

Automated finite element modeling method for steel bridges integrating 3D point clouds and intelligent drawing recognition technology

Yixuan Chen ^{a,1}, Chenhao Gao ^{c,1}, Qijing Chen ^a, Jian Zhang ^{a,b,*}

^a School of Civil Engineering, Southeast University, 210096 Nanjing, China

^b Advanced Ocean Institute of Southeast University, Nantong 22600, China

^c Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China



ARTICLE INFO

Keywords:

Steel bridge

3D point clouds ; automated finite element modeling ; intelligent drawing recognition ; large language model

ABSTRACT

Laser scanning is widely recognized for capturing bridge geometry, yet automation of information extraction and finite element model (FEM) generation remains limited by manual intervention. Therefore, an automated FEM framework for bridges is proposed by integrating point cloud with intelligent recognition techniques. This paper presents three key contributions: (1) A high-precision external dimension extraction algorithm is developed based on bridge-specific features and secondary segmentation, combining projection density, adaptive thresholding, and region-growing RANSAC; (2) An internal drawing extraction framework is established using deep learning-based search, optical character recognition (OCR), and large language models for automated retrieval of structural information; (3) A FEM generation process is implemented by aligning internal and external data through component naming conventions, using a three-step algorithm involving segmentation, element creation, boundary and load assignment. Validations on arch bridge model and pedestrian bridge are conducted. This paper provides an initial exploration toward automated digital modeling in bridge engineering.

1. Introduction

As a vital component of transportation infrastructure, bridges are recognized for their crucial role in driving regional economic growth and facilitating social connectivity. However, with the extension of their service life, the actual stress state of bridge structures tends to degrade [1]. This degradation phenomenon not only leads to a reduction in the structural load-bearing capacity but also poses potential threats to safety. Finite element models (FEM) are employed as essential tools for assessing the service state of bridges [2]. However, limitations such as low modeling efficiency and susceptibility to human error have been encountered with existing FEM methods. In recent years, substantial application potential has been demonstrated in the field of bridge engineering by Digital Twin (DT) technology, which is capable of creating accurate virtual mappings of physical structures and enabling performance prediction and state assessment [3,4].

Point cloud technology has increasingly been recognized as a fundamental enabler of bridge digitalization, demonstrating exceptional value in automated structural modeling and intelligent structural health monitoring. Characterized by high spatial resolution, non-contact

capabilities, and automation, point cloud models enhance both the efficiency and accuracy of bridge life cycle management. High-precision three-dimensional point cloud models of bridges have been constructed using techniques such as aerial photogrammetry [5], depth camera acquisition [6], and laser scanning [7]. In structural health monitoring, significant advancements have been achieved through the integration of point cloud-based geometric information with mechanical parameter inference, enabling precise measurement of bridge alignment [8,9], accurate estimation of cable forces [10], and detailed analysis of spatiotemporal deformation characteristics [11]. These research outcomes have been applied to various bridge types, including continuous beam, cable-stayed, and suspension bridges. Moreover, novel approaches that integrate point clouds and images have been developed for the automatic identification of surface cracks and quantitative assessment of degradation levels [12]. Overall, point cloud technology constitutes a robust technical foundation for advancing the digital and intelligent transformation of bridge engineering, and exhibits considerable application prospects in the field.

The establishment of mechanical models from point cloud geometric models remains a prominent research focus in the field of structural

* Corresponding author at: School of Civil Engineering, Southeast University, 210096 Nanjing, China.

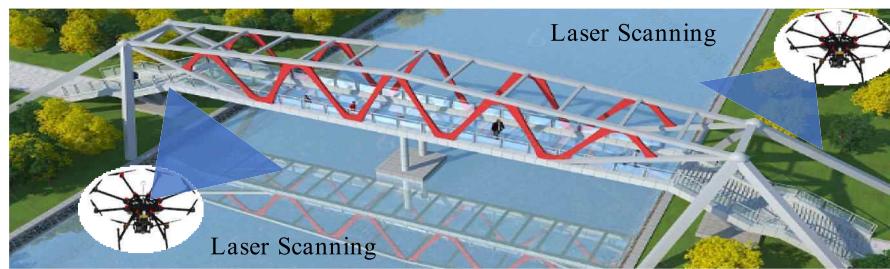
E-mail address: jian@seu.edu.cn (J. Zhang).

¹ These authors (Y. Chen and C. Gao) contributed to this work equally and should be regarded as co-first authors.

digitalization. Automatic finite element modeling often involves converting geometric models into three-dimensional solid models through surface reconstruction algorithms, such as triangulation and Poisson surface reconstruction, which address the limitations associated with using discrete point sets for structural analysis. Stull and Earls [13] proposed a technique for converting damaged bridge geometrical models into closed models, facilitating the establishment of mechanical models and estimation of residual load capacity. Duh et al. [14] developed a meshless discretization algorithm based on non-uniform rational B-spline (NURBS) models to efficiently generate quasi-uniform node sets for isogeometric analysis of CAD-defined domains. Kasotakis et al. [15] utilized Poisson reconstruction for converting point cloud geometric models into solid forms. Scholars such as Kudela et al. [16] explored direct modeling methods to convert point clouds into mesh models, validating direct finite element model establishment from point clouds. Cui et al. [17] developed a clock simulation extraction algorithm for

uniform nodes, forming the basis for automatic finite element modeling of tunnels. Despite these advances, surface reconstruction and direct meshing techniques often require high geometric regularity, which constrains their applicability to complex or irregular structural configurations.

The automatic modeling method integrating point cloud models and physical knowledge has emerged as a significant advancement in the digital representation of complex structural systems. Lu and Brilakis [18] developed a slice-based object fitting method to create high-precision digital twins of reinforced concrete bridges from marked point cloud clusters, enhancing modeling accuracy and reducing time costs. Yang et al. [19] introduced an enhanced scan-to-BIM framework that combines synthetic data augmentation with parametric modeling. Shu et al. [20] developed an automatic geometric digital twin method for box girder bridges using laser scanning point clouds. Talebi et al. [21] enabled the direct transformation of BIM into FEM from point cloud



Challenge : The deep mining of complex bridge point cloud data is challenging and lacks internal information, making automatic finite element modeling difficult.

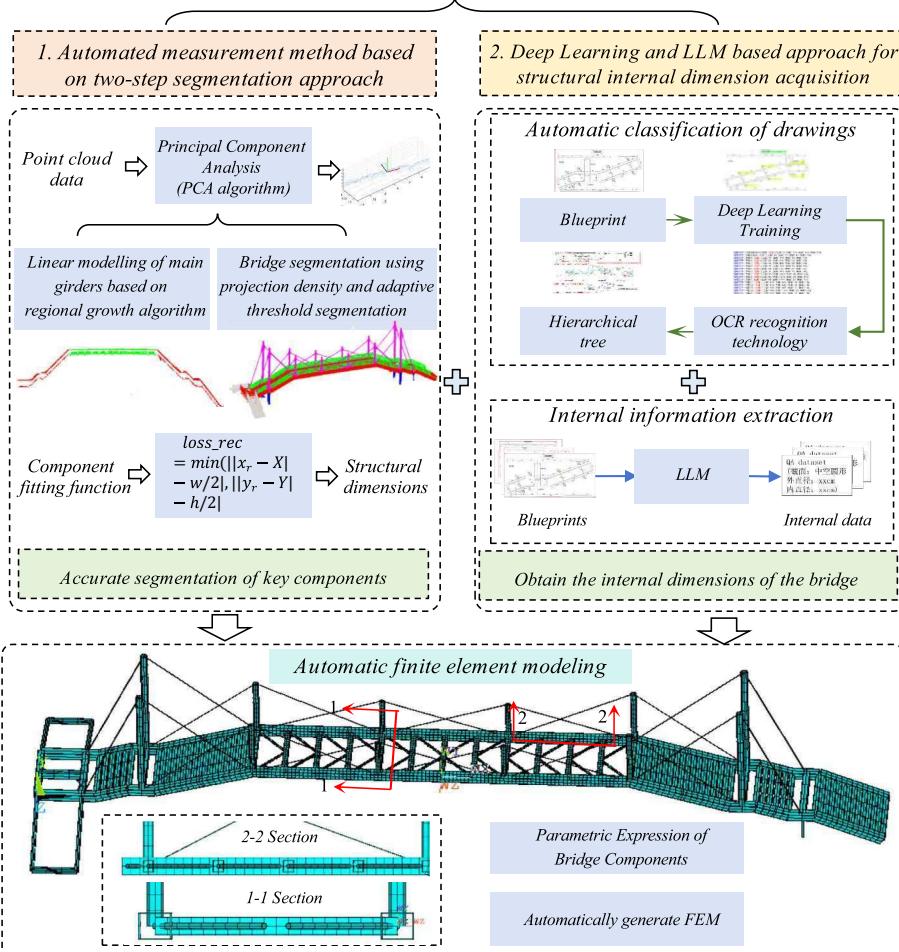


Fig. 1. Proposed methodological framework.

models. Huang et al. [22] proposed a parametric modeling and safety assessment method for shield tunnels within an automated BIM-FEM framework. Gao et al. [23] proposed a method for detecting concrete spalling damage and assessing seismic performance in shear walls, utilizing three-dimensional reconstruction and numerical model updating. Conde-Camero et al. [24] achieved direct conversion of point clouds into FEMs for structural condition evaluation. Yan and Hajjar. [25,26] introduced automatic geometric modeling for superstructure components of steel beam bridges and presented a scan-to-FEM strategy for creating high-fidelity finite element models. Despite theoretical advances, critical challenges remain, including the difficulty in extracting dimensions from complex bridge structures and the inability of point cloud models to reflect internal structural details accurately.

To advance the automation of bridge information extraction and finite element modeling, a method for steel bridges is proposed, in which three-dimensional point cloud data are integrated with intelligent drawing recognition techniques. As illustrated in Fig. 1, Three key innovations are introduced in the proposed methodology: (1) A two-step segmentation algorithm is developed to address the challenges associated with automated extraction of external dimensions. This approach integrates projection density analysis, adaptive thresholding, and region-growing based on RANSAC to accurately identify geometric boundaries while accounting for the physical characteristics of the bridge. (2) To overcome the limitations of point clouds in capturing internal structural details, an automated technical drawing interpretation framework is established, incorporating deep learning-based search, optical character recognition (OCR), a hierarchical structure tree, and a large language model. Through this framework, internal parameters are automatically extracted, enabling parametric modeling

in a fully automated manner. (3) A finite element model (FEM) generation process is established through the alignment of internal and external data via component naming conventions. A structured three-step algorithm comprising component segmentation, element generation, and boundary and load application is adopted to enhance the efficiency and consistency of the modeling workflow. The effectiveness of the proposed method is demonstrated through steel arch bridge model and pedestrian bridge. Compared with conventional approaches, automated geometric extraction and efficient FEM construction are achieved, thereby providing a reliable foundation for the development of bridge digital twins. The remainder of this paper is organized as follows: Section 1 introduces the research background; Section 2 presents the innovative concepts of the proposed method; Section 3 discusses the automatic extraction method for key external information of bridge components based on point cloud models; Section 4 describes a two-step approach for quickly obtaining internal dimensions of structures based on deep learning and large language models; Section 5 provides the method of automatic finite element modeling; Section 6 provides an explanation of the proposed method based on Laboratory steel arch bridge model. Section 7 present a detailed explanation of the proposed method in conjunction with a practical example of a steel pedestrian bridge; and Section 8 summarizes the findings of this paper.

2. Proposed framework

As illustrated in Fig. 2, an automated finite element modeling method for steel bridges, integrating three-dimensional point cloud segmentation and intelligent drawing recognition technology is proposed in this study. The objective is to seamlessly automate the

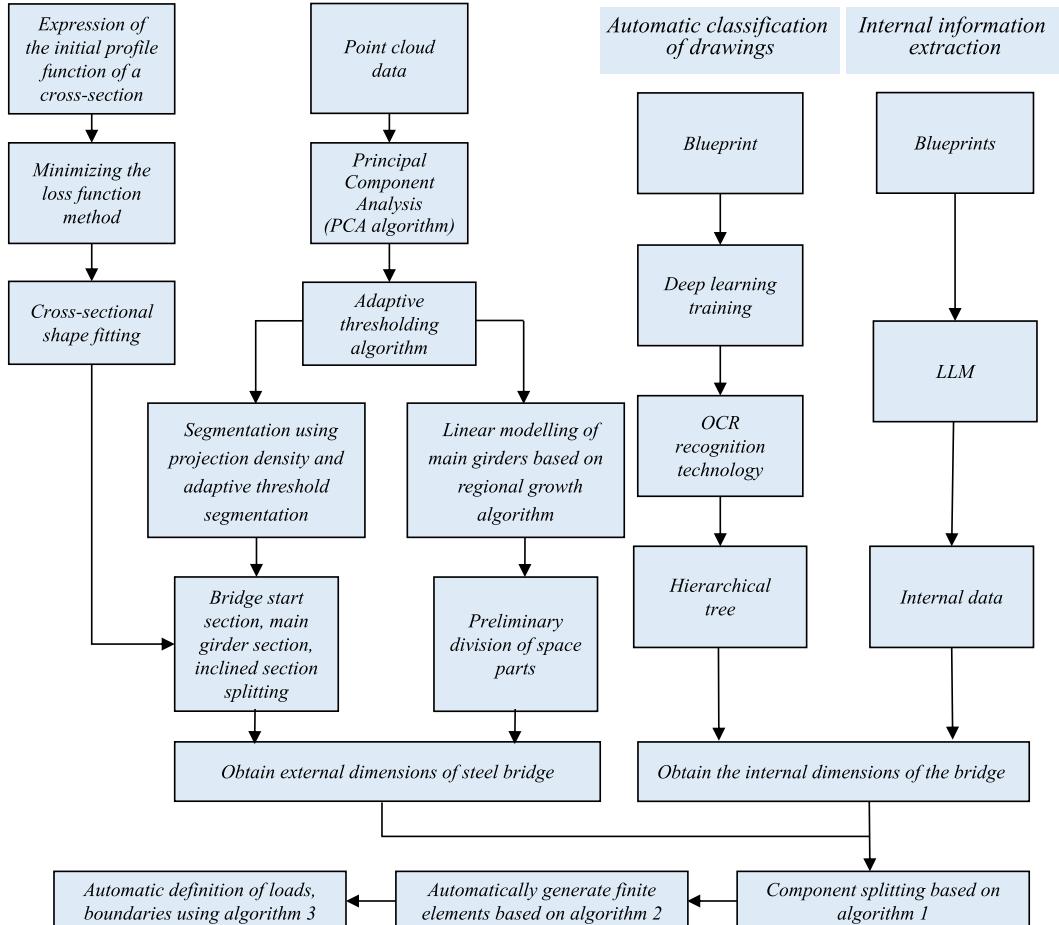


Fig. 2. Process of the proposed method.

extraction of key bridge dimensions to enhance the efficiency of bridge automated modeling, thereby promoting the application of digital twin technologies in bridge engineering. For external information, point cloud data undergo Principal Component Analysis (PCA) and adaptive thresholding to enable segmentation and linear modeling of main girders. Projection-based density analysis and region-growing algorithms are then used to split bridge sections and define spatial partitions, ultimately yielding the external geometric dimensions. Simultaneously, blueprint classification is performed using a deep learning-based OCR recognition pipeline, followed by a hierarchical parsing structure. Internal information such as structural dimensions is extracted using large language models (LLMs) and blueprint interpretation. Cross-sectional profiles are reconstructed by minimizing loss functions and performing shape fitting. The fused external and internal data are then used to generate finite element components, with boundary conditions, loads, and mesh elements defined through three dedicated algorithms. This method not only emphasizes the critical role of automated information extraction but also advances automated finite element modeling, forming a synergistic framework that significantly improves both the accuracy and efficiency of digital modeling in bridge engineering.

3. Automatic extraction of external information for key bridge constructions based on point cloud modeling

As shown in Fig. 3, an automatic segmentation method for critical bridge components based on point cloud models is proposed, which is guided by physical knowledge to achieve precise recognition and division of bridge structures. The method begins with the preprocessing of multi-site point cloud data, where principal component analysis (PCA) algorithms are applied to extract the main girder point cloud and perform coordinate transformation. Subsequently, adaptive threshold algorithms are utilized to segment the main girder, allowing for the rapid division of the bridge's superstructure, deck system, and substructure. On this basis, a secondary segmentation is conducted by incorporating projection density information and structural characteristics, further subdividing the superstructure into bridge towers and stay cables, and the deck system into main girders, cross beams, and braces. Ultimately, the automatic extraction and information retrieval of critical components are achieved. This method substantially improves the efficiency and accuracy of segmenting key bridge components, thereby providing effective support for engineering applications.

