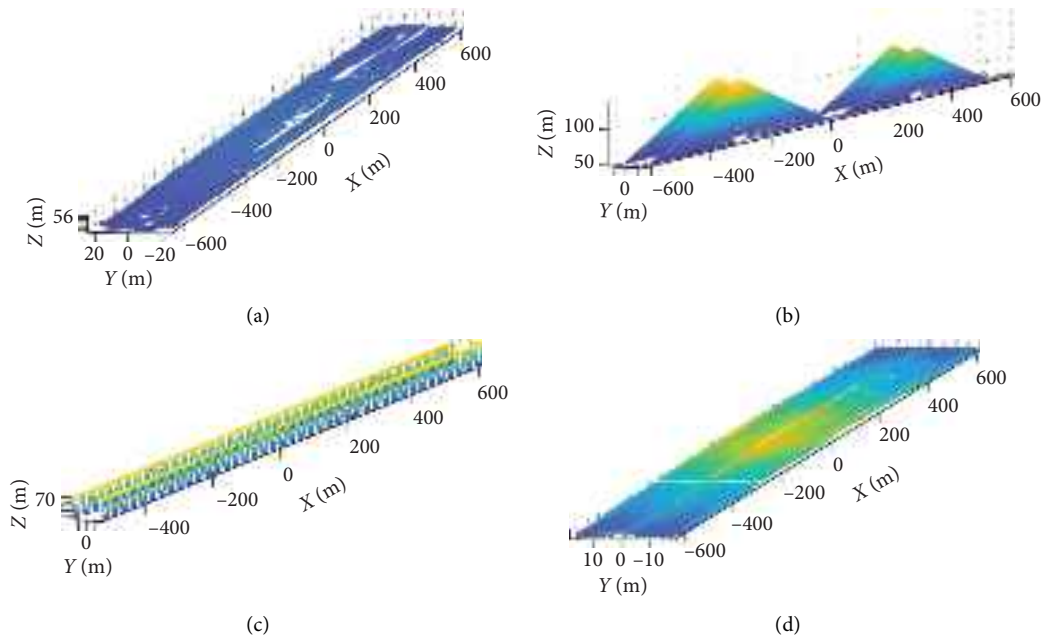


FIGURE 28: Segmentation results for railway surface, stay-cable, main truss, and deck surface. (a) Railway surface points. (b) Stay-cable points. (c) Main truss points. (d) Deck surface points.

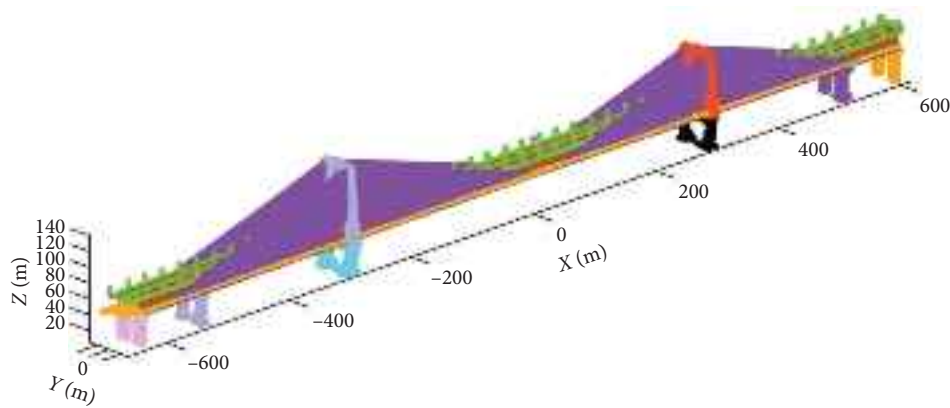


FIGURE 29: Segmentation results for the entire bridge.

component points but are labeled as other component points in the ground-truth annotation. FN represents the number of component points that are labeled as the corresponding component points in ground-truth annotations but mistakenly segmented as other component points.

The precision as given by equation (4) is the ratio of correctly extracted component points to the total predicted corresponding component points. The recall computed through equation (5) measures the ratio of correctly segmented component points to each corresponding component point in annotations. Finally, F1-score is the weighted average of the precision and recall by taking both false positives and false negatives into account. It is defined as equation (6).