3.1. Point cloud data acquisition and pre-processing

A terrestrial laser scanning system is employed to capture the geometry of the structure from multiple viewpoints. After the point cloud model of the bridge is obtained, a comprehensive preprocessing procedure is carried out. The preprocessing steps include the registration of multi-station point cloud data, the manual removal of various

environmental interferences (such as trees, pedestrians, vehicles, etc.), point cloud downsampling, point cloud denoising, and coordinate transformation. During the multi-site stitching process, GPS data is initially used for coarse alignment, followed by refined alignment using the iterative closest point (ICP) algorithm. Regarding the removal of environmental interferences, several automated processing approaches have been proposed. Yan and Hajjar. [27] applied a morphology-based ground filtering algorithm to identify ground-related points. Xu and Zhang. [28] and Zeng et al. [29] segmented point clouds from background regions using two-dimensional masks. When the point cloud model is acquired in conjunction with camera images, these methods can be used to automatically suppress background interference. Xia et al. [30] developed a method combining local descriptors with machine learning techniques to detect bridge structural components directly from point clouds, thereby eliminating the need for manual background filtering and denoising. By scanning multiple bridges or generating virtual datasets, deep learning-based methods can be employed to minimize manual preprocessing efforts.

Point cloud downsampling is performed using a voxel-based method, which is designed to reduce the number of points and decrease computational complexity. Point cloud denoising is conducted using a statistical filtering algorithm that calculates the distances from each point to its neighboring points within the point cloud model. Points with distances that exceed a specified threshold (typically defined as the mean distance plus three times the standard deviation) are identified as outliers.

The objective of the point cloud coordinate transformation is to align the longitudinal axis of the point cloud with the coordinate axes, thereby facilitating the extraction of projection density in subsequent procedures. The transformation process is carried out using the Principal Component Analysis (PCA) method, which is based on the calculation of point dispersion (variance), and the direction with the highest variance is designated as the principal axis. During implementation, the covariance matrix of the point cloud is computed and analyzed. For the point cloud $P = \{p_i = (x_i, y_i, z_i) \in R^3, i \in [1, N]\}$, its covariance matrix can be represented as follows:

$$\sum = \begin{pmatrix} \text{cov}(X, X) & \text{cov}(X, Y) & \text{cov}(X, Z) \\ \text{cov}(Y, X) & \text{cov}(Y, Y) & \text{cov}(Y, Z) \\ \text{cov}(Z, X) & \text{cov}(Z, Y) & \text{cov}(Z, Z) \end{pmatrix} \quad (1)$$

where $X = \{x_1, x_2, \dots, x_N\}$, $Y = \{y_1, y_2, \dots, y_N\}$, $Z = \{z_1, z_2, \dots, z_N\}$.

This method is susceptible to the influence of point asymmetry and noise. Although downsampling and denoising are applied during preprocessing, the point cloud may still exhibit non-uniform density, which can result in deviations in the principal axis direction obtained through Principal Component Analysis (PCA) during global structural analysis. Therefore, in practical implementation, structural characteristics are considered, and the point cloud corresponding to the edge beams located at the mid-span section of the structure is selected for principal

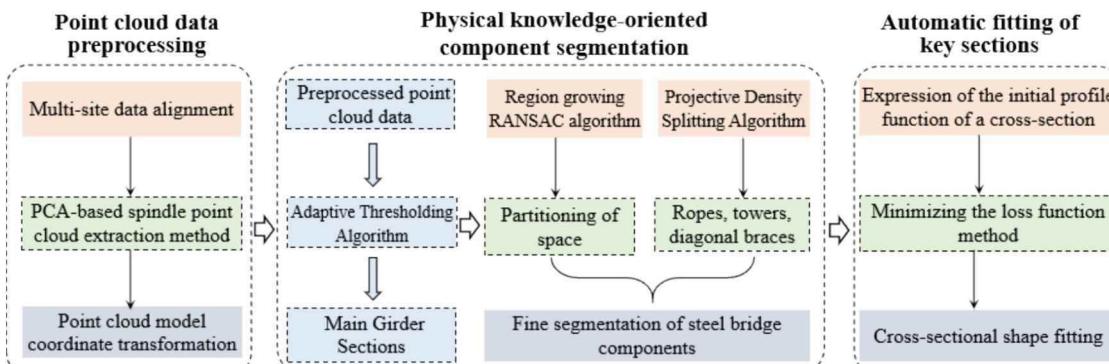


Fig. 3. Automatic extraction of external geometric information for key bridge components based on point cloud modeling.

axis computation, as illustrated in Fig. 4.

First, the centroid coordinate X_c of the point cloud along the X-axis is calculated. Subsequently, an effective interval $[X_c - \Delta, X_c + \Delta]$ is determined by extending forward and backward, and the valid point cloud data within this interval is extracted:

$$P_s = \{p_{si} = (x_i, y_i, z_i) \in R^3, i \in [1, K], x_i \in [x_c - \Delta, x_c + \Delta]\} \quad (2)$$

In practical calculations, Δ is set to 2.5 m. Next, the point cloud is sliced along the X-axis at intervals of Δx (set to 0.01 m during the calculation), and points with the maximum absolute y-coordinate values are extracted:

$$\begin{aligned} Q = \left\{ q = (x_q, y_q, z_q) \in R^3, \forall x_q \in [x_{min} + n \times \Delta t, x_{max} + n \times \Delta t], \left| y_q^* \right| \geq \left| y_{others} \right| \right\} \end{aligned} \quad (3)$$

Finally, the extracted points Q is filtered, and the three principal axes are computed using the Principal Component Analysis (PCA) algorithm. The point with the minimum X-coordinate in the original point cloud is designated as the origin, a coordinate system is established, and the denoised point cloud is transformed into the new coordinate system to complete the coordinate transformation.

3.2. Automatic segmentation algorithm for critical bridge components based on physical knowledge orientation

As shown in Fig. 5, this method is designed to incorporate the structural characteristics of the bridge. First, vertical structural filtering is achieved using projection density analysis and an adaptive threshold algorithm. Then, the primary beam profile of the bridge is extracted based on the region-growing RANSAC algorithm. Subsequently, the superstructure, deck system, and substructure are delineated in conjunction with the extracted beam profile and the region-growing algorithm. Building on this, secondary segmentation is conducted, further integrating the structural features of the bridge and projection density information. In this step, the superstructure is divided into towers and stay cables, while the deck system is categorized into main girders, crossbeams, and bracing components. Finally, the critical components of the entire bridge are accurately segmented.

The proposed segmentation algorithm, which is based on projection density and an adaptive thresholding strategy, is illustrated in Fig. 6, with the segmentation of a bridge tower provided as an illustrative example. The point cloud, pre-aligned within the defined coordinate system, is used as input, and slicing is conducted along the X-axis with a

predefined slice thickness. For each slice, the number of points it contains is calculated, and this value is interpreted as the projection density. A smoothing operation is subsequently applied to generate a one-dimensional continuous distribution, referred to as the smoothed density profile, which is clustered using the DBSCAN algorithm. Among the local maxima identified across all clusters, the minimum value is selected and multiplied by a factor of 1.2 to define the adaptive threshold for segmentation.

$$thre = 1.2 \times \min[\max(class_1), \max(class_2)] \quad (4)$$

After the threshold is determined, a region aggregation selection algorithm is employed to identify the structures to be selected: (1) all local maxima in the smoothed projection density array are extracted, and those greater than the threshold are filtered out; (2) local minima on both sides of the filtered maxima are calculated, and the regions defined by the x-coordinates of these two minima is designated as the initial regions; (3) adjacent initial regions are merged to form effective areas, resulting in the extraction outcome.

The objective of step 2 is to enable coarse extraction of the main girder. The specific implementation is described as follows: along the X-axis direction, the projection density along the Z-axis is extracted within each slice interval (which is the same as the interval of the projected slice). Two local regions corresponding to the peaks of the Z-axis projection density are identified, and clustering is performed based on the Z-coordinates of the detected regions.

Step 3 involves the automated computation of the main girder profile parameters based on the regional growth RANSAC algorithm proposed in this study. The detailed execution logic of the algorithm is described as follows:

- (1) The upper (or lower) plane of the main girder is projected onto the XOZ plane, resulting in the point cloud model $S = \{s_i = (x_i, z_i) \in R^3, i \in [1, K]\}$.
- (2) Points with smaller X values are selected to form the initial point set $S_{ini} = \{s_j = (x_j, z_j), 0 < x_j < ini, s_j \in S\}$ (in the actual computation, ini is set to 1.5). The RANSAC algorithm is subsequently applied to the initial points to perform line fitting, resulting in the initial line L . The principle of the RANSAC algorithm is summarized as follows: a subset of points is randomly sampled from the point set S_{ini} for line fitting, and the distances from the remaining points in the set to the line are calculated. If the distance is less than the threshold $RANSAC_thre$ (set to 0.005 m in the actual computation), it is classified as an inlier; otherwise, it is considered an outlier. This process is iterated multiple times, and the

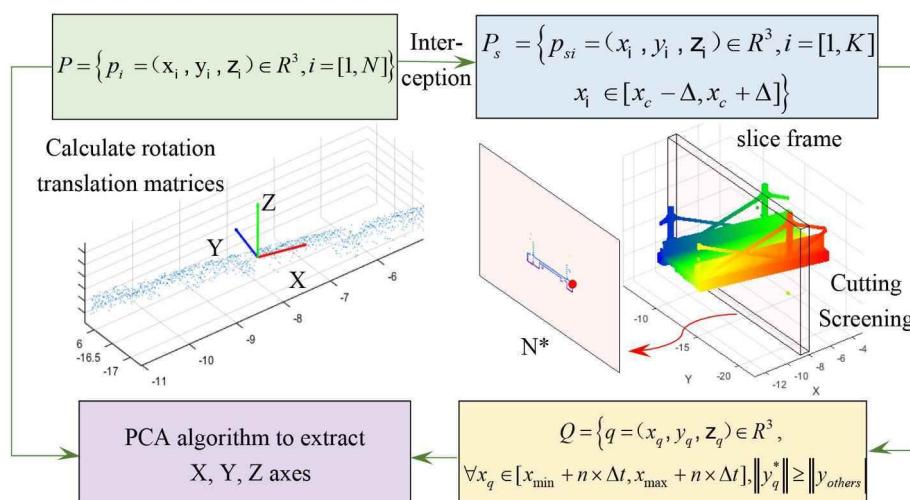


Fig. 4. Point cloud spindle extraction.

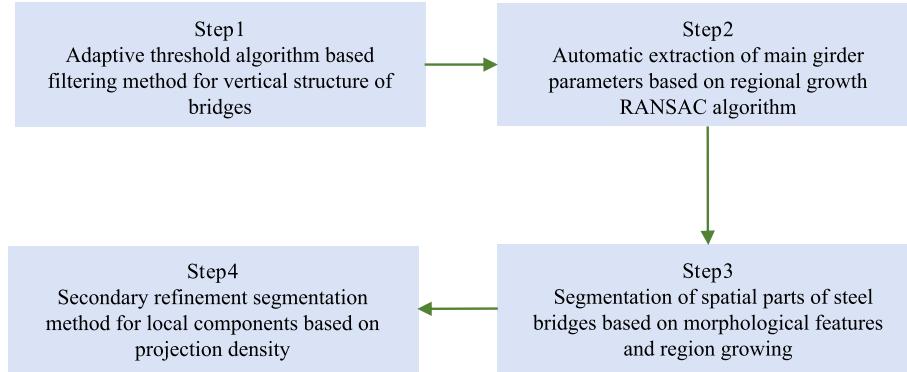


Fig. 5. Physical knowledge-oriented algorithm for fast segmentation of critical bridge components.

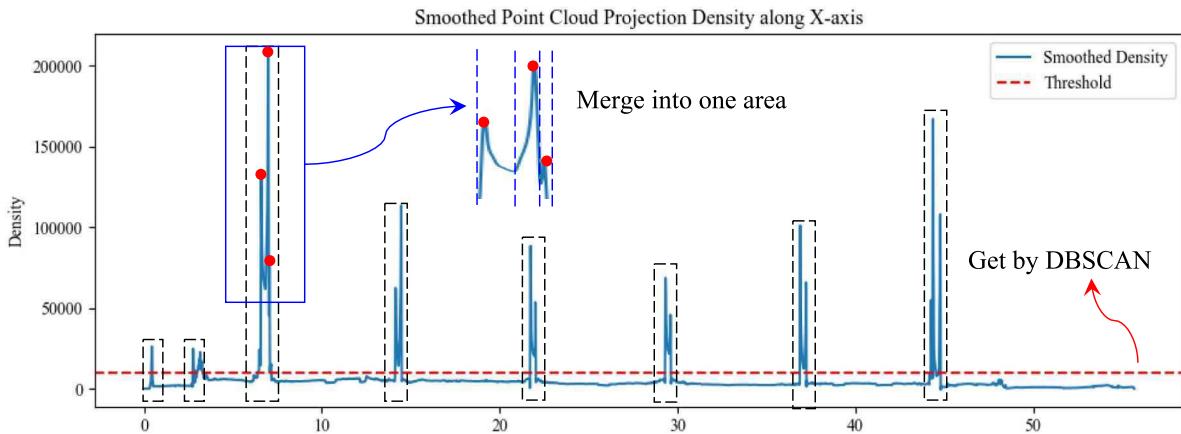


Fig. 6. X-axis projection density map.

model with the maximum number of inliers is selected as the final fitting result, with the corresponding inlier set denoted as I at that time.

- (3) Along the X-axis, new points S_{new} are continuously added, and the distance d between point S_{new} and the initial line L is calculated. If $d < thre$, S_{new} is classified as an inlier and included in the inlier set I . If $d > thre$, the point S_{new} is temporarily marked as an outlier. If, among the subsequent n added points (with n set to 30 in the actual computation), $2/3n$ points show a continuous slight increase in distance (indicating the presence of a slope) or all lie within the range of $[d - thre, d + thre]$ (indicating the formation of a new platform), then S_{new} is confirmed as a turner; otherwise, it is regarded as a noise point. Which means:

$$S_{new,1} \in \text{turner} \text{ if } \text{distance}(S_{new,1}, S_{new,2}, \dots, S_{new,n}) > thre \quad (5)$$

- (4) Once the outlier point is identified, the advancement along the X-axis is terminated, and the inlier set I is utilized to re-fit the line L' , resulting in the representation of the main girder prior to the outlier point.
(5) Let the coordinates of the outlier point be (X_c, Z_c) . All points in the point cloud model S with X coordinates less than X_c are removed, and the X coordinate of the outlier point is set to 0, resulting in a new point cloud model $S = \{s_i = (x_i - X_c, z_i) \in R^3, i = [1, K]\}$.

Steps (2) to (5) are repeated until the parametric representation of the main girder is calculated.

The proposed algorithm is designed to detect the distance to the

outlier point and to fit the upper and lower surfaces of the main girder as multiple linear segments. The underlying computational principle is illustrated in Fig. 7(a), and the corresponding results are presented in Fig. 7(b). The parametric representation of the main girder is ultimately expressed as follows:

$$Z = \begin{cases} K_{un}X_n + b_{un} & \text{upper planes} \\ K_{ln}X_n + b_{ln} & \text{lower planes} \end{cases} \quad (6)$$

Through the above preprocessing, further segmentation of the bridge components can be conducted, dividing the structure into the superstructure, substructure, and deck system. Components with Z coordinates located above the upper plane of the main girder are classified as the superstructure, while components located below the lower plane are categorized as the substructure. In order to further refine the segmentation, key components need to be identified by considering the characteristics of the bridge structure. For the superstructure, the bridge towers and cables must be distinguished. The projection density is calculated along the X-axis to determine the position of the bridge towers, allowing for the segmentation of the bridge tower area and cable area from the superstructure. As for the deck system, the beam positions are first identified using the projection density method, and the positions of the deck boards are extracted through clustering based on Z coordinates, leaving the remaining area as the brace area, which can be segmented using local PCA clustering.

3.3. Automatic fitting the external parameter of bridge components

Two different shapes of cross-sectional profiles, rectangular and circular (which are the only two shapes present in the pedestrian bridge

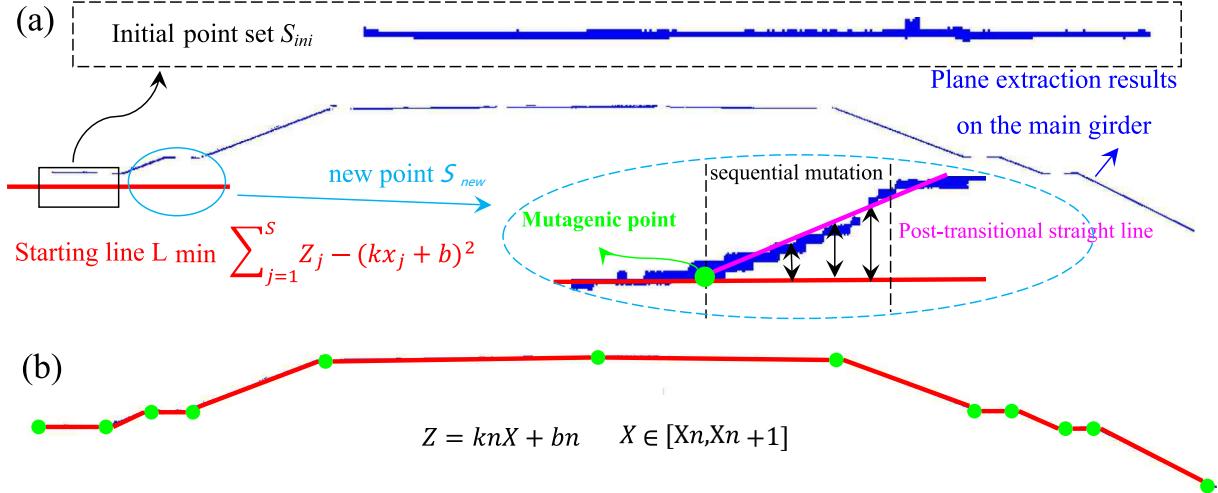


Fig. 7. Main girder parameterization results: (a) calculation principle; (b) main girder parameterization results.

under testing), are considered and parameterized accordingly. The parameterized model for the rectangular cross-section includes positional parameters (X, Y, θ) and shape parameters (W, H). The circular cross-section includes positional parameters (X, Y) and shape parameters (R). For the rectangular fitting, the fitting is performed by minimizing the loss function $loss_rec$, which is defined as the sum of the squares of the distances from each point to its nearest rectangular boundary. To handle the rotation of the rectangle, each point is initially transformed with respect to the center of the rectangle, and then the shortest distance to the rotated rectangular boundary is calculated. For any point (x_i, y_i) , its coordinates after being rotated by an angle θ around the rectangle's center (X, Y) are calculated as (x_r, y_r) using the formula:

$$x_r = (x_i - X) \times \cos(\theta) - (y_i - Y) \times \sin(\theta) + X \quad (7)$$

$$y_r = (x_i - X) \times \sin(\theta) + (y_i - Y) \times \cos(\theta) + Y \quad (8)$$

The loss function $loss_rec$ is given by:

$$loss_rec = \min(|x_r - X| - w/2|, |y_r - Y| - h/2|) \quad (9)$$

The parameterization algorithm for the circular cross-section is similarly operated to that of the rectangular cross-section, with the difference being that the sample points are fitted to a circle using the least squares method, where the objective of the least squares optimization is to minimize the sum of the squares of the distances from all points to the center of the circle.

For different components, segmentation is conducted by advancing along the principal axis direction with a specified slice step length, as indicated in Table 3. The orientation of the bridge tower is aligned with the Z-axis, while the orientation of the cables is determined based on the locally derived principal direction from Principal Component Analysis (PCA) clustering. When analyzing the parametric representation of the tower column, the point cloud of the tower column is defined as follows:

$$P_{tower} = \{p_i = (x_i, y_i, z_i) \in R^3, i \in [1, K_{tower}]\} \quad (10)$$

The slicing procedure is performed with Δ_{tower} as the step length and Δc_1 as the thickness for slicing in the direction of advancement (where Δ_{tower} is set to 1/30 of the tower height, and Δc_1 is set to 0.02 m). The subset of the point cloud is subsequently defined as:

$$\begin{aligned} P_{subtower} = & \{p_i = (x_{si}, y_{si}, z_{si}) \in R^3, Z_{min} + n\Delta_{tower} - \Delta c/2 \leq z_{si} \\ & \leq Z_{min} + n\Delta_{tower} + \\ & \Delta c/2, i \in [1, K_{tower}]\} \end{aligned} \quad (11)$$

An example of parameter fitting results is illustrated using the cable.

As shown in Fig. 8, the parameterization of the cable is performed in a manner similar to that of the bridge tower. However, the direction of advancement is no longer aligned with the Z-axis but follows the local principal direction derived from PCA-based clustering. A coordinate transformation is conducted accordingly, aligning the data with this local principal direction prior to the calculation.

4. A two-step method for automated internal dimension acquisition of structures using deep learning and large language modeling

The measurement method based on the point cloud model is capable of providing key geometric information of the structure; however, it fails to capture the internal cross-sectional dimensions. This limitation is particularly pronounced in bridge structures, which frequently comprise numerous hollow cross-section components (such as box girders), where reliance solely on point cloud data precludes access to internal details. Therefore, further investigation has been undertaken to employ deep learning networks to assist in the automatic retrieval of relevant drawings, when available, thereby identifying the structural components requiring measurement. This approach is intended to reduce manual screening efforts and facilitate the acquisition of internal dimensional information.

As illustrated in Fig. 9, an automated method is proposed for extracting internal dimensional information of bridge structures, aiming to address the limitations of point cloud models in capturing internal geometric features. The method is divided into two principal stages. In the first stage, bridge drawings are processed using deep learning-based image recognition and optical character recognition (OCR) techniques to enable the automatic identification and classification of key structural elements. Advanced image preprocessing algorithms are applied, followed by the deployment of deep neural networks to detect components and dimensional annotations. Simultaneously, OCR is utilized to extract textual information, which is then organized into a hierarchical structure tree, forming a structured representation of the drawing content. In the second stage, internal dimensional data are extracted from key sectional drawings using a fine-tuned open-source large language model (LLM) through prompt-based inference. The obtained information is structured into a standardized digital format, providing a reliable basis for subsequent finite element modeling and structural analysis. This method significantly enhances automation and accuracy in acquiring internal dimensions, offering technical support for the intelligent modeling of complex bridge systems.

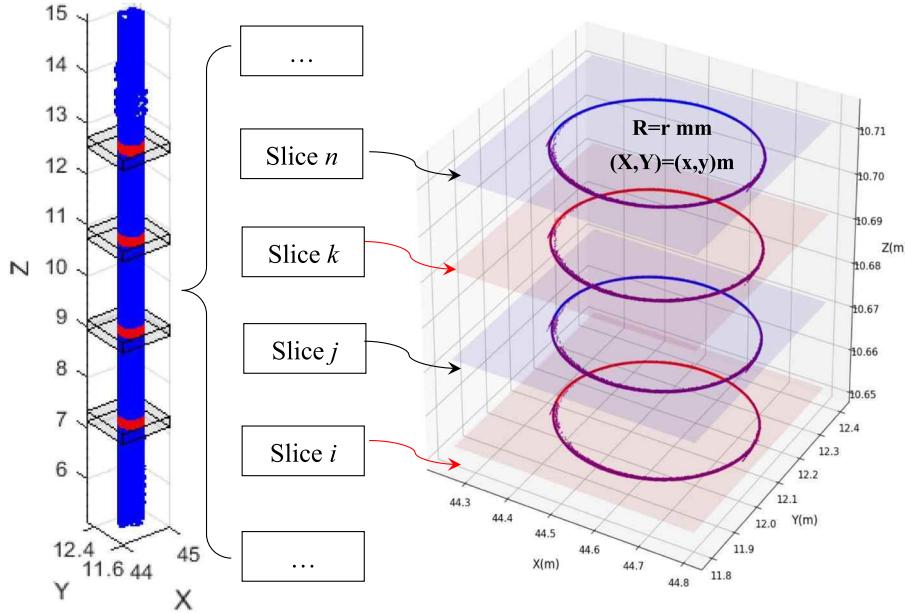


Fig. 8. Automatic fitting the external parameter of bridge components.

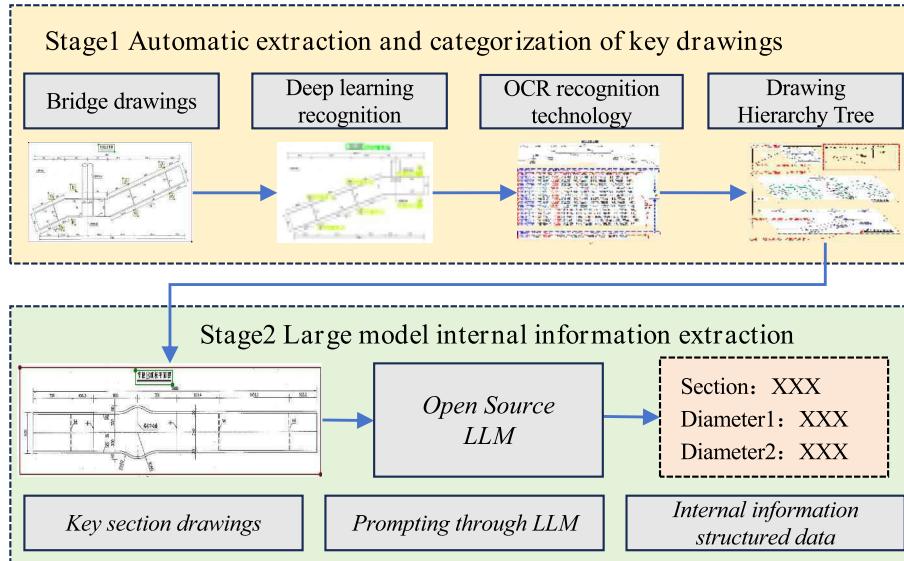


Fig. 9. A two-step fast method for obtaining internal dimensions of structures based on deep learning and large language modeling.

4.1. Deep learning-based automatic classification method for section drawings

Based on the measurement method using point cloud models, key external geometric features of structures can be obtained; however, the internal cross-sectional dimensions are difficult to capture. This limitation is particularly evident in bridge structures, which often comprise a large number of hollow cross-sectional components (such as box girders). Since point cloud data alone cannot provide internal geometric details, additional sources of information are required. Therefore, in this study, a deep learning-based approach is explored to achieve the automated retrieval of structural drawings, assuming that the corresponding documentation is available. This method enables the rapid identification of component-specific drawings necessary for internal dimension measurements, thereby reducing the manual workload associated with drawing review and supporting the extraction of critical internal information. Given that permanent deformations may occur during long-term

operation and maintenance, external dimensions are not derived from design drawings but rather obtained from point cloud measurements, ensuring alignment with the actual structural state.

Based on the measurement method utilizing point cloud models, key external geometric features of structural components can be acquired; however, internal cross-sectional dimensions remain difficult to determine. This limitation is particularly prominent in bridge structures, which frequently incorporate numerous hollow cross-section elements (such as box girders). Since point cloud data alone cannot capture internal geometries, essential structural details may be omitted. Therefore, in this study, a deep learning-based approach is investigated to enable the automated retrieval of engineering drawings, provided that such documentation is available. This method facilitates the efficient identification of component-specific drawings necessary for internal dimension extraction, thereby substantially reducing the manual effort involved in drawing inspection and supporting the accurate retrieval of internal geometric information. It is noteworthy that, due to the

potential for irreversible deformations arising from long-term service conditions, external dimensions are not obtained from original design drawings. Instead, they are extracted from point cloud data to ensure consistency with the actual structural configuration.

As illustrated in Fig. 10, the automated drawing retrieval framework is organized into three hierarchical stages. The first stage involves the extraction of graphical elements; the second entails the application of optical character recognition (OCR) for identifying relevant textual information within key graphical regions; and the third stage incorporates the construction of a hierarchical drawing tree structure based on the recognition results, thereby enabling automated classification and retrieval of component-related drawings. The algorithmic methodologies corresponding to these three modules will be elaborated upon in the subsequent sections.

4.1.1. Graphic extraction

The so-called “graphic extraction” refers to locating the corresponding graphic elements, which can be categorized as a target detection problem. For bridge construction drawings, three types of targets need to be identified: graphic blocks, drawing titles, and section symbols. In this context, graphic blocks are defined as the combination of each drawing title and its corresponding content. The YOLOv5 series of target detection models has demonstrated high recognition rates and detection accuracy, with validation across large databases indicating its capability for industrial applications. Additionally, the YOLOv5 series models exhibit excellent performance in multi-platform portability. The pre-trained model is based on the open-source YOLOv5. The target detection process is structured as follows: the original drawing is input into the system, converted into multiple page-level images, followed by the detection of graphic blocks. Based on the results of graphic block detection, drawing titles and section symbols are subsequently identified. The YOLOv5 network architecture is employed, using the PyTorch deep learning framework, with CUDA version 11.1. The training parameters are configured as follows: momentum of 0.937, weight decay of 0.0005, learning rate of 0.01, and training conducted over 1000 epochs.

A total of 20 sets of construction drawings and 280 valid images were collected to form the dataset for detecting graphic blocks, while 840 valid images were collected for the dataset for detecting drawing titles and section symbols. During the dataset creation process, Labelme was used for annotation, generating JSON annotation files, which were then converted into VOC format data for training. The dataset is shown in Fig. 11, where Fig. 11(a) represents the annotated data, and Fig. 11(b) shows the disqualified data that has been filtered out, including cases where the figure name is located below the figure block and where there is severe overlap between figure blocks.

For the task of detecting graphic blocks, the network image size was configured to 640 (with all images downsampled to 640×640 pixels for detection), and the batch size was set to 4. When training the network for detecting drawing titles and section symbols, given that the detection targets are relatively small, require a smaller receptive field, and

demand higher image clarity, the image size was increased to 1920, and the batch size was reduced to 2. The GPU used for training was an NVIDIA RTX 3090 with 24 GB of memory, and all networks were trained over 1000 epochs (where one epoch is defined as a complete pass through the entire training dataset).

4.1.2. Automatic recognition of drawing text using OCR recognition technology

Currently, significant achievements have been achieved in OCR recognition technology for Chinese characters; therefore, it is no longer necessary to train a network independently to accomplish this functionality. Instead, the Baidu Qianfan API is directly utilized to complete OCR text recognition. When invoking the API, authorization is first obtained from the Baidu official website, followed by the acquisition of access credentials, and finally, the interface is called using Python to implement the functionality. It supports ten languages, including Chinese, English, and mixed Chinese-English, with a character library exceeding 20,000 entries. Additionally, it provides positional information for the recognized text within the image. During the training process, specific optimizations were made for issues such as image blurriness, slanting, and flipping, resulting in strong robustness. Additionally, since the images input to the model are mostly clear frontal views, the detection complexity is relatively low, and there are no obscure characters in the drawings, this model is considered sufficient for the task of automatic text recognition in drawings.

The text is categorized into three types: the text associated with drawing titles, the numbers corresponding to the section symbols, and the text contained within the graphic blocks (excluding the drawing titles). Since the detection of graphic blocks, drawing titles, and section symbols has already been completed, it is only necessary to compare the positions of the recognized text with those of the detected elements to determine which texts correspond to drawing titles, which to section symbols, and which constitute in-graphic text. After recognition, regular matching is performed to extract the text that meets the established criteria. The text that complies with the set rules includes: text composed entirely of Chinese characters, text made up of Chinese characters and English letters, and text consisting of Chinese characters, English letters, and numbers. The detection results are shown in Fig. 12. To enhance presentation, the schematic diagram of the main girder segmentation has been cropped and displayed.

4.1.3. Automatic categorization of drawings based on structural hierarchical trees

The purpose of this step is to establish a hierarchical structure based on the text within the drawings, thereby enabling the automatic classification of the drawings. The categories of semantic segmentation of the point cloud primarily include steel beams, tower columns, bridge piers, and cables (hereinafter collectively referred to as structural categories). The graphic blocks containing structural categories in the drawing titles are designated as the first-level drawings, and the corresponding structural categories are referred to as root nodes. This structured

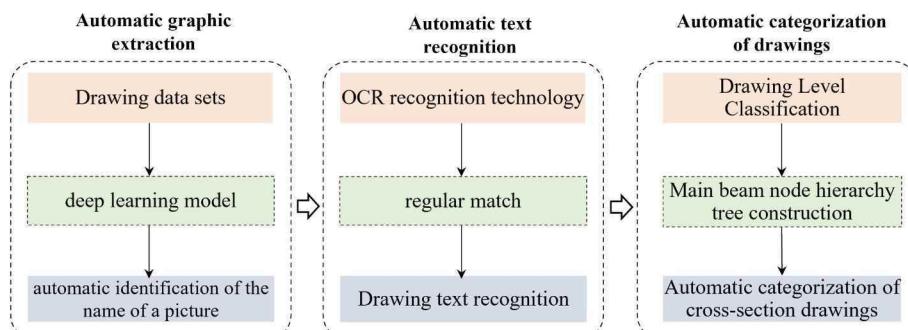


Fig. 10. Deep learning based automatic drawing recognition method.