The point-wise accuracy of different bridge components is shown in Tables 2, 3, 4, 5, 6, and 8.

The average precision, recall, and F1-score for extracting pier points are 99.70%, 93.20%, and 96.22%, respectively, as shown in Table 2. The recall of 4# and 6# pier is lower than 90%, which indicates that some correct points were not recognized, the number of which for 4# and 6# is 1609 and 1502, respectively. This is due to the lower support heights of 4# and 6#, which caused the edge points theoretically located on the supports not to be detected during the edge point segmentation process in step four of Section 3.4. This led to deviations in the subsequent determination of the reversal point locations. This issue can be resolved by reducing the rolling circle radius in the Alpha Shapes algorithm during the edge segmentation process.

We should demonstrate that the 3# and 4# piers are the bottom of the cable towers 1# and 2#, respectively. Thus, we merged them with the corresponding cable tower points, and the point-wise accuracy for the cable tower is calculated, as shown in Table 3.

The average precision, recall, and F1-score for extracting cable tower points are 92.26%, 99.96%, and 95.95%, respectively. It is evident that the precision for extracting cable tower points is slightly lower than the value of extracting pier points. A possible reason is that the side surface of the cable tower is not strictly flat but has a certain degree of curvature, as illustrated in Figure 13. However, in Section 3.5, we considered it as a flat surface, leading to the misidentification of several points as cable towers. Given that the number of this part of points is quite small and the absolute value of the precision is relatively high, we believe that it is acceptable.

The average precision, recall, and F1-score for extracting wind fairing plate points are 99.09%, 95.87%, and 97.44%, respectively, as shown in Table 4. From the experimental results, it is observed that the recall for each segment is lower than the precision. This discrepancy is attributed to the experimental process where the overall cutting plane of the wind fairing plate on the railway surface was initially obtained. This inclusion resulted in some railway surface points being incorporated into the top of the segmented wind fairing plate. Subsequently, the MLESAC method was employed to remove the railway surface, utilizing a fixed parameter value of 1 m and a fixed angle range of 1°. However, the connection between the railway surface and the wind fairing plate is relatively smooth, leading to the

misclassification of the wind fairing plate at the connection point as the railway surface. This misclassification contributes to the overall recall being lower than precision in the evaluation metrics.

The average precision, recall, and F1-score for extracting railway surface points are 99.22%, 97.89%, and 98.55%, respectively, as shown in Table 5. The average precision, recall, and F1-score for stay-cable points are 97.17%, 99.43%, and 98.29%, respectively, as shown in Table 6. The average precision, recall, and F1-score for main truss points are 97.82%, 98.56%, and 98.18%, respectively, as shown in Table 7.

Due to the interconnection between the bottom of the stay-cable and the main truss with the railway surface, the connected portion is classified into a point set by the classification algorithm outlined in Section 3.7, as depicted in Figure 17(b) along with the railway surface points. Subsequently, the points in the connected region are separated using the MLESAC algorithm, as described in Section 3.8.2. However, there are challenges in the extraction process. The x-distance of the cross section in this study is quite large, approximately 300 m, and the railway surface is not perfectly flat. Consequently, certain railway surface points were missed during the extraction process and were erroneously classified as stay-cable points and main truss in subsequent steps. As a result, the overall recall of the railway surface is lower than the precision, and the overall precision of the main truss and cables is lower than the recall. This highlights the complexity of accurately segmenting points in a scenario with large cross-sectional distances and non-uniform surface characteristics.

The average precision, recall, and F1-score for deck surface points are reported as 96.23%, 96.66%, and 96.40%, respectively, as shown in Table 8. It is noted that the difference in precision and recall extracted from each slice of the deck surface is not significant. However, the precision and recall of the different slices fluctuate considerably, with values above 92%. This phenomenon is attributed to the fixed parameter used in the MLESAC algorithm during the process. Additionally, the extraction results improve as each slice of the road surface becomes flatter. This highlights the sensitivity of the extraction process to the surface characteristics and the importance of considering variations in the surface profile.