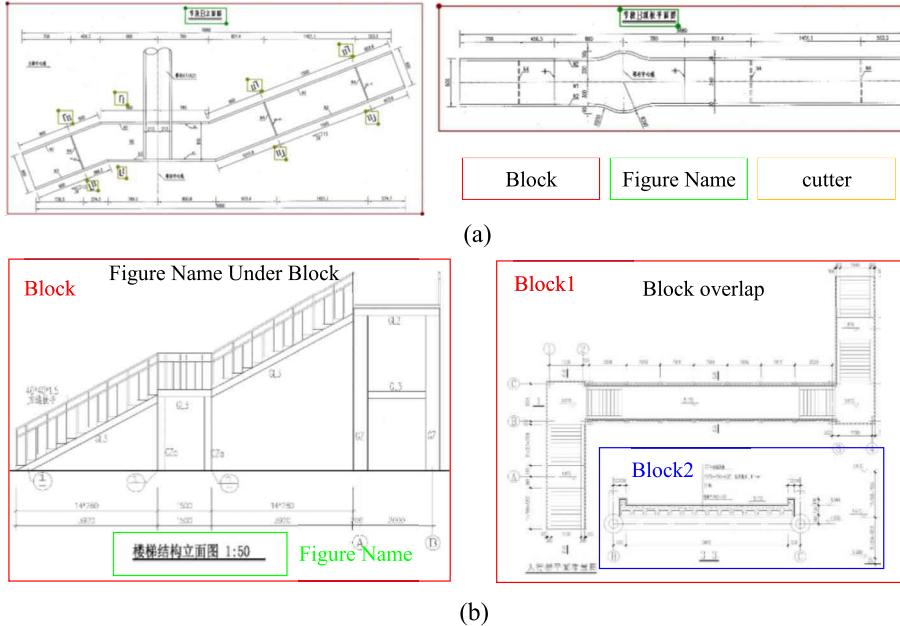


Fig. 11. Figure name and cutter recognition results: (a) Block recognition results; (b) Figure name and cutter recognition results.

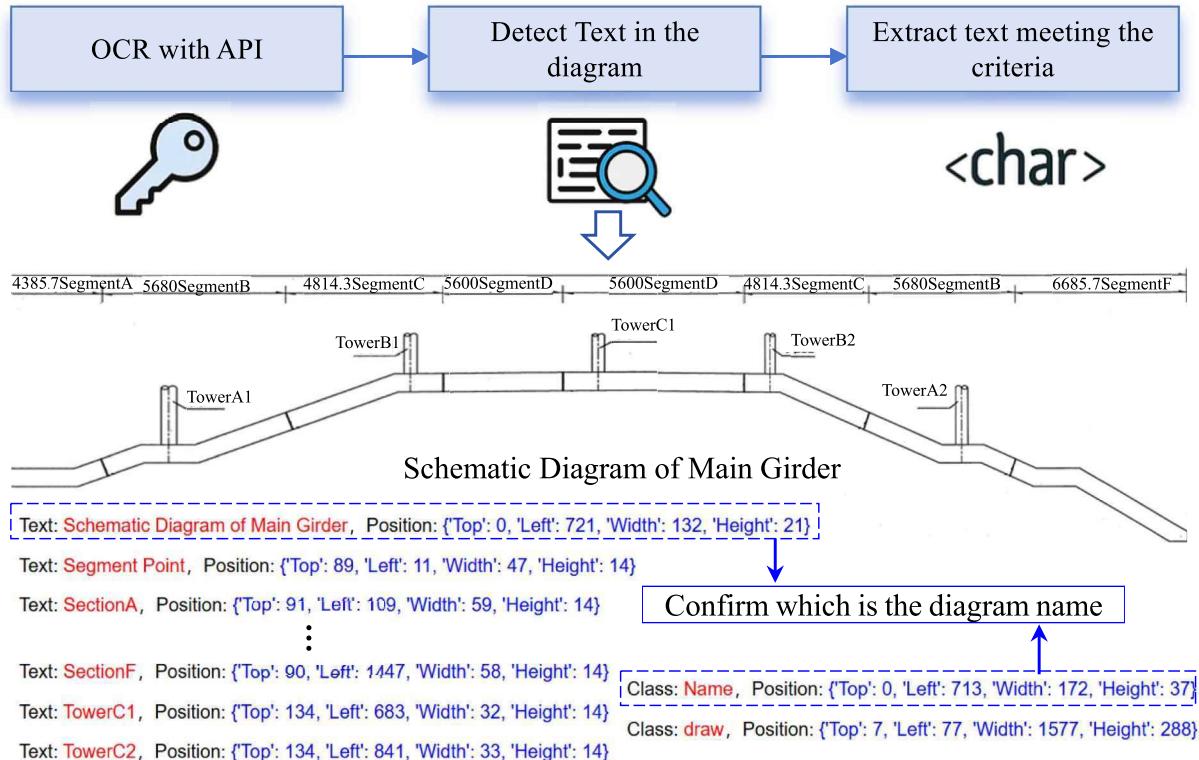


Fig. 12. Drawing text recognition using OCR recognition technology.

organization of the drawings facilitates clearer categorization, and the segmented point cloud model can be more effectively matched with the drawings.

The process of constructing the hierarchical tree of the main girder root nodes is illustrated in Fig. 13. First, after the first-level drawings are determined, the second-level drawings are sought based on these. The rule for searching is as follows: if a non-structural category name, such as Section A, appears within the first-layer graphic blocks, it is recorded under the corresponding first-layer graphic block to form a leaf node.

Subsequently, the graphic blocks that contain the text of the leaf nodes in their titles are classified as second-layer graphic blocks, and the search for third-layer graphic blocks is conducted accordingly, forming the hierarchical structure of the drawings. In this manner, the hierarchical structure of the drawings is progressively constructed.

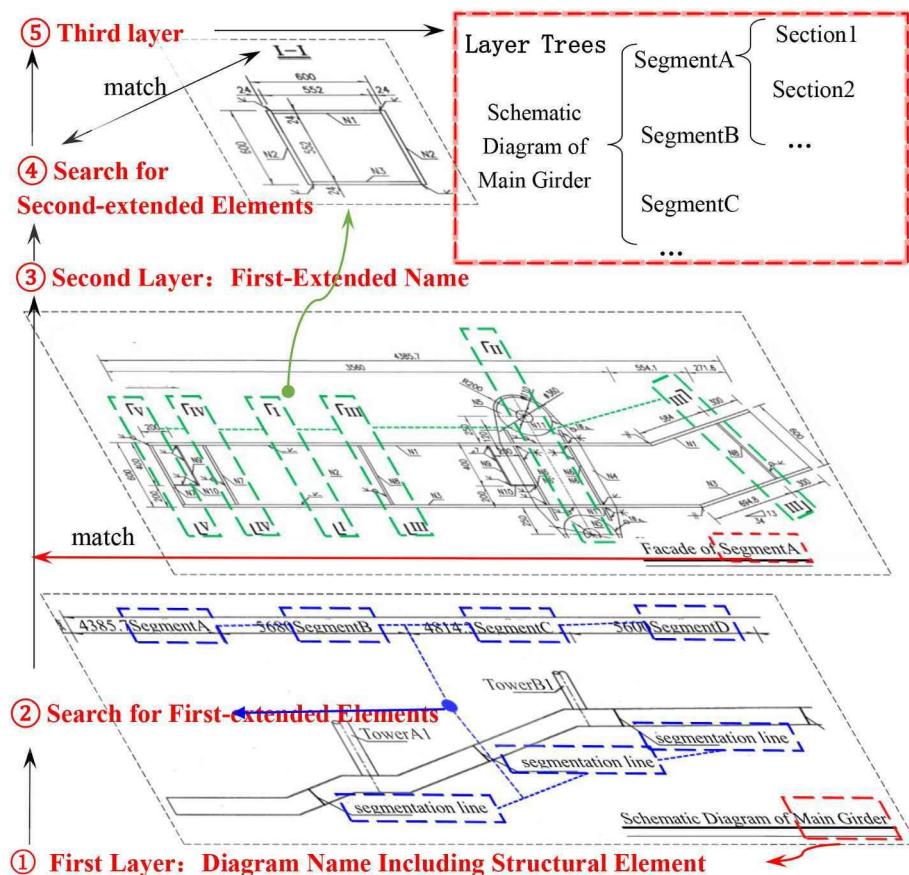


Fig. 13. Structural hierarchy tree schematic diagram.

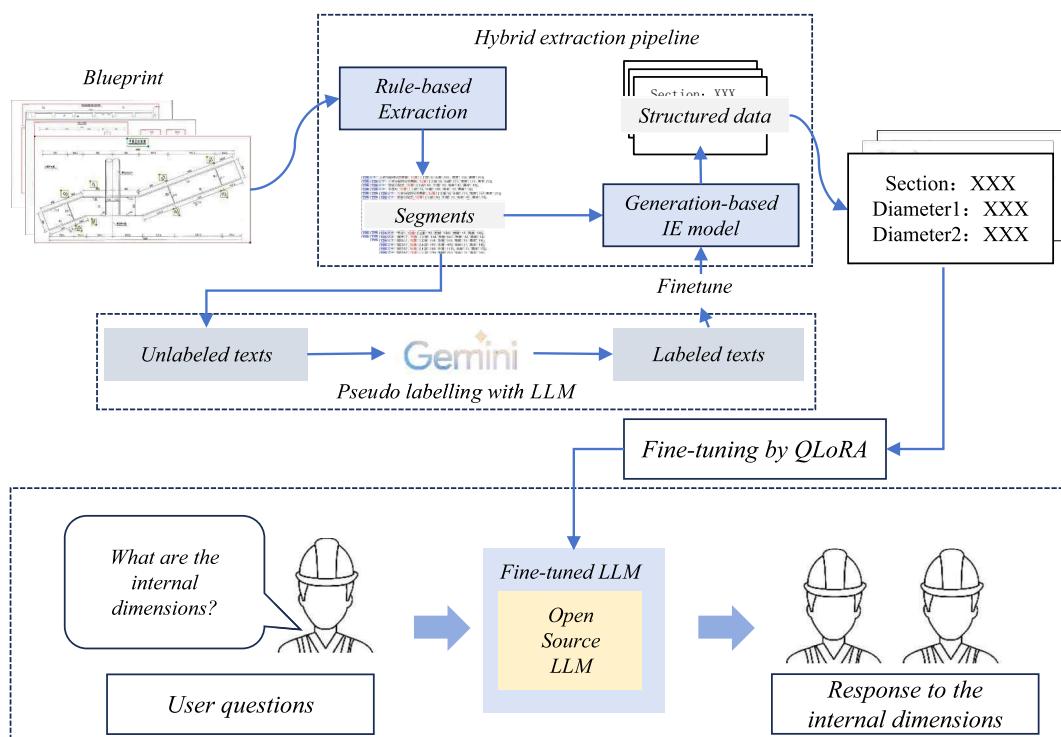


Fig. 14. Fine-tuned LLMs.

4.2. Automatic extraction of internal information of key sections based on fine-tuned macrolanguage modeling

The automatic classification of key section drawings has been achieved above; however, the automatic extraction of internal parameters from the drawings remains a critical challenge. Therefore, an automatic extraction method for internal information of key sections, based on the fine-tuning of large language models (LLMs), is proposed in this study to enhance the efficiency of information extraction. Although closed-source large language models (LLMs) like ChatGPT and Gemini have demonstrated outstanding performance under ideal conditions, their high costs and limited accessibility, particularly in regions such as China, constrain their large-scale application. In contrast, open-source models like GLM4 not only offer advantages in terms of cost and regulatory compliance but also exhibit strong capabilities in bilingual information extraction involving both Chinese and English. As illustrated in Fig. 14, GLM4 has been utilized as the core component, and an automatic extraction method for internal information of key sections based on fine-tuning large language models has been proposed to enhance accuracy and efficiency.

Despite the abundance of drawing data resources, obtaining high-quality labeled training data in specific domains continues to present considerable challenges. In particular, within the field of bridge inspection, manually extracting information from engineering drawings and assigning labels is both labor-intensive and time-consuming. To address this limitation, a pseudo-labeling method based on large language models (LLMs) is proposed in this study, aiming to reduce the manual labeling workload by automatically generating high-quality labels. Closed-source LLMs such as Gemini are utilized to generate pseudo-labels from a dataset of bridge drawings. To further evaluate the quality of the generated labels, a few-shot prompting strategy is adopted, in

which a small amount of manually labeled drawing data is used to guide the LLM in producing more accurate annotations. Finally, the performance of the pseudo-labeling approach is assessed by comparing the automatically generated labels with those obtained through manual annotation.

In order for the closed-source LLM to interpret the drawings and generate descriptions of structural shapes and dimensions, three categories of prompts have been designed in this study: (1) creating a drawing parsing dataset; (2) extracting geometric and dimensional information; and (3) generating structured descriptions. Each prompt comprises four components: role and task, reasoning chain, output format, and considerations. An iterative mechanism incorporating human feedback is employed, referred to as the Prompt Development Cycle (PrDC).

As shown in Fig. 15(b), the PrDC process is structured into four steps. First, professional roles such as building designers or structural engineers are assigned, accompanied by general instructions for identifying component types, geometric shapes, and dimensional attributes. Second, specific task directives are provided to guide the interpretation of dimension annotations and geometric symbols within the drawings. Third, a Chain-of-Thought (CoT) strategy is employed to decompose complex parsing tasks into sequential reasoning steps. This process leverages contextual information, including drawing types and annotation standards, to support the large language model (LLM) in performing accurate inference. Finally, the parsing results are evaluated, and errors are iteratively corrected to refine model performance. The fine-tuned model is capable of automatically recognizing geometric features, dimensional annotations, and material specifications in the drawings. It generates structured outputs in either JSON format or natural language, which include detailed descriptions of component types, geometry, and dimensions. This approach significantly improves the efficiency and

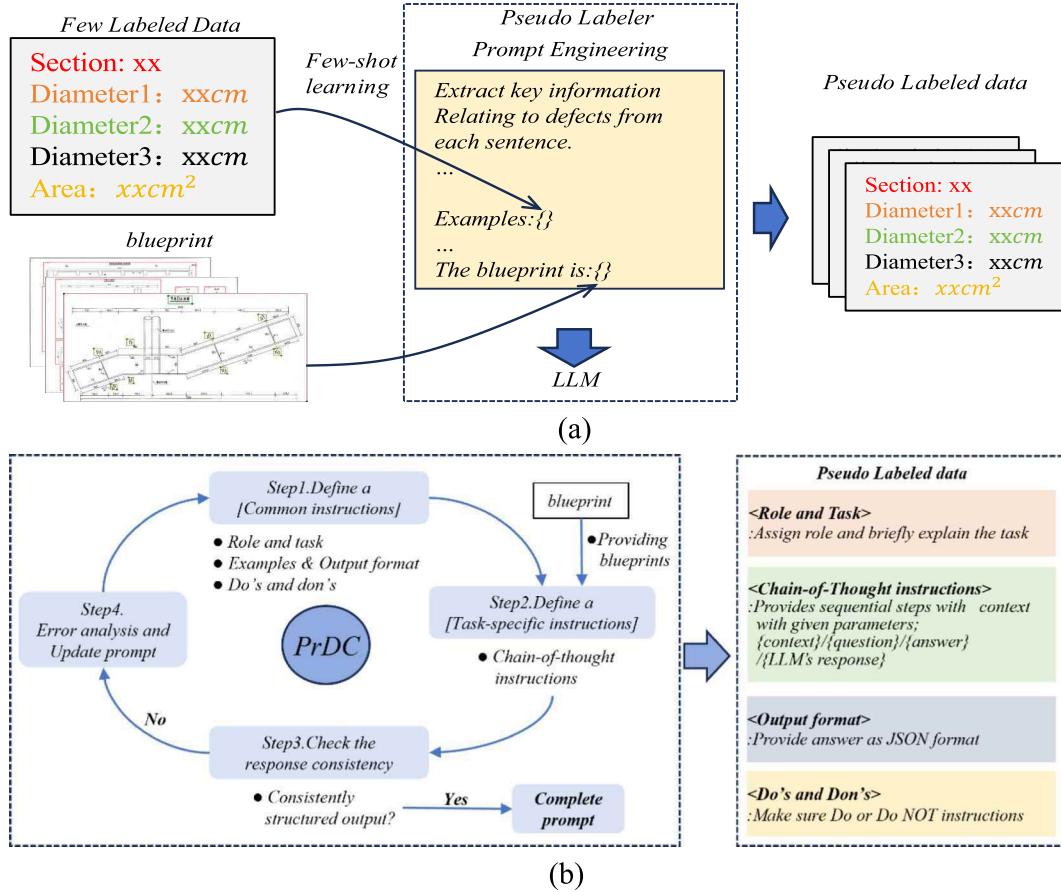


Fig. 15. Methods for generating training data based on closed-source models: (a) Label generation methods; (b) Prompt development cycle.

accuracy of drawing interpretation and is well-suited for intelligent recognition tasks involving complex engineering drawings.

Although GLM4 demonstrates strong general capabilities, its direct application to domain-specific drawing information extraction tasks has shown limited effectiveness. Therefore, additional fine-tuning is required to enhance its effectiveness in extracting defect-related data from engineering drawings. To address this, the labeled text within the drawings is incorporated into the prompt inputs, and the structured JSON-format data is converted into string representations to serve as the model's target output. Subsequently, Quantized Low-Rank Adaptation (QLoRA) is adopted for model fine-tuning. As a result, the fine-tuned model is capable of generating structured JSON outputs or coherent natural language descriptions from drawing information, including component types, geometric characteristics, and dimensional details. The efficiency of the QLoRA approach enables accurate drawing recognition under resource-constrained conditions, offering a feasible and scalable pathway for the intelligent interpretation of complex engineering drawings.

5. Automated finite element modeling

After extracting both internal and external information, automatic finite element model transformation can be conducted. The transformation process is illustrated in Fig. 16. Initially, external and internal information is obtained based on point cloud and corresponding graphic. The alignment between internal and external information is achieved through component naming conventions. Taking bridge towers as an example, the tower labeled "towerA" in the drawings corresponds to point cloud-derived tower T1, with its relevant cross-section information assigned to T1. Similarly, "towerB" and "towerC" are mapped to T2 and T3 respectively. As illustrated in Table 1, algorithm 1 is

Table 1
Algorithm for split components at connection points.

```

Algorithm1 : Split components at connection points
Input: all components; external information of components.
Output: components for finite element model
components.start_center_coords = components.start_center_coords - G1.start_coords
// The definition of G1 refers to Table 4.
components.end_center_coords = components.end_center_coords - G1.start_coords
record=[ ]
for components[i] in components do
    coords1, coords2 = components[i].start_center_coords, components[i].end_center_coords
    record.append([coords1, coords2, components[i]])
    for record[j] in record[-1:] do
        a=get_area(record[j]) // determine the coverage area of components[i] using
        record[j].coords
    if coords1 in a then
        record[j]=[sort(record[j].coords,coords1), components[i]]
    if coords2 in a then
        record[j]=[sort(record[j].coords,coords2), components[i]]
new_components=generate_new_components(record)
return new_components

```

employed to split components based on their connection positions.

As illustrated in Table 2, this step ensures that connecting nodes between two components are not omitted during element creation. As demonstrated in Table 3, the algorithm 2 is employed to create elements. During this process, the external coordinates of each component are converted into key points, which are then compiled into an array. When a new key point is introduced, it is first verified against the array to check for any existing nodes at the same location. If a match is found, there is no need to store it again; instead, the information can be directly

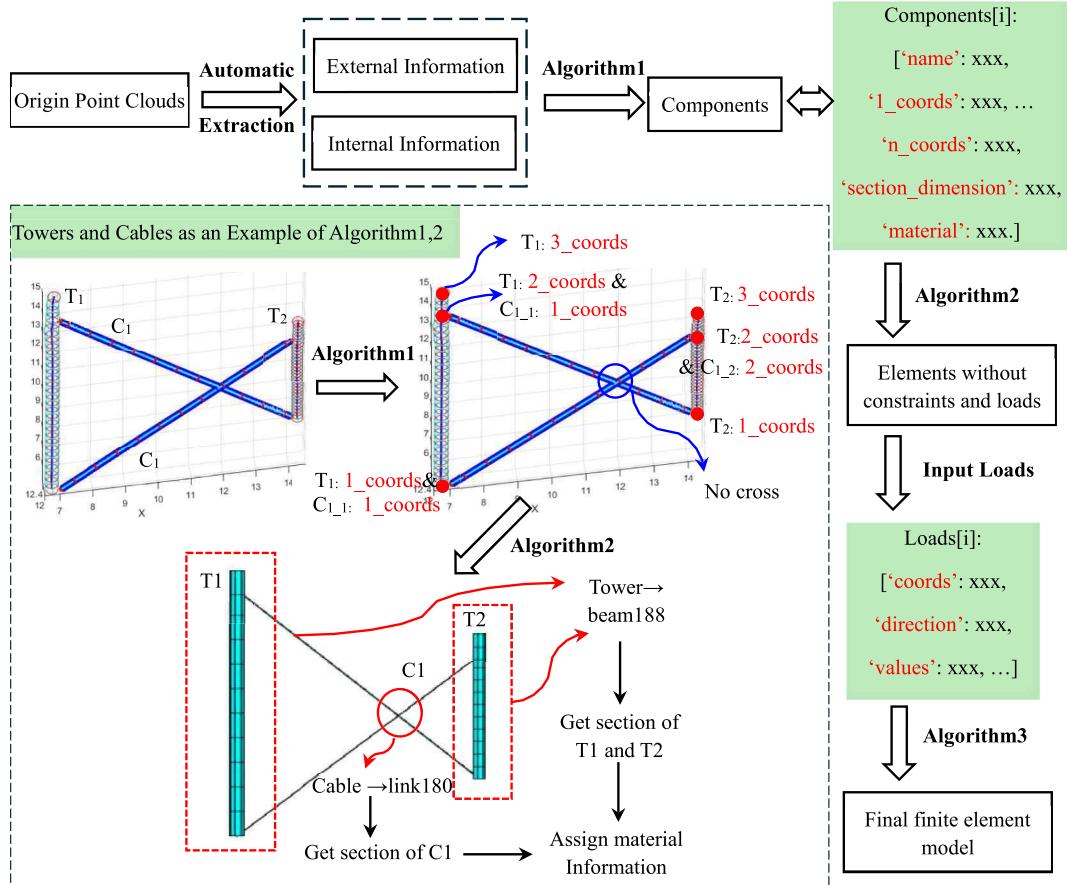


Fig. 16. Automatic finite element modeling.

Table 2

Algorithm for element generation.

Algorithm2 : Element generation

```

Input: all components; external information, internal information and material of components.
Output: Elements without constraints and loads
k_array=[ ] // Initialize an empty list
for components[i] in components do
    for coords[n] in components[i].coords do
        kn= coords[n].x, coords[n].y, coords[n].z
        if kn not in k_array then
            create keypoint[i*100+n] in Ansys
            k_array.append(kn)
        else
            index=findindex(k_array, kn) //find the index of kn in k_array
            keypoint[i*100+n]= k_array[index]
            if n > 1 then
                line[i*100+n-1]=connect(keypoint[i*100+n-1], keypoint[i*100+n])
                //connect two keypoints in Ansys
                if components[j*100+n-1]==girder, beam, link, tower or girder then
                    line[i*100+n-1].element_type=beam188
                    line[i*100+n-1].section=combine(components[i].external_dimensions,
                    components[i].internal_dimensions)
                else
                    line[i*100+n-1].element_type=link180
                    line[i*100+n-1].area=calculate (components[i].external_dimensions,
                    components[i].internal_dimensions)
                    line[i*100+n-1].material= components[i].material
                    line[i*100+n-1].mesh_num=distance( keypoint[i*100+n],
                    keypoint[i*100+n])/N
            Mesh all in Ansys
    return elements

```

Table 3

Algorithm for defining constraints and loads.

Algorithm3 : Assign constraints and loads

```

Input: elements, k_array, loads
Output: final finite element model
ks= k_array.where(z=z_min, tolerance=10cm) // find the key points in contact with the
ground
ks.displacement=0
for loads[i] in loads then:
    location=loads[i].location
    node=search_nearest( elements.node, location) //Search nearest element.node in
elements
    node.load=loads[i].value
return final finite element model

```

retrieved from the array. Once the key points are determined, the points are connected, and elements are assigned based on the component type. The final phase involves the definition and application of boundary conditions and loads, which is implemented using Algorithm 3. At component interfaces, since beam elements ensure node continuity, the co-located nodes generated in the first step are sufficient, eliminating the need for additional constraints. As presented in Table 4, for boundary conditions, all degrees of freedom are fixed at the node with the lowest Z-coordinate. Regarding load application, the system allows for the automatic specification of load positions and types based on coordinate information, assigning the loads to the nearest nodes of the corresponding finite elements, thereby enabling automated load configuration.

The pseudocode for the three algorithms is presented in the following sections. ANSYS APDL is employed for the generation of finite element models, with the modeling process fully automated through the automatic generation of APDL scripts. This automated modeling technique significantly reduces the time required for students to acquire finite element modeling skills and minimizes manual effort for engineers in

Table 4

Measurement results of critical arch bridge structural components.

Item	Calculated Value (mm)	Measured Value(mm)	Error(mm)
Girder	b _f : 126.9; t _f : 11.2; h: 124.4; t _w : 9.3; L: 4769.2	b _f : 125.0; t _f : 9.0; h: 125.0; t _w : 6.25; L: 4760.0	b _f : 1.9; t _f : 2.2; h: -0.6; t _w : 3.05; L: 9.2;
	b _f : 98.2; t _f : 10.3; h: 101.6; t _w : 8.1;	b _f : 100.0; t _f : 8.0; h: 100.0; t _w : 6.0; L: L: 684.0	b _f : -1.8; t _f : 2.3; h: 1.6; t _w : 2.1; L: 3.0;
	b _f : 100.5; t _f : 6.6;	b _f : 100.0; t _f : 8.0;	b _f : 0.5; t _f : -1.4;
	b _f : 102.7; t _w : 6.9;	b _f : 100.0; t _w : 6.0; L: L: 890.5;	b _f : 2.7; t _w : 0.9; L: 4.5
	R: 6.7; L: 227.6	R: 6.0; L: 230.0	R: 0.7; L: 2.4
	R: 6.7; L: 423.4	R: 6.0; L: 420.0	R: 0.7; L: 3.4
	R: 6.7; L: 552.2	R: 6.0; L: 550.0	R: 0.7; L: 2.2
	R: 6.7; L: 611.6	R: 6.0; L: 610.0	R: 0.7; L: 1.6
	R: 6.7; L: 641.2	R: 6.0; L: 636.0	R: 0.7; L: 5.2
	A R:35.9	R: 35.0	R:0.9
Link	L R:24.4; L: 890.5	R:25.0; L: 886.0	R:-0.6; L:4.5

Note: b_f is flange width; t_f means flange thickness; h means web depth, t_w means web thickness.

interpreting and processing structural drawings. More importantly, with the ongoing advancement of Artificial General Intelligence (AGI), this approach lays a solid foundation for enabling AGI systems to autonomously perform finite element modeling tasks, thereby facilitating a fully automated, human-free modeling workflow.

6. Model validation experiments

To verify the effectiveness of the proposed method, preliminary experiments were conducted on a scaled-down steel arch bridge model in a laboratory setting. Point cloud data of the model were acquired using terrestrial laser scanning, and intelligent recognition algorithms were employed to automatically identify and classify key structural components such as arch ribs and main girders. Subsequently, the extracted geometric and semantic information was used to automatically generate the finite element model of the bridge.

6.1. Overview of the experiment

As illustrated in Fig. 17, the bridge has a total length of 5.2 m and a width of 0.875 m. The structure is fabricated from Q235 steel and consists of tie rods, crossbeams, arch ribs, and supports. A total of nine displacement sensors were installed around the beam to measure the structural displacement response.

6.2. Analysis results for bridge structure

In Fig. 18, the key members of the steel arch bridge are illustrated. Due to the symmetry of the bridge, only the components on the right side of the bridge are labeled with letters. The components indicated in red represent the main girder, labeled as G. The components represented in blue represent the beams. For simplicity, B₁, B₂ are used to denote beams as there are only two distinct sizes for the beams. B₁ is shorter in length and is circled in Fig. 18. The components represented in green represent the slings, designated from left to right as S₁, S₂, ..., S₉. The components represented in pink are the arches, labeled as A. The cyan components are links, labeled as L, and the yellow components are anchors.

In the processing workflow for the arch bridge, the arches and links are classified as superstructure, whereas the beams and girders are considered substructure. Initially, these two structural systems are separated based on projection density along the z-axis. Subsequently, transverse projection density is employed to distinguish girders from beams, and arches from links. Finally, the slings are segmented along the longitudinal axis of the bridge using projection density analysis.

In Fig. 19, the fitting results of key structural sections extracted from

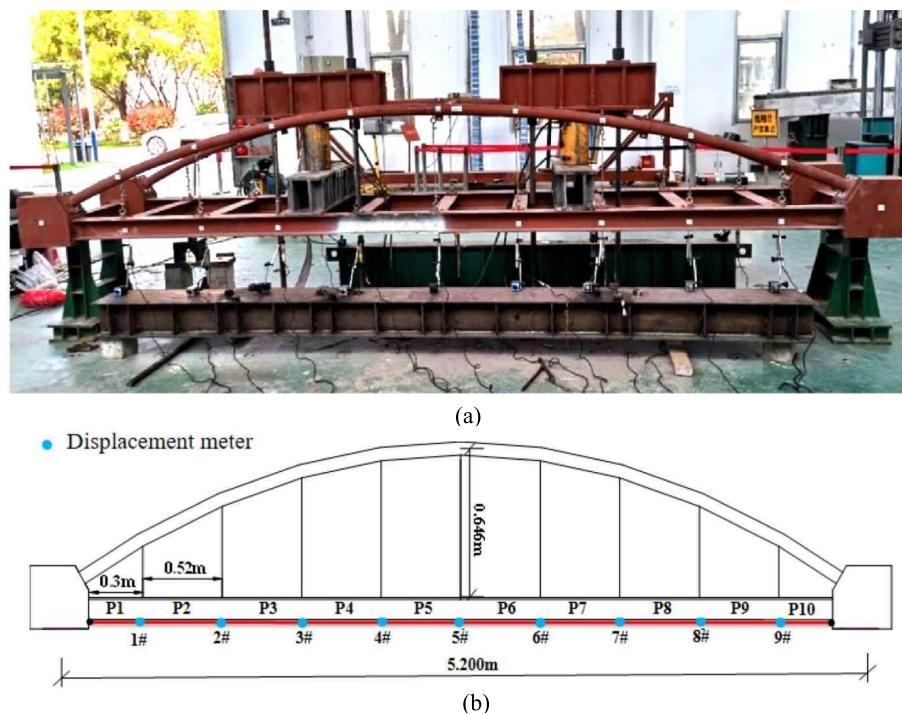


Fig. 17. Laboratory test site: (a) model; (b) sensor deployment.

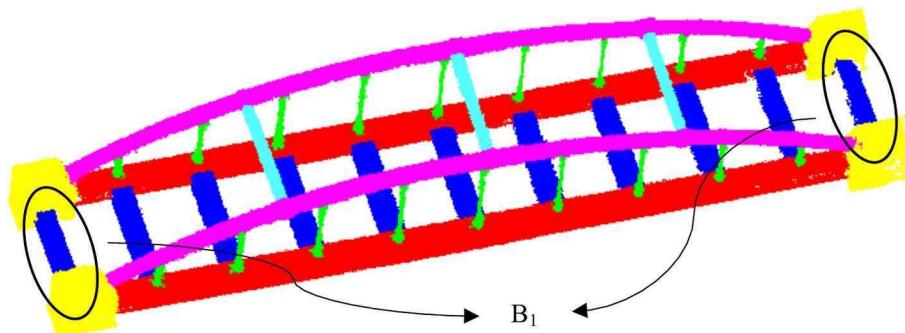


Fig. 18. Key members of the steel arch bridge.