The precision, recall, and F1-score for extracting bridge structure points are high with a value over 96% on the UAV LiDAR dataset.

4.4. Discussion. Previous studies, such as Lu et al. [16] and Xia et al. [18], have investigated the RC bridge segmentation from point cloud data, the former utilized the top-down idea and the latter utilized the local descriptor and machine learning. Lamas et al. [41] proposed a methodology to segment each truss element and classified them as chord, diagonal, vertical post, interior lateral brace, bottom lateral brace, or strut. Riveiro et al. [42] presented a new method for fully automated point cloud segmentation of masonry arch bridges.

TABLE 2: Point accuracy of pier point extraction.

Pier	TP	FP	FN	Precision (%)	Recall (%)	F1-score (%)
1	5216	0	540	100.00	90.62	95.08
2	5990	0	50	100.00	99.17	99.58
3 (ct)	12,068	222	0	98.19	100.00	99.09
4 (ct)	13,851	1	1609	99.99	89.59	94.51
5	10,520	0	267	100.00	97.52	98.75
6	6992	0	1502	100.00	82.32	90.30
Average				99.70	93.20	96.22

TABLE 3: Point accuracy of cable tower point extraction.

Cable tower	TP	FP	FN	Precision (%)	Recall (%)	F1-score (%)
Part 1	39,364	3647	30	91.52	99.92	95.54
Part 2	37,769	2845	4	93.00	99.99	96.37
Average				92.26	99.96	95.95

TABLE 4: Point accuracy of wind fairing plate point extraction.

Wind fairing plate	TP	FP	FN	Precision (%)	Recall (%)	F1-score (%)
Part 1_1	10,961	92	520	99.17	95.47	97.28
Part 1_2	10,870	264	183	97.63	98.34	97.99
Part 2_1	8653	3	553	99.97	93.99	96.89
Part 2_2	11,886	47	538	99.61	95.67	97.60
Average				99.09	95.87	97.44

TABLE 5: Point accuracy of railway surface point extraction.

Railway surface	TP	FP	FN	Precise (%)	Recall (%)	F1-score (%)
Slice 1	27,588	759	28	97.32	99.90	98.59
Slice 2	28,556	275	465	99.05	98.40	98.72
Slice 3	19,377	110	420	99.44	97.88	98.65
Slice 4	24,267	375	386	98.48	98.43	98.46
Slice 5	15,938	49	590	99.69	96.43	98.03
Slice 6	18,969	11	319	99.94	98.35	99.14
Slice 7	21,269	19	672	99.91	96.94	98.40
Slice 8	20,326	10	665	99.95	96.83	98.37
Average				99.22	97.89	98.55

TABLE 6: Point accuracy of stay-cable point extraction.

Stay-cable	TP	FP	FN	Precise (%)	Recall (%)	F1-score (%)
Slice 1	43,048	1116	593	97.47	98.64	98.05
Slice 2	43,787	1475	196	96.74	99.55	98.13
Slice 3	48,360	1396	236	97.19	99.51	98.34
Slice 4	50,321	1397	273	97.30	99.46	98.37
Slice 5	28,451	864	183	97.05	99.36	98.19
Slice 6	28,662	774	84	97.37	99.71	98.53
Slice 7	29,839	913	146	97.03	99.51	98.26
Slice 8	31,564	908	89	97.20	99.72	98.45
Average				97.17	99.43	98.29

In contrast, long-span rail-and-road cable-stayed bridges present greater complexity due to the numerous components they encompass, particularly those situated between the railway and the road. The intricate interconnections among these components make them challenging to

distinguish and separate, thereby increasing the difficulty of segmentation.

The primary goal of the study is to offer a case study on the segmentation and extraction of the primary components of a long-span rail-and-road cable-stayed bridge using UAV

TABLE 7: Point accuracy of main truss point extraction.