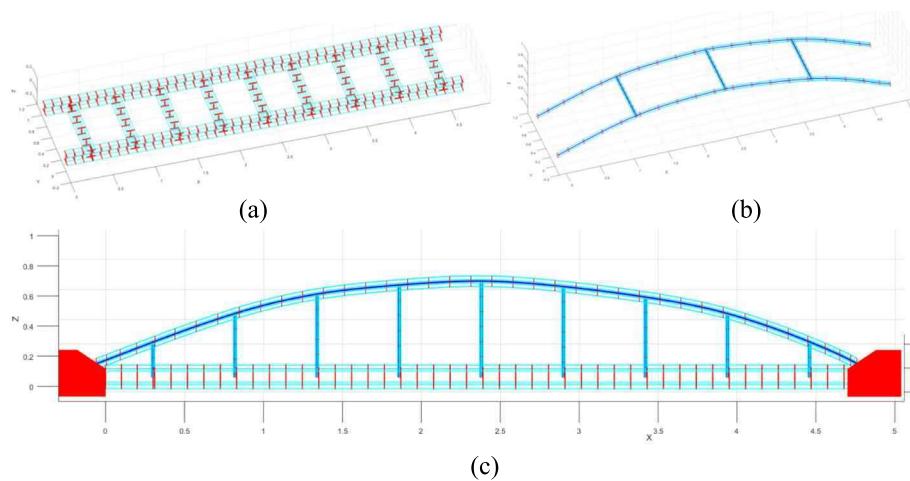


Fig. 19. Visualization of parameter fitting for key parts of the bridge (a) Girders and beams; (b) Arches and links; (c) Façade of the bridge.

point cloud data are presented, encompassing the geometric characteristics of the main girders, beams, slings, arches, and links. Since both the main girders and crossbeams exhibit I-shaped cross-sections, an I-beam section fitting method was adopted. Owing to the symmetry of the I-beams used in the structure, the upper and lower flanges were assumed to have identical dimensions. The anchors were fitted using a trapezoidal section fitting approach. For sections that do not conform to circular or rectangular shapes, a parameterized fitting procedure was employed based on optimization algorithms. Specifically, the target section was expressed as an organized point cloud in parametric form. The optimal parameters were determined by minimizing the distance between the observed point cloud and the parametric representation through iterative optimization techniques.

Table 4 provides the calculated values, measured values, and corresponding error analysis for the structural dimensions of key bridge components extracted from point cloud data. The radii of the slings were obtained by averaging the measurements across all individual sling members to ensure consistency. For the arches, the radius at the highest point was selected as the reference dimension, considering the inclination of the structure. In general, shape deviations were constrained within ± 3.05 mm, while length errors were controlled to be less than 9.2 mm. These results indicate that the geometric characteristics of critical structural elements can be accurately captured through the proposed point cloud processing method, thereby demonstrating its effectiveness for precision modeling in structural applications.

Fig. 20 illustrates the internal dimension identification results for four representative bridge components: arch rib, main beam, cross beam, and connection plate. Each subfigure compares manual interpretation with the model automated output, demonstrating its capability to accurately extract key geometric parameters. Fig. 20(a) presents a circular arch rib section, with an outer radius of 70 mm, inner radius of 62 mm, and edge thickness of 4 mm successfully identified. Fig. 20 (b) and (c) show I-beam sections of the main and cross beams, where the model accurately extracts the total height (125 mm and 100 mm), flange width (125 mm and 100 mm), flange thickness (9 mm and 8 mm), and web thickness (6.5 mm and 6 mm), respectively. Fig. 20 (d) highlights a beveled square hollow section of the connection plate, with outer dimensions of 333.39 mm, bevel length of 150.26 mm, and inset size of 255.96 mm \times 205 mm. Importantly, the wall thickness of 10 mm is precisely identified.

An automated finite element modeling process is established by integrating geometric data extracted from point clouds with internal dimensions obtained from structural drawings. Internal and external information is aligned via standardized component naming. Component segmentation is performed using algorithm 1, which identifies connection points to ensure topological consistency. Algorithm 2 is employed to

convert component geometries into finite element entities, wherein deduplicated key nodes are systematically organized and used for element generation. Algorithm 3 is responsible for automatically applying boundary conditions and structural loads, based on the spatial location and functional attributes of each node. The entire process is implemented in APDL through automatically generated scripts, achieving high-efficiency modeling with minimal manual intervention. The bridge structure of steel arch bridge is composed of Q235 steel, with an elastic modulus of 2×10^5 MPa, a density of 7.85 g/cm³, and a Poisson's ratio of 0.3. The suspension cables are characterized by a density of 2.68 g/cm³ and a Poisson's ratio of 0.33. Fig. 21 shows the finite element model of the bridge generated using the proposed method. The model includes the arch ribs, tie beams, cross beams, and deck panels, providing effective support for subsequent mechanical analysis.

To validate the accuracy of the proposed method, static analyses were conducted under two loading conditions: 20 kN (Case 1) and 63 kN (Case 2), with the loads applied at positions P4 and P7. As illustrated in Fig. 22(a) and (b), the simulated displacement profiles exhibit close agreement with the experimental measurements across the entire span, indicating high fidelity of the model under both load cases.

7. Filed validation experiments

To further validate the effectiveness of the proposed method in practical bridge engineering, this paper presents a comprehensive analysis of a real steel pedestrian bridge. The study systematically demonstrates and evaluates the performance of the proposed techniques, including component segmentation, cross-sectional fitting, automated drawing recognition, and automated finite element modeling, within an actual engineering context.

7.1. Overview of the experiment

As shown in Fig. 23, the footbridge investigated in this research is a cable-stayed bridge, with the main bridge being a mast-type cable-stayed structure. The main span of the bridge is 37.6 m, and the deck is constructed using an orthotropic plate with a total width of 5 m. The main pier features a bridge tower with a circular steel tube section of diameter 0.48 m and a height of 10 m above the deck. The central tower pole has a circular steel tube section, with height and diameter gradually decreasing. The main girder of the cable-stayed bridge is designed as a double-box steel beam section, with the side box section having a height of 0.6 m and a width of 0.6 m.

Several load conditions were conducted for this bridge, including static load tests and impact vibration tests induced by pedestrian

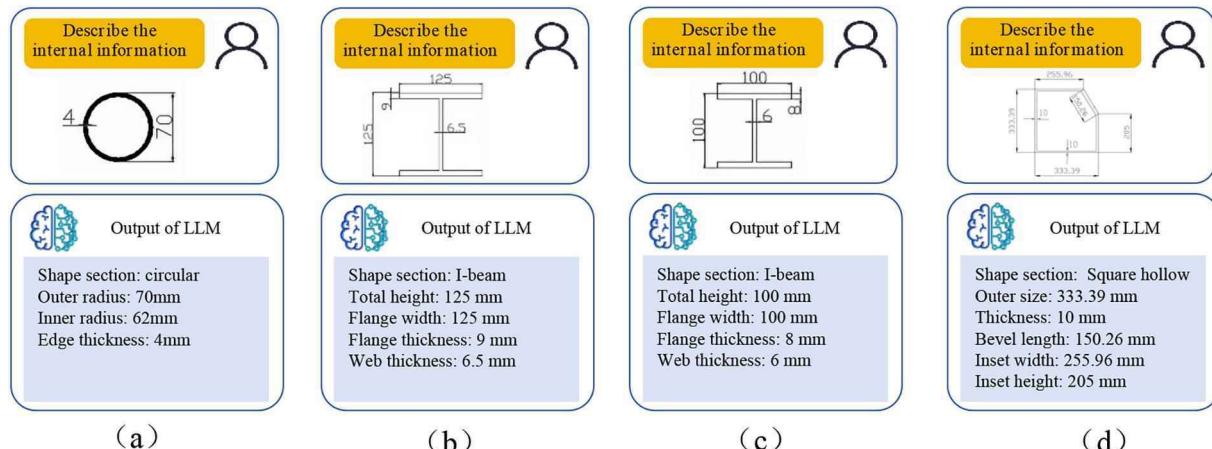


Fig. 20. Large model internal dimension identification results: (a) Arch rib section; (b) Main beam section; (c) Cross beam section; (d) Connection plate section.

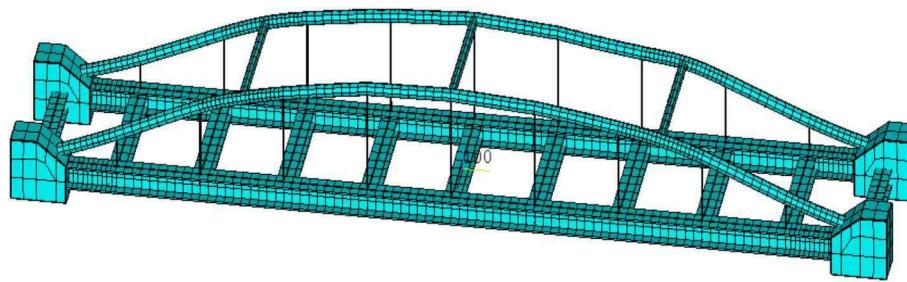


Fig. 21. Automated finite element modeling results.

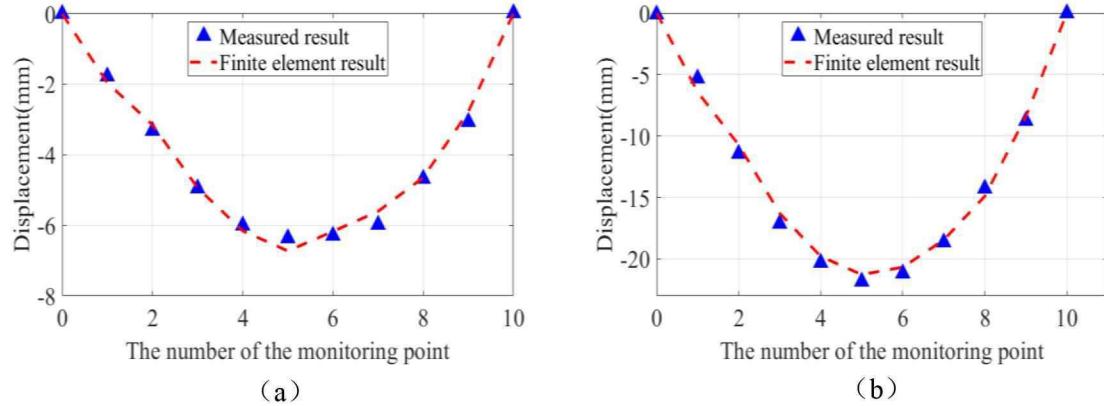


Fig. 22. Static analysis results: (a) Displacement under case1; (b) Displacement under case 2.



Fig. 23. Pedestrian bridge: (a) site plan; (b) sensor deployment plan.

jumping. Among these, two static load conditions were performed: Condition One involved applying an approximately 700 N static load at each measurement point, while Condition Two involved applying approximately 1200 N loads at measurement points 3, 4, and 5, respectively. Seven PCB piezoelectric accelerometers and industrial cameras were arranged at equal intervals on both sides of the footbridge to collect the dynamic responses of the structure under three testing

conditions, with a spacing of 3 m between the sensors. The scanning was performed using a RIEGL VZ-400i laser scanner, which produced a high-resolution point cloud model with an average point spacing of approximately 3 mm, providing precise geometric data for subsequent structural analysis.

7.2. Analysis results for bridge structure

Components bounded by the upper and lower planes of the main girder are designated as the deck system, as illustrated in Fig. 24(a). In this figure, the red regions indicate the superstructure, which includes the tower, cables, and handrails; the blue regions denote the deck system; and the green regions correspond to the substructure. In Fig. 24(b), the non-gray components represent the final parameterized geometric representations of the structural elements. Due to the bridge's symmetry, only the data for the right half of the bridge is presented.

To validate the effectiveness of the proposed method, comparisons are made with two advanced point cloud semantic segmentation models for bridges, as shown in Table 5. The precision of the point cloud segmentation for different bridge components is calculated based on the point cloud evaluation results, which means the fraction of instances that are correctly labeled as one category out of all instances labeled as that category. It is found that our method exhibits significant accuracy in detecting various bridge components, surpassing existing advanced methods based on bridge structural characteristics [31]. Moreover, it demonstrates superior segmentation accuracy compared to deep learning-based advanced algorithms [32], further indicating the effectiveness of the proposed method in this paper.

Fig. 25 presents the fitting results of the key structural sections of the bridge extracted from point cloud data, including the geometric characteristics of the main girder, crossbeams, braces, and stay cables. The fitting results for the main girder cross-section are shown in Fig. 25 (a), where the linear features of the girder have been accurately extracted through the analysis of high-precision point cloud data. Fig. 25 (b) illustrates the region containing the crossbeam braces, where the fitted sectional profiles successfully reconstruct the spatial configuration and connection relationships. In Fig. 25 (c), the fitting results for a representative cross-section of the stay cables are displayed, in which centroid fitting and line fitting methods are employed to determine the

connection points and intersection relationships of the stay cables. The identified intersection points are labeled as "Fitted intersection". The variables involved in the parameter fitting process, along with their corresponding values, are listed in Table 6.

Table 7 presents the calculated values, measured values, and error analysis of the structural dimensions of key components of the bridge extracted from point cloud data, including the main girder, tower, stay cables, crossbeams, and braces. Given the large number of key bridge components, the results in the table have been summarized by category based on consistent dimensions, with the measured values representing the average within each respective category. Overall, the error between the calculated and measured values for the location of the main girder has generally been controlled within ± 20 mm, indicating a high level of dimensional accuracy achieved by the proposed point cloud extraction method. The errors in the radius and height dimensions of the tower are relatively minor, particularly the height error, which is kept within ± 7 mm. The radius errors of the stay cables do not exceed ± 2 mm. The extraction results for the crossbeams and braces also demonstrate high precision, with the height and width errors of the crossbeam cross-section remaining within ± 1.1 mm, and the radius error for the braces being as low as 0.5 mm. These findings confirm that the proposed method can effectively extract dimensional features of the main girder, tower, stay cables, crossbeams, and braces. The extracted values closely align with manual measurements, thereby enhancing the efficiency of external dimension acquisition and providing robust data support for automated finite element modeling.

The detection results for graphic blocks, drawing titles, and section symbols are shown in Fig. 26. Fig. 26 (a) illustrates the detection outcomes for the graphic blocks, while Fig. 26 (b) presents the detection results for the drawing titles and section symbols. Due to the close spatial proximity between the section symbols and the dimension lines, some detected section symbols may partially include dimension line segments, and certain section symbols may appear tilted. Nevertheless,

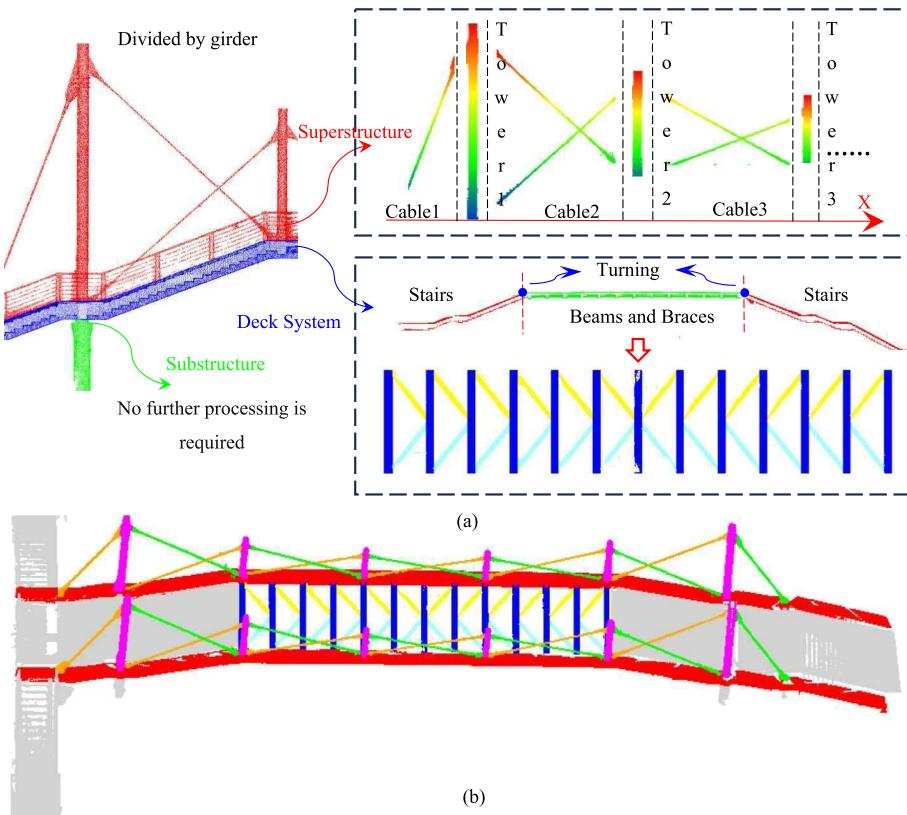


Fig. 24. Automatic segmentation results of bridge: (a) Secondary segmentation results for key bridge components; (b) Full bridge segmentation results.

Table 5

Comparison of segmentation precision of proposed methods.