Main truss	TP	FP	FN	Precision (%)	Recall (%)	F1-score (%)
Slice 1	10,020	21	342	99.79	96.70	98.22
Slice 2	10,635	175	272	98.38	97.51	97.94
Slice 3	10,380	247	147	97.68	98.60	98.14
Slice 4	10,630	167	316	98.45	97.11	97.78
Slice 5	9352	491	20	95.01	99.79	97.34
Slice 6	10,041	149	56	98.54	99.45	98.99
Slice 7	13,101	373	6	97.23	99.95	98.57
Slice 8	13,259	345	81	97.46	99.39	98.42
Average				97.82	98.56	98.18

TABLE 8: Point accuracy of deck surface point extraction.

Deck surface	TP	FP	FN	Precision (%)	Recall (%)	F1-score (%)
Slice 1	20,764	315	1175	98.51	94.64	96.54
Slice 2	21,605	224	1728	98.97	92.59	95.68
Slice 3	20,654	291	675	98.61	96.84	97.71
Slice 4	20,800	1785	328	92.10	98.45	95.17
Slice 5	16,090	1153	430	93.31	97.40	95.31
Slice 6	15,161	958	165	94.06	98.92	96.43
Slice 7	17,494	535	424	97.03	97.63	97.33
Slice 8	18,469	518	608	97.27	96.81	97.04
Average				96.23	96.66	96.40

LiDAR point cloud data, and this type of bridge has not been previously explored. This study thoroughly examines the geometric and topological relationships between different components of large-scale bridges. We employ a combination of a coarse-to-fine approach and a top-down strategy, with tailored algorithmic workflows designed for each component to achieve high-precision extraction. Over the past decade, deep learning algorithms have gained popularity in point cloud processing due to their ability to learn complex patterns and features. However, compared to deep learning methods, the advantages of our method are as follows: (1) Interpretability: Our approach offers clear interpretability by using geometric and topological constraints, crucial for understanding decisions in engineering applications. (2) Domain-Specific Knowledge: It leverages specific knowledge of bridge structures, enhancing segmentation accuracy for complex bridges where component relationships are predictable. (3) Less Data-Intensive: The method achieves high accuracy with less data, making it advantageous when labeled point cloud data are scarce or difficult to obtain. (4) Computational Efficiency: It is more computationally efficient, especially for large datasets, compared to deep learning models that require significant resources for training and inference. One of the significant challenges encountered during the experiment was related to the operational limitations of the UAV, particularly its battery life. UAV batteries typically last between 20 and 30 min, which is considerably shorter than the total time required for comprehensive LiDAR measurements, especially when capturing large and complex structures like long-span rail-and-road cable-stayed bridges. To address this issue, we carefully planned the flight missions by dividing the bridge into multiple sections, ensuring that each section

could be scanned within the UAV's battery life. After each flight, the UAV returned to the base for a battery replacement before proceeding with the next section. This approach ensured complete coverage of the bridge without compromising data quality. Additionally, we employed multiple UAVs in a sequential manner, allowing for continuous data acquisition with minimal downtime. However, it is important to note that each extraction step in our process is dependent on the point set remaining from the previous segmentation. As a result, the accuracy of each step directly influences the subsequent steps. This dependency is an inherent characteristic of our segmentation strategy and, while unavoidable, can be mitigated by developing more targeted algorithmic combinations. The segmented bridge components obtained from point cloud data have broad practical applications in various fields, including structural health monitoring, maintenance and repair, asset management, safety and risk assessment, and historical documentation. While this study has focused on segmenting primary components, future work will address more detailed segmentation tasks, such as individual steel and stay-cable segmentation.

5. Conclusion

This study proposes a novel top-down method for the segmentation of bridge components in a long-span rail-and-road cable-stayed bridge from point clouds. Target types of structural components include bridge piers, main truss, wind fairing plate, deck surface, railway surface, stay-cable, and cable towers. The validation results prove that our method has a high degree of segmentation accuracy on large-scale bridges. Since there are fewer studies on the

extraction of LiDAR point cloud components for large-scale bridges, the automated extraction of bridge components presented in this study is of great reference significance.

The novel contributions of this research are as follows:

- (1) This study begins by analyzing the overall bridge structure and examines the geometric relationships within large-scale bridges. It provides valuable insights for other researchers working on manually constructing end-to-end point cloud features for large-scale bridges, as well as for those designing end-to-end machine learning and deep learning models.
- (2) This study employs a combined approach that integrates a coarse-to-fine concept with a top-down strategy. It designs specific algorithmic workflows for different bridge components to achieve high-precision automated extraction.
- (3) Compared to other types of bridges, long-span rail-and-road cable-stayed bridges are more complex. Therefore, the methodology presented in this work could be adapted to other large-scale structures, especially for cable-stayed bridges.

However, this method still has some significant limitations, such as the high requirements for data quality. Additionally, due to the limited dataset, we have only conducted experiments on a large-scale bridge. As more point cloud data are collected in future work, further investigation may be carried out to evaluate the performance of the proposed methodology in processing point clouds with a broader range of data completeness and quality.

Data Availability Statement

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

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

This work was supported by the National Natural Science Foundation of China under Grant No. 41801379 and Grant No. 42271450.

Endnotes

¹<https://www.danielgm.net/cc/>.

References

- [1] Mot, "Statistical Bulletin on the Development of the Transportation Industry," *Tunnel Construction* 44, no. 6 (2024): 1320.

- [2] N. Ye, L. van, and L. Panagiotis, "Analysing the Potential of UAV Point Cloud as Input in Quantitative Structure Modelling for Assessment of Woody Biomass of Single Trees - ScienceDirect," *International Journal of Applied Earth Observation and Geoinformation* 81 (2019): 47–57.
- [3] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, "Automatic Reconstruction of As-Built Building Information Models From Laser-Scanned Point Clouds: A Review of Related Techniques," *Automation in Construction* 19, no. 7 (2010): 829–843, <https://doi.org/10.1016/j.autcon.2010.06.007>.
- [4] L. Zhang and L. Zhang, "Deep Learning-Based Classification and Reconstruction of Residential Scenes From Large-Scale Point Clouds," *IEEE Transactions on Geoscience and Remote Sensing* 56, no. 4 (2018): 1887–1897, <https://doi.org/10.1109/tgrs.2017.2769120>.
- [5] B. Quintana, S. A. Prieto, A. Adán, and F. Bosché, "Door Detection in 3D Coloured Point Clouds of Indoor Environments," *Automation in Construction* 85 (2018): 146–166, <https://doi.org/10.1016/j.autcon.2017.10.016>.
- [6] Y. Wang, Y. Ma, A. X. Zhu, H. Zhao, and L. Liao, "Accurate Facade Feature Extraction Method for Buildings From Three-Dimensional Point Cloud Data Considering Structural Information," *ISPRS Journal of Photogrammetry and Remote Sensing* 139 (2018): 146–153, <https://doi.org/10.1016/j.isprsjprs.2017.11.015>.
- [7] L. Wang, Y. Zhao, and Y. Li, "A Greyscale Voxel Model for Airborne Lidar Data Applied to Building Detection," *Photogrammetric Record* 33, no. 164 (2018): 470–490, <https://doi.org/10.1111/phor.12266>.
- [8] Z. Kang, J. Yang, R. Zhong, Y. Wu, Z. Shi, and R. Lindenbergh, "Voxel-Based Extraction and Classification of 3-D Pole-Like Objects From Mobile LiDAR Point Cloud Data," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11, no. 11 (2018): 4287–4298, <https://doi.org/10.1109/jstars.2018.2869801>.
- [9] M. Soilán, A. Sánchez-Rodríguez, P. del Río-Barral, C. Perez-Collazo, P. Arias, and B. Riveiro, "Review of Laser Scanning Technologies and Their Applications for Road and Railway Infrastructure Monitoring," *Infrastructures* 4, no. 4 (2019): 58, <https://doi.org/10.3390/infrastructures4040058>.
- [10] E. Che and M. J. Olsen, "An Efficient Framework for Mobile Lidar Trajectory Reconstruction and Mo-Norvana Segmentation," *Remote Sensing* 11, no. 7 (2019): 836, <https://doi.org/10.3390/rs11070836>.
- [11] A. Dimitrov and M. Golparvar-Fard, "Segmentation of Building Point Cloud Models Including Detailed Architectural/structural Features and MEP Systems," *Automation in Construction* 51 (2015): 32–45, <https://doi.org/10.1016/j.autcon.2014.12.015>.
- [12] S. B. Walsh, D. J. Borello, B. Guldur, and J. F. Hajjar, "Data Processing of Point Clouds for Object Detection for Structural Engineering Applications," *Computer-Aided Civil and Infrastructure Engineering* 28, no. 7 (2013): 495–508, <https://doi.org/10.1111/mice.12016>.
- [13] D. F. Laefer and L. Truong-Hong, "Toward Automatic Generation of 3D Steel Structures for Building Information Modelling," *Automation in Construction* 74 (2017): 66–77, <https://doi.org/10.1016/j.autcon.2016.11.011>.
- [14] Y. Xu, S. Tuttas, L. Hoegner, and U. Stilla, "Voxel-based Segmentation of 3D Point Clouds from Construction Sites Using a Probabilistic Connectivity Model," *Pattern Recognition Letters* 102 (2018): 67–74, <https://doi.org/10.1016/j.patrec.2017.12.016>.