Method	Main Girder	Transverse Beam	Diagonal Brace	Tower	Cable	Pier	Mean
Proposed	0.979	0.984	0.995	0.968	0.993	0.987	0.984
Algorithm 1[31]	0.905	0.923	0.924	0.959	0.989	0.977	0.946
Algorithm 2[32]	0.965	0.979	0.987	0.965	0.990	0.982	0.978

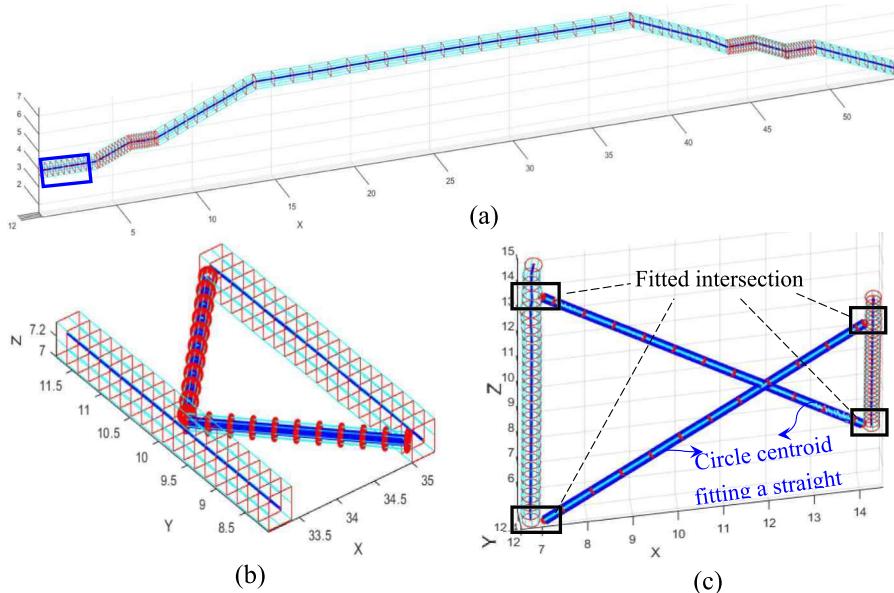


Fig. 25. Visualization of parameter fitting for key parts of the bridge: (a) Main girder; (b) Cross girder diagonal braces; (c) Diagonal cables.

Table 6

Parameter fitting session related variables and value.

Application stages	Variable meaning	Variable name	Variable value
Tower section fitting	Slicing step size	Δ_{tower}	Tower height 1/30
	Slicing thickness	Δc_1	0.02 m
Cable section fitting	Slicing step size	Δ_{cable}	Total stay cable length 1/10
	Slicing thickness	Δc_2	0.1 m
Main girder section fitting	Slicing step size	Δ_{girder}	Length of each segment 1/20
	Slicing step size	Δc_3	0.02 m
Transverse beam section fitting	Slicing thickness	Δ_{beam}	Transverse beam length 1/20
	Slicing step size	Δc_4	0.02 m
Diagonal brace section fitting	Slicing thickness	Δ_{link}	Total diagonal brace length 1/10
	Inlier distance threshold	Δc_5	0.1 m
Section RANSAC Fitting	Neighborhood Radius	$thre_rec$	0.002 m
	Minimum Points in Neighborhood	$DBSCAN_R$	Median 1/10 of the cross-section parameter list
Anomaly Filtering with DBSCAN	Minimum Points in Neighborhood	$DBSCAN_NUM$	

these interfering factors do not significantly impact the subsequent recognition and interpretation of the section symbols.

After constructing the structural hierarchy tree, relevant documents can be generated, and the corresponding hierarchical files can be searched based on the names of each structure, thereby enabling the automatic retrieval of drawings layer by layer. This hierarchical tree search approach offers an effective means to parse bridge structural drawing data, thereby supporting efficient information management for the geometric modeling and finite element analysis of the main girder and other structural components. Fig. 27 presents the internal

Table 7

Measurement results of critical structural components.

Item	Calculated Value (mm)	Measured Value (mm)	Error(mm)
Girder	G ₁ : L:3556.2; H: 651.9; W:598.5	L:3560.0; H:650; W:600	L:-3.8; H:1.9; W:1.5
	G ₂ : L:2197.8; H:652.1; W:601.3	L:2184.0; H:650; W:600	L:13.8; H:2.1; W:1.3
	G ₃ : L:1673.7; H:650.3;	L:1660.0; H:650; W:600	L:13.7; H:0.3; W:0.9
	G ₄ : L:6556.5; H:648.6; W:600.5	L:6552.1; H:650; W:600	L:4.4; H:-1.4; W:0.5
	G ₅ : L:11987.4; H:649.2;	L:12000.6; H:650; W:602.8	L:-13.2; H:-0.8; W:2.8
	G ₆ : L:12007.6; H:652.3; W:600.1	L:12000.6; H:650; W:600	L:7.0; H:2.3; W:0.1
	G ₇ : L:6542.4; H:648.5; W:597.4	L:6552.1; H:650; W:600	L:-9.7; H:-1.5; W:2.6
	G ₈ : L:1651.2; H:650.6;	L:1660.0; H:650; W:599.3	L:-8.8; H:0.6; W:0.7
	G ₉ : L:2202.7; H:652.7;	L:2184.0; H:650; W:600	L:18.7; H:2.7; W:0.4
	G ₁₀ : L:1650.9; H:651.3; W:601.3	L:1660.0; H:650; W:600	L:-9.1; H:1.3; W:1.3
	G ₁₁ : L:4686.0; H:652.2; W:598.3	L:4695.7; H:650; W:600	L:-9.7; H:2.2; W:1.7
Tower	T ₁ : R:211.0; H:10006.7	R:213.0; H:10000.0	R:-2.0; H:6.7
	T ₂ : R:175.8; H:5001.3	R:175.5; H:5000.0	R:0.3; H:1.3
	T ₃ : R:149.7; H:3503.7	R:149.5; H:3500.0	R:0.2; H:3.7
Cable	C ₁ : R:23.2	R:24.5	R:-1.3
	C ₂ : R:32.1	R:31.5	R:0.6
	C ₃ : R:19.2	R:19.5	R:-0.3
Beam Link	B ₂ : H:449.9; W:298.9 L: R:76.5	H:450.0; W:350.0 R:76.0	H:-0.1; W:-1.1 R:0.5

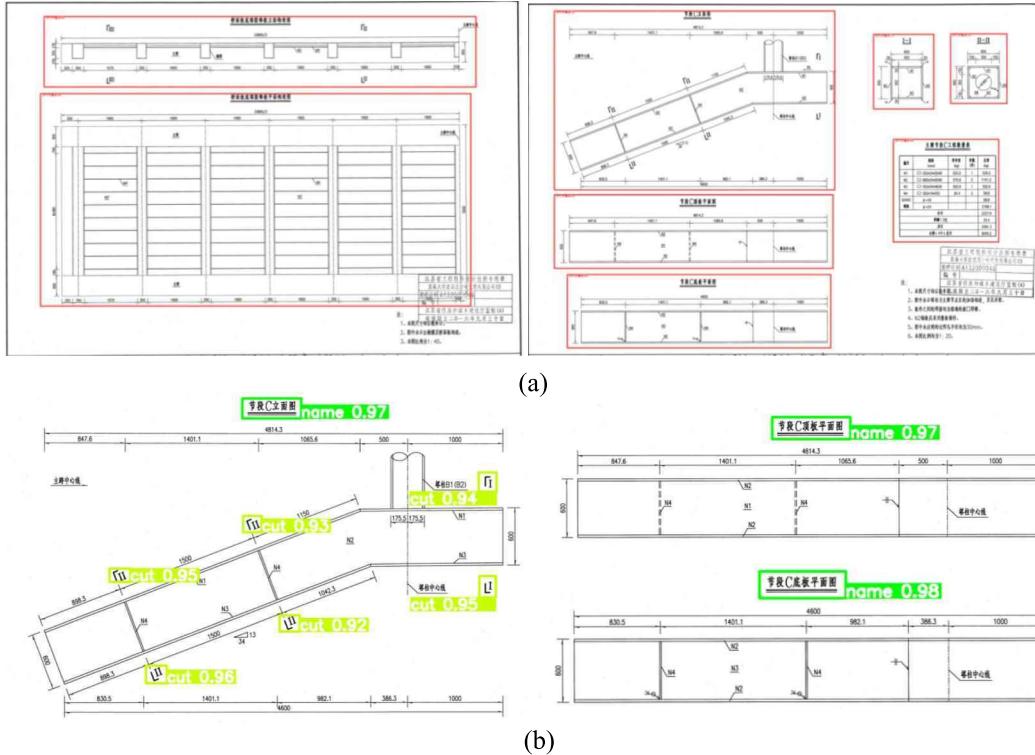


Fig. 26. Figure name and cutter recognition results: (a) Block recognition results; (b) Figure name and cutter recognition results.

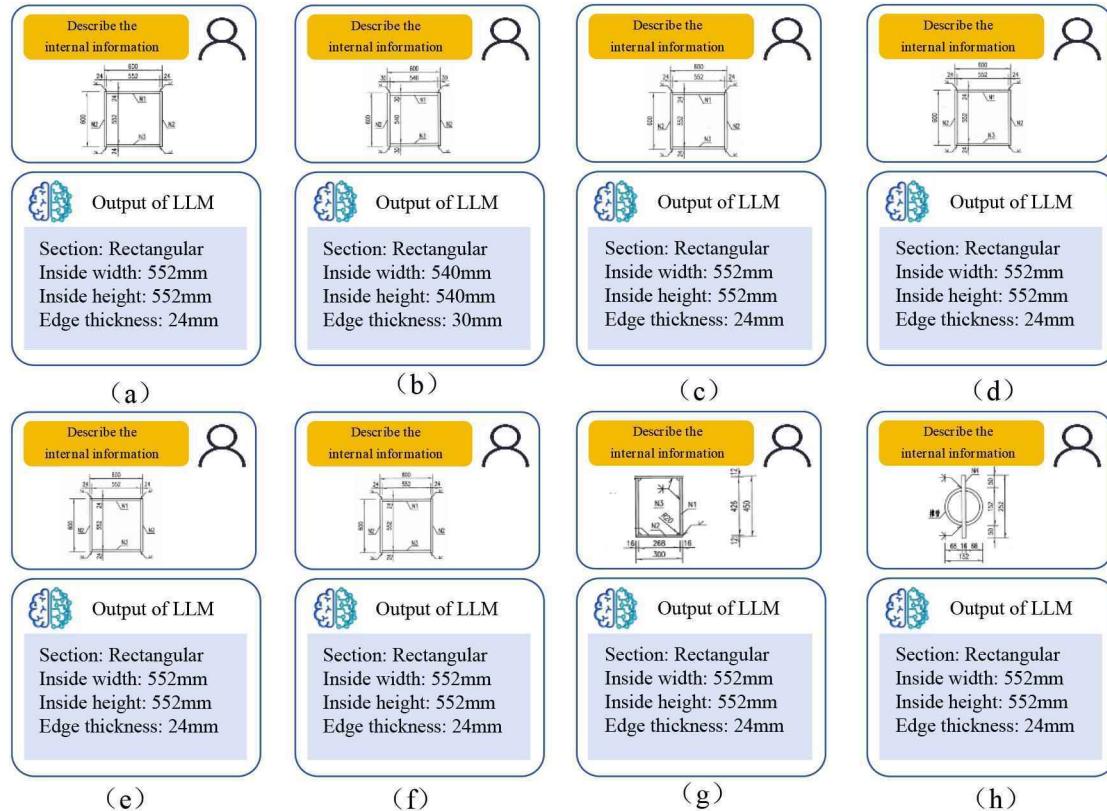


Fig. 27. Large model internal dimension identification results: (a) Main girder A section; (b) Main girder B section; (c) Main girder C section; (d) Main girder D section; (e) Main girder E section; (f) Main girder F section; (g) Transverse girder.

dimension recognition results of large structural components based on a large language model (LLM). Different structural cross-sections are included in the figure, such as the main girder, crossbeams, braces. Each subfigure compares the manually described internal structural information with the output results generated by the LLM. The LLM is able to accurately identify key geometric parameters, such as shape, internal width, internal height, outer diameter, and edge thickness, encompassing various cross-sectional forms including rectangular, box, and circular shapes. The results indicate that this method has a high applicability in the extraction and interpretation of internal dimensions from engineering drawings.

Fig. 28 illustrates the finite element model automatically generated based on point cloud data and bridge drawing information. The modeling process begins with the discretization of the main girder using three-dimensional solid elements (e.g., SOLID elements), accurately capturing the geometric and spatial configuration of the bridge deck and web. A bridge tower model is subsequently constructed, utilizing either solid or beam elements to ensure faithful representation of its load-bearing and stiffness characteristics. According to design parameters, stay cables are modeled using cable elements (e.g., LINK180), and are connected to the main girder and tower to simulate the cable force transmission mechanism. Following the completion of the superstructure, a pier model is established using solid or beam elements, and is linked to the main girder through support or rigid connection elements to reproduce realistic boundary conditions.

To automate this process, three custom algorithms are embedded within the framework. Algorithm 1 segments structural components based on connection locations, ensuring the completeness of critical nodes. Algorithm 2 converts component coordinates into key node arrays and generates finite elements according to structural type. Algorithm 3 automatically applies boundary conditions and loads: all degrees of freedom are constrained at the node with the lowest Z-coordinate, and user-defined loads are applied to the nearest finite element nodes based on spatial coordinates. All structural components except cables are modeled using BEAM188 elements, while cables are modeled with LINK180 elements. Sectional properties are extracted directly from the engineering drawings. The material properties for BEAM188 include an elastic modulus of 2.06×10^5 MPa, Poisson's ratio of 0.28, and a density of 7850 kg/m³. For LINK180 elements, the elastic modulus is 1.95×10^5 MPa, with a Poisson's ratio of 0.30 and the same density. This

modeling strategy achieves fully automated, high-precision finite element modeling of complex bridge structures, laying a solid foundation for subsequent structural health monitoring, dynamic analysis, and comprehensive structural assessment.

Regarding computational efficiency, the proposed algorithm incurs the greatest time consumption in two primary stages: point cloud segmentation and structural information extraction. Under the parameter settings adopted in this study, the complete workflow from raw point cloud input to fully segmented and annotated structural data requires approximately 2 to 3 h. The runtime is affected by several factors, including point cloud density and scale, segmentation granularity, and hardware performance. It is important to note that this study focuses exclusively on the automated finite element modeling process and does not consider the point cloud acquisition phase.

Fig. 29 (a) displays the measured data and finite element simulation results for Condition One, while **Fig. 29** (b) shows the measured data and finite element simulation results for Condition Two. Measurement results obtained by shock vibration method. The comparison results indicate that the trends in displacement variation between the finite element simulation results and the measured data are generally consistent under different loading conditions, thereby validating the accuracy of the established finite element model in static analysis.

To verify the dynamic accuracy of the established finite element model, a comparative modal analysis was carried out between numerical predictions and experimental measurements. The first three natural frequencies and corresponding mode shapes were obtained from hammer excitation tests conducted at symmetric points of the footbridge. As illustrated in **Fig. 30**, a good agreement is observed between the simulated and tested mode shapes. Specifically, the first three natural frequencies measured from the test are 2.68 Hz, 6.39 Hz, and 9.72 Hz, while those computed from the finite element model are 2.52 Hz, 5.86 Hz, and 9.67 Hz, respectively. These results demonstrate that the developed model can accurately capture the dynamic characteristics of the footbridge, thereby validating its applicability for subsequent structural performance evaluation.

8. Conclusions and future work

This paper presented an automatic finite element modeling method for steel bridges that integrated three-dimensional point cloud data and

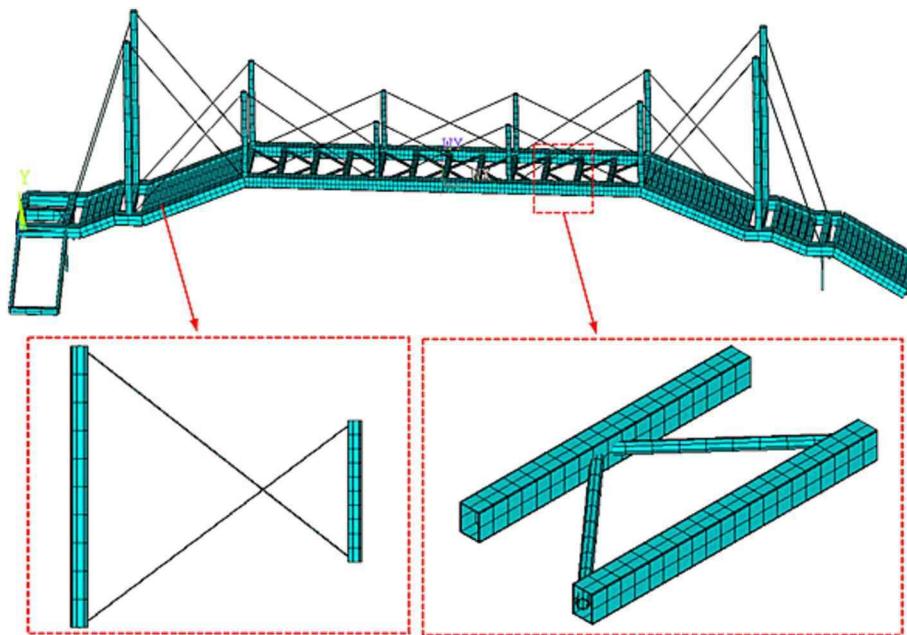


Fig. 28. Automated finite element modeling results.