- [15] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, "Structural Approach for Building Reconstruction from a Single DSM," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, no. 1 (2010): 135–147, <https://doi.org/10.1109/tpami.2008.281>.
- [16] R. Lu, I. Brilakis, C. R. J. C.-A. C. Middleton, and I. Engineering, "Detection of Structural Components in Point Clouds of Existing RC Bridges," *Computer-Aided Civil and Infrastructure Engineering* 34, no. 3 (2019): 191–212, <https://doi.org/10.1111/mice.12407>.
- [17] M. Chen, A. Feng, R. McAlinden, and L. Soibelman, "Photogrammetric Point Cloud Segmentation and Object Information Extraction for Creating Virtual Environments and Simulations," *Journal of Management in Engineering* 36, no. 2 (2020): 04019046, [https://doi.org/10.1061/\(asce\)me.1943-5479.0000737](https://doi.org/10.1061/(asce)me.1943-5479.0000737).
- [18] T. Xia, J. Yang, and L. Chen, "Automated Semantic Segmentation of Bridge Point Cloud Based on Local Descriptor and Machine Learning," *Automation in Construction* 133, no. 6 (2022): 103992, <https://doi.org/10.1016/j.autcon.2021.103992>.
- [19] D. Maturana and S. Scherer, in *Voxnet: A 3d Convolutional Neural Network for Real-Time Object Recognition* (2015).
- [20] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-View Convolutional Neural Networks for 3D Shape Recognition," *2015 IEEE International Conference on Computer Vision (ICCV)* no. 1 (2015): 945–953, <https://doi.org/10.1109/iccv.2015.114>.
- [21] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Computer Vision and Pattern Recognition*, no. 1 (2017), 77–85, <https://doi.org/10.1109/cvpr.2017.16>.
- [22] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *Conference on Neural Information Processing Systems* (June 2017), 5099–5108.
- [23] H. Kim, J. Yoon, and S. H. Sim, "Automated Bridge Component Recognition from Point Clouds Using Deep Learning," *Structural Control and Health Monitoring* 27, no. 9 (2020): 1545–2255, <https://doi.org/10.1002/stc.2591>.
- [24] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN - Convolution on X-Transformed Points," in *Conference on Neural Information Processing Systems* (March 2018), 828–838.
- [25] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," *ACM Transactions on Graphics* 38, no. 5 (2019): 1–12, <https://doi.org/10.1145/3326362>.
- [26] H. Kim and C. Kim, "Deep-Learning-Based Classification of Point Clouds for Bridge Inspection," *Remote Sensing* 12, no. 22 (2020): 3757, <https://doi.org/10.3390/rs12223757>.
- [27] J. S. Lee, J. Park, and Y.-M. Ryu, "Semantic Segmentation of Bridge Components Based on Hierarchical Point Cloud Model," *Automation in Construction* 130 (2021): 103847, <https://doi.org/10.1016/j.autcon.2021.103847>.
- [28] Y. Jing, B. Sheil, and S. Acikgoz, "Segmentation of Large-Scale Masonry Arch Bridge Point Clouds with a Synthetic Simulator and the BridgeNet Neural Network," *Automation in Construction* 142 (2022): 104459, <https://doi.org/10.1016/j.autcon.2022.104459>.
- [29] Y. Jing, B. Sheil, and S. Acikgoz, "A Lightweight Transformer-based Neural Network for Large-scale Masonry Arch Bridge Point Cloud Segmentation," *Computer-Aided Civil and Infrastructure Engineering* 39, no. 16 (2024): 2427–2438, <https://doi.org/10.1111/mice.13201>.
- [30] X. Yang, E. del Rey Castillo, Y. Zou, L. Wotherspoon, and Y. Tan, "Automated Semantic Segmentation of Bridge Components from Large-Scale Point Clouds Using a Weighted Superpoint Graph," *Automation in Construction* 142 (2022): 104519, <https://doi.org/10.1016/j.autcon.2022.104519>.
- [31] X. Xiong, A. Adan, B. Akinci, and D. Huber, "Automatic Creation of Semantically Rich 3D Building Models From Laser Scanner Data," *Automation in Construction* 31 (2013): 325–337, <https://doi.org/10.1016/j.autcon.2012.10.006>.
- [32] D. H. Wolpert, "Stacked Generalization," *Neural Networks* 5, no. 2 (1992): 241–259, [https://doi.org/10.1016/s0893-6080\(05\)80023-1](https://doi.org/10.1016/s0893-6080(05)80023-1).
- [33] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic Labeling of 3D Point Clouds for Indoor Scenes," in *Conference on Neural Information Processing Systems* (April 2011), 244–252.
- [34] G. Zhang, P. A. Vela, and I. Brilakis, "Detecting, Fitting, and Classifying Surface Primitives for Infrastructure Point Cloud Data," *Computing in Civil Engineering* (2013): 589–596, <https://doi.org/10.1061/9780784413029.074>.
- [35] G. Zhang, P. A. Vela, and I. Brilakis, "Automatic Generation of As-Built Geometric Civil Infrastructure Models from Point Cloud Data," in *Proceedings of the 31st International Conference of CIB* (May 2014), 406–413, <https://doi.org/10.1061/9780784413616.051>.
- [36] W. Dai, H. Kan, R. Tan, et al., "Multisource Forest Point Cloud Registration With Semantic-Guided Keypoints and Robust RANSAC Mechanisms," *International Journal of Applied Earth Observation and Geoinformation* 115 (2022): 103105, <https://doi.org/10.1016/j.jag.2022.103105>.
- [37] D. Girardeau-Montaut, "CloudCompare," *France: EDF R&D Telecom ParisTech* 11, no. 5 (2016).
- [38] C. Gu, K. Ushiwaka, S. Tokai, and C. Zhang, "Point Cloud Filtering via Generalized Gaussian Mixture Model," in *Proceedings of the 7th IIAE International Conference on Intelligence Systems and Image Processing* (July 2019).
- [39] Y. Zhou, Z. Xiang, X. Zhang, Y. Wang, D. Han, and C. Ying, "Mechanical State Inversion Method for Structural Performance Evaluation of Existing Suspension Bridges Using 3D Laser Scanning," *Computer-Aided Civil and Infrastructure Engineering* 37, no. 5 (2022): 650–665, <https://doi.org/10.1111/mice.12765>.
- [40] B. Yang, Z. Dong, Y. Liu, F. Liang, and Y. Wang, "Computing Multiple Aggregation Levels and Contextual Features for Road Facilities Recognition Using Mobile Laser Scanning Data," *ISPRS Journal of Photogrammetry and Remote Sensing* 126 (2017): 180–194, <https://doi.org/10.1016/j.isprsjprs.2017.02.014>.
- [41] 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* 151 (2023): 104865, <https://doi.org/10.1016/j.autcon.2023.104865>.
- [42] B. Riveiro, M. DeJong, and B. Conde, "Automated Processing of Large Point Clouds for Structural Health Monitoring of Masonry Arch Bridges," *Automation in Construction* 72 (2016): 258–268, <https://doi.org/10.1016/j.autcon.2016.02.009>.