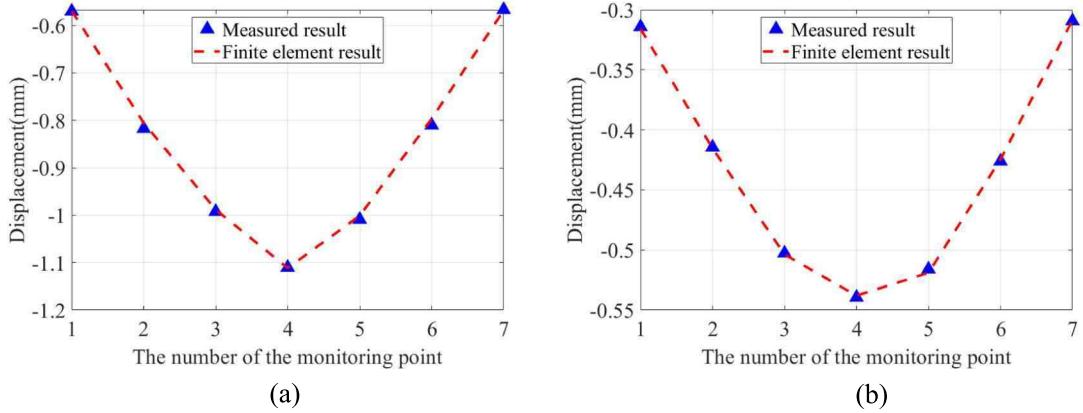


Fig. 29. Measured and finite element calculation results of bridge static force: (a) Case 1; (b) Case2.

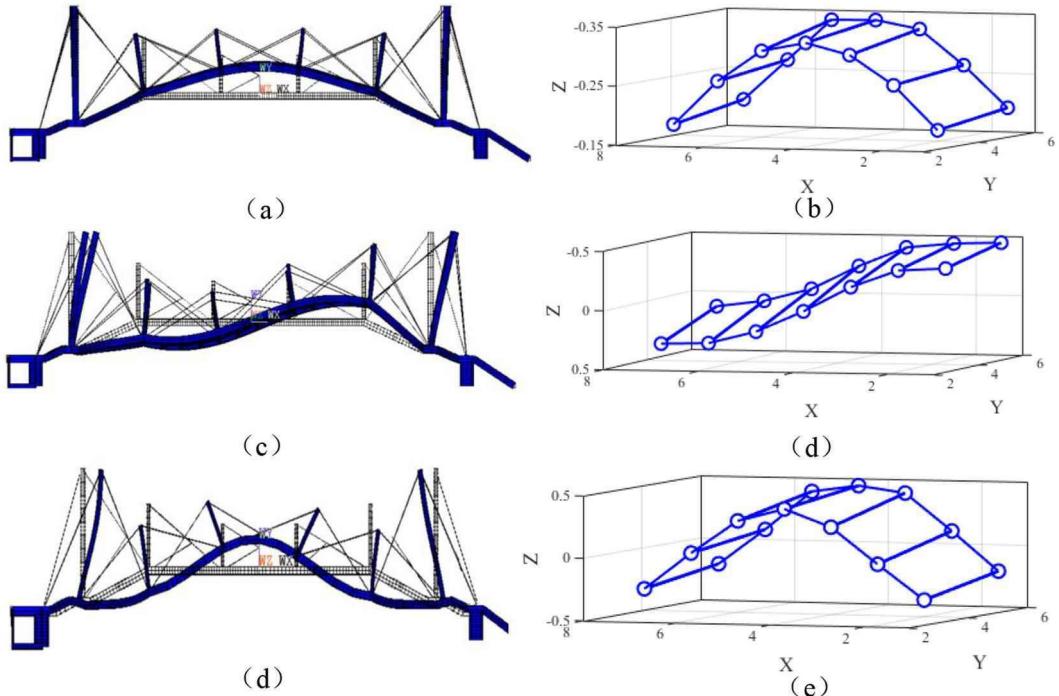


Fig. 30. Dynamic analysis results of steel footbridge: (a) Mode shape 1 of the proposed method; (b) Mode shape 1 of test; (c) Mode shape 2 of the proposed method; (d) Mode shape 2 of test; (e) Mode shape 3 of the proposed method; (f) Mode shape 3 of test.

intelligent drawing recognition technology, aiming to achieve automated extraction of bridge structure information and automated generation of finite element models, thereby providing strong technical support for bridge digital twin. The main conclusions are as follows:

- (1) Automated classification engineering drawings using novel deep learning-based search strategy has been proposed. This method integrates adaptive threshold algorithms with region-growing RANSAC algorithms, based on the projection density of complex bridge structures, to facilitate the automated extraction of critical component information such as bridge towers, cables, and main girders. This innovation significantly diminishes the reliance on manual intervention, thereby enhancing the automation level of external dimension extraction in bridge engineering.
- (2) A deep learning-based search strategy for automatic drawing has been developed, enabling the automated identification of drawings associated with components requiring measurement. This method streamlines the drawing screening process, markedly

reducing the manual effort involved. Furthermore, it achieves automatic classification and retrieval of drawings through the extraction of graphic features, text recognition, and the construction of hierarchical structures.

- (3) An automatic extraction framework for drawing using intelligent recognition technology has been developed. It utilized deep learning-based search strategy, optical character recognition (OCR), and large language model. This framework facilitates the automated extraction of internal information pertaining to key cross-sections through systematic fine-tuning of the large language model.
- (4) A FEM generation framework is established in which internal and external data are aligned through standardized component naming conventions. The process consists of three steps: component segmentation, finite element creation, and automated assignment of boundary conditions and loads based on spatial coordinates. This approach enables the automatic transformation of geometric and semantic information into a reliable finite

- element model, supporting downstream structural analysis and digital twin applications.
- (5) Finally, validation using both the scaled steel arch bridge model and a full-scale pedestrian cable-stayed bridge confirms that the proposed methodology can effectively automate the integration of point cloud data, engineering drawings, and physical semantics to rapidly generate accurate finite element models. This dual validation underscores the robustness and versatility of the framework across different bridge types. The proposed approach not only provides innovative solutions for advancing digital twin applications in bridge engineering but also establishes a reliable foundation for the informatization and automation of structural modeling processes, thereby offering substantial contributions to the field.

To further minimize manual intervention in the workflow, future improvements will focus on the integration of advanced automation techniques. Specifically, deep learning-based semantic segmentation can be employed to identify and remove non-structural environmental objects from raw point clouds, while adaptive denoising algorithms based on local geometric features can effectively eliminate measurement noise. These strategies are expected to significantly enhance the autonomy and robustness of the proposed modeling framework. Moreover, future work will include systematic benchmarking under varying hardware configurations and problem scales to further evaluate the practical feasibility of the proposed method.

CRediT authorship contribution statement

Yixuan Chen: Writing – original draft, Methodology, Investigation, Conceptualization. **Chenhai Gao:** Methodology, Investigation, Conceptualization. **Qijing Chen:** Visualization, Investigation. **Jian Zhang:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

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

Acknowledgment

The authors would like to gratefully acknowledge the support from the National Key R&D Program of China (No. 2022YFC3801700), the National Natural Science Foundation of China (No. 52378289), Research Fund for Advanced Ocean Institute of Southeast University (Key Program). And the first author gratefully acknowledges the financial support from the financial support from the China Scholarship Council (CSC, No. 202506090018), the Postgraduate Research&Practice Innovation Program of Jiangsu Province (No.SJCX24_0085), and the SEU Innovation Capability Enhancement Plan for Doctoral Students (No. CXJH_SEU 24096).

Data availability

Data will be made available on request.

References

- [1] Y. Chen, S. He, J. Zhang, Structural performance evaluation via digital-physical twin and multi-parameter identification, *Autom. Constr.* 170 (2025) 105907, <https://doi.org/10.1016/j.autcon.2025.105907>.
- [2] K. Lin, Y.-L. Xu, X. Lu, Z. Guan, J. Li, Digital twin-based collapse fragility assessment of a long-span cable-stayed bridge under strong earthquakes, *Autom. Constr.* 123 (2021) 103547, <https://doi.org/10.1016/j.autcon.2020.103547>.
- [3] S. Honghong, Y. Gang, L. Haijiang, Z. Tian, J. Annan, Digital twin enhanced BIM to shape full life cycle digital transformation for bridge engineering, *Autom. Constr.* 147 (2023) 104736, <https://doi.org/10.1016/j.autcon.2023.104736>.
- [4] L. Wang, H. Liu, F. Zhang, L. Guo, Z. Chen, Spatial structure digital twins: application in intelligent health monitoring of cable dome structures, *Autom. Constr.* 165 (2024) 105489, <https://doi.org/10.1016/j.autcon.2024.105489>.
- [5] Y. Xu, J. Zhang, UAV-based bridge geometric shape measurement using automatic bridge component detection and distributed multi-view reconstruction, *Autom. Constr.* 140 (2022) 104376, <https://doi.org/10.1016/j.autcon.2022.104376>.
- [6] Z. Shang, Z. Shen, Single-pass inline pipeline 3D reconstruction using depth camera array, *Autom. Constr.* 138 (2022) 104231, <https://doi.org/10.1016/j.autcon.2022.104231>.
- [7] D. Lamas, A. Justo, M. Soilán, B. Riveiro, Automated production of synthetic point clouds of truss bridges for semantic and instance segmentation using deep learning models, *Autom. Constr.* 158 (2024) 105176, <https://doi.org/10.1016/j.autcon.2024.105176>.
- [8] F. Wang, Z. Ma, Y. Cheng, W. Chen, J. Zhang, Automated measurement of cable shape in super-long span suspension bridges, *Autom. Constr.* 168 (2024) 105748, <https://doi.org/10.1016/j.autcon.2024.105748>.
- [9] F. Wang, B. Li, Y. Xu, J. Zhang, J. Zhang, Automated measurement method for cable shapes based on laser scanners and cameras, *J. Civ. Struct. Heal. Monit.* (2024) 1–13, <https://doi.org/10.1007/s13349-024-00723-1>.
- [10] S. Lin, L. Duan, J. Liu, J.L. Duan, J. Miao, C. Zhao, Automated geometric reconstruction and cable force inference for cable-net structures using 3D point clouds, *Autom. Constr.* 165 (2024) 105543, <https://doi.org/10.1016/j.autcon.2024.105543>.
- [11] Y. Zhu, I. Diaw, W. Wong, Cable-stayed bridge all-round morphology identification based on 3D point cloud model, *Struct. Infrastruct. Eng.* (2024) 1–16, <https://doi.org/10.1080/15732479.2024.105543>.
- [12] J.L. Ao, J.S. Fan, Y.F. Liu, B.L. Li, J.G. Nie, Region of interest (ROI) extraction and crack detection for UAV-based bridge inspection using point cloud segmentation and 3D-to-2D projection, *Autom. Constr.* 158 (2024) 105226, <https://doi.org/10.1016/j.autcon.2024.105226>.
- [13] C.J. Stull, C.J. Earls, A rapid assessment methodology for bridges damaged by truck strikes, *Steel Compos. Struct.* 9 (3) (2009) 223–237, <https://doi.org/10.12989/scs.2009.9.3.223>.
- [14] U. Duh, V. Shankar, G. Kosec, Discretization of non-uniform rational B-spline (NURBS) models for meshless isogeometric analysis, *J. Sci. Comput.* 100 (2) (2024) 51, <https://doi.org/10.1007/s10915-024-02644-6>.
- [15] N. Kasotakis, V. Sarhosis, B. Riveiro, B. Conde, A.M. D'Altri, J. Mills, G. Milani, S. de Miranda, G. Castellazzi, Three-dimensional discrete element modelling of rubble masonry structures from dense point clouds, *Autom. Constr.* 119 (2020) 103365, <https://doi.org/10.1016/j.autcon.2020.103365>.
- [16] L. Kudela, S. Kollmannsberger, U. Almac, E. Rank, Direct structural analysis of domains defined by point clouds, *Comput. Methods Appl. Mech. Eng.* 358 (2020) 112581, <https://doi.org/10.1016/j.cma.2019.112581>.
- [17] L. Cui, L. Zhou, Q. Xie, J. Liu, B. Han, T. Zhang, H. Luo, Direct generation of finite element mesh using 3D laser point cloud, *Structures* 47 (2023) 1579–1594, <https://doi.org/10.1016/j.istruc.2023.07.098>.
- [18] R. Lu, I. Brilakis, Digital twinning of existing reinforced concrete bridges from labelled point clusters, *Autom. Constr.* 105 (2019) 102837, <https://doi.org/10.1016/j.autcon.2019.102837>.
- [19] L. Yang, Y.-C. Lin, H. Cai, A. Habib, From scans to parametric BIM: an enhanced framework using synthetic data augmentation and parametric modeling for highway bridges, *J. Comput. Civ. Eng.* 38 (2) (2024) 5640, <https://doi.org/10.1061/JCCEE5.CPENG-5640>.
- [20] J. Shu, Z. Zeng, W. Li, S. Zhou, C. Zhang, C. Xu, H. Zhang, Automatic geometric digital twin of box girder bridge using a laser-scanned point cloud, *Autom. Constr.* 168 (2024) 105781, <https://doi.org/10.1016/j.autcon.2024.105781>.
- [21] A. Talebi, F. Potenza, V. Gattulli, Interoperability between BIM and FEM for vibration-based model updating of a pedestrian bridge, *Structures* 53 (2023) 1092–1107, <https://doi.org/10.1016/j.istruc.2023.08.125>.
- [22] H. Huang, B. Ruan, X. Wu, Y. Qin, Parameterized modeling and safety simulation of shield tunnel based on BIM-FEM automation framework, *Autom. Constr.* 162 (2024) 105362, <https://doi.org/10.1016/j.autcon.2024.105362>.
- [23] W. Gao, C. Zhang, X. Lu, W. Lu, Concrete spalling damage detection and seismic performance evaluation for RC shear walls via 3D reconstruction technique and numerical model updating, *Autom. Constr.* 156 (2023) 105146, <https://doi.org/10.1016/j.autcon.2023.105146>.
- [24] B. Conde-Camero, B. Riveiro, P. Arias, J.C. Caamaño, Exploitation of geometric data provided by laser scanning to create FEM structural models of bridges, *J. Bridg. Eng.* 20 (9) (2015) 04015003, [https://doi.org/10.1061/\(ASCE\)CF.1943-5509.0000807](https://doi.org/10.1061/(ASCE)CF.1943-5509.0000807).
- [25] Y. Yan, J.F. Hajjar, Geometric models from laser scanning data for superstructure components of steel girder bridges, *Autom. Constr.* 142 (2022) 104484, <https://doi.org/10.1016/j.autcon.2022.104484>.
- [26] Y. Yan, J.F. Hajjar, A parametric approach for creating finite element models from point clouds for steel superstructure components in steel girder bridges, *Struct. Infrastruct. Eng.* (2024), <https://doi.org/10.1080/15732479.2024.2408452>.
- [27] Y. Yan, J.F. Hajjar, Automated extraction of structural elements in steel girder bridges from laser point clouds, *Autom. Constr.* 125 (2021) 103582, <https://doi.org/10.1016/j.autcon.2021.103582>.
- [28] Y. Xu, J. Zhang, UAV-based bridge geometric shape measurement using automatic bridge component detection and distributed multi-view reconstruction, *Autom. Constr.* 140 (2022) 104376, <https://doi.org/10.1016/j.autcon.2022.104376>.

- [29] Y. Zeng, F. Huang, G. Xiong, Bridge point cloud semantic segmentation based on view consensus and cross-view self-prompt fusion, *Autom. Constr.* 171 (2025) 106003, <https://doi.org/10.1016/j.autcon.2025.106003>.
- [30] T. Xia, J. Yang, L. Chen, Automated semantic segmentation of bridge point cloud based on local descriptor and machine learning, *Autom. Constr.* 133 (2022) 103992, <https://doi.org/10.1016/j.autcon.2021.103992>.
- [31] T. Yang, Y. Zou, X. Yang, E. del Rey Castillo, Domain knowledge-enhanced region growing framework for semantic segmentation of bridge point clouds, *Autom. Constr.* 165 (2024) 105572, <https://doi.org/10.1016/j.autcon.2024.105572>.
- [32] F. Wang, Z. Ma, Y. Cheng, W. Chen, J. Zhang, Automated measurement of cable shape in super-long span suspension bridges, *Autom. Constr.* 168 (2024) 105748, <https://doi.org/10.1016/j.autcon.2024.105748>.