3D laser scanning for automated structural modeling and deviation monitoring of multi-section prefabricated cable domes

DOI:
10.1016/j.autcon.2024.105573

Document Version
Accepted author manuscript

Zhang, A.L., Ma, H., Zhao, X., Zhang, Y.X., Wang, J., Su, M.N., (2024), "3D laser scanning for automated structural modeling and deviation monitoring of multi-section prefabricated cable domes", Automation in Construction, 165: 105573.

3D Laser Scanning for Automated Structural Modeling and Deviation Monitoring of Multi-Section Prefabricated Cable Domes

Ailin Zhang [a, b], Hao Ma [a], Xi Zhao [a, d, *], Yanxia Zhang [a, d], Jie Wang[a], Meini Su [c]

[a] School of Civil and Transportation Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China
[b] Beijing University of Technology, Beijing, China
[c] School of Engineering, University of Manchester, Manchester M1 7JR, UK
[d] Beijing Energy Conservation & Sustainable Urban and Rural Development Provincial and Ministry Co-construction Collaboration Innovation Center, Beijing, China

ABSTRACT

This paper presents a multi-member automatic structural modeling (MASM) method for high-thrust deviation monitoring of prefabricated cable domes. Point cloud data generated by three-dimensional (3D) laser scanning were segmented into structural modules to effectively reduce the method's computational complexity. A multimember central shrinkage algorithm was developed for skeleton-point recognition. Subsequently, skeleton members were detected with sequentially identified joints, and the structural model of the cable dome was built. The MASM method was validated with respect to its 1) accuracy, ensuring a satisfactory signal-to-noise ratio, and 2) efficiency, ensuring competitive runtime. The use case of the cable-dome deviation monitoring was studied in detail. The proposed MASM method systematically evaluates prefabricated cable domes with multi-section members. This study enables high-fidelity analysis using a structural digital twin for predicting future structural performance.

Keywords: automatic structural modeling, cable dome structure, deviation monitoring, 3D laser scanning, structural digital twin

1. Introduction

Cable domes are long-span spatial structures used widely in stadiums [1], gymnasiums [1], and other large spaces. Cable-dome structures have long span, small weight, and aesthetic appearance, and feature continuously pre-tensioned cables and discrete compressed struts [4]. The broad applications of these structures garnered substantial attention with respect to the safety of cable domes, from the design stage [5] to the maintenance stage [11]. The safety of a typical structure relies significantly on its global stability and structural stiffness, both of which are sensitive to geometric deviations. Geometric deviations can reduce the cable prestress, consequently affecting the structure's structural stiffness. Furthermore, low structural stiffness compromises structural stability [13]. Thus, quantification and monitoring of geometric deviations are critical for ensuring the safety of cable domes and similar structures [14].

*** Correspondence author at:** School of Civil and Transportation Engineering, Beijing University of Civil Engineering and Architecture, Beijing,100044, China

Correspondence author. Email addresses: zhaoxi@bucea.edu.cn (X. Zhao).

Traditional techniques of deviation monitoring range from manual assessment and measurements of exploratory engineering positioning to total station techniques and measurements of member installation positioning [12]. These conventional inspection techniques face challenges such as low labor efficiency and scarcity of measurements owing to device-related constraints [15]. Therefore, substantial efforts have been made to develop better solutions for more efficient, thorough, and precise inspection. The development of digital-twin technology has provided new insight into this issue [17]. The digital-twin technology involves creating precise digital replicas of real-world entities and incorporating real-world data into models for various applications [21]. A digital twin of a structure contains accurate and thorough global geometric information about the corresponding structural entity and enables structure synchronization in the real world [22]. Subsequently, the geometry of a structural digital twin (SDT) can be compared with the building information modeling (BIM) model used in the initial design, to derive geometric deviations thoroughly, accurately, and locally [23].

The accuracy of an SDT depends on both the input source for high-threshold data measurements and the software used for efficient processing of volume point clouds [24]. Regarding the measurement technology, three-dimensional (3D) laser scanning and unmanned aerial vehicles (UAVs) can provide potential solutions for the global measurement of structural entities [25]. UAVs are generally employed in large-scale cartographic surveys and live-action modeling, in which the accuracy requirement is relatively relaxed [26], whereas 3D laser scanning has been developed based on the laser triangulation technique, which has a high level of accuracy [27]. 3D laser scanning provides an accuracy of 1.2 mm out of 100 m [29] and covers areas of up to 1400 m at 12.2 million points per second [30]. However, when using 3D laser scanning, challenges arise in terms of efficient point-cloud post-processing [31]. Currently, there are commercial software packages for data processing, such as Bentley Descartes [32], Realworks [33], and Cyclone [34]. Using these software methods, the scan results can be registered and denoised. However, these methods are only partially automated and require manual interventions for refining [35]. Moreover, general modeling processes may not be able to directly extract specific and essential structural data for the structural modeling of SDTs [36].

Therefore, substantial efforts have been made to develop structural modeling methods for preprocessed point clouds of structures [37]. Structural modeling methods require significant computational resources, owing to the large volume of point clouds [39]. However, the existing structural modeling methods are not applicable to complicated cable-dome structures. In general, structural modeling methods include member recognition and model reconstruction [40]. Although model reconstruction that re-aligns members is a well-developed process that requires minimal computational effort [37], member recognition of 3D point clouds of structures [41] remains a major challenge. Member recognition involves two steps: (a) skeleton recognition and (b) member segmentation [42].

- Skeleton recognition targets the simplification of structural point clouds that maintain the necessary geometric features of the members [43]. Rotational symmetry axis (ROSA) [44], Laplace contraction [45], and morphological image-based methods [46] have been used to identify the skeleton points of buildings and infrastructure. The ROSA method determines the normal vectors of point-cloud-formed surfaces, where the intersections of vector lines are considered as the skeleton of structural members [44]. This method is suitable for pipeline structures. The Laplace contraction approach utilizes Laplacian factors to contract the points to form the skeleton lines [47]. This is more feasible for tubular structures with the same cross-sections of members [39]. By contrast, morphological image-based methods utilize gray processing to determine crack locations that are more suitable for plane-crack recognition [48]. However, a typical cable-dome structure comprises various cross-sections of spatially connected members. The ROSA and Laplacian contraction methods fail to search for skeletons at the joints of multidirectional members. However, morphological image-based methods cannot support the spatial recognition of cable-dome members.
- Member segmentation is then conducted on the skeleton points of the structures, where member skeletons are recognized by the structural entity [49]. General methods for member extraction include edge contraction [50] and region growth [51]. Edge contraction creates a matrix representing the connectivity of the members and searches for the skeleton points at nodes to identify the member skeletons [42]. Region growing calculates the angles formed between neighboring points and searches for node positions determined by thresholds to extract the member skeletons [39]. However, these methods are not suitable for spatial structures with multi-direction members with various cross-sections, in which the misalignment of members is frequent owing to fixed thresholds.

In summary, better member-recognition methods for spatial cable-dome structures must be developed. The algorithms for both skeleton recognition and member extraction should accommodate multidirectional and multi-section members in spatial cable-domes. Furthermore, efficient structural modeling methods should be developed to incorporate the member-recognition methods within specialized data structures. It would be advantageous to build SDTs and monitor geometric deviations to address safety concerns more accurately and efficiently.

To address the above, the present study aimed to develop a more efficient and accurate multimember automatic structural modeling (MASM) method for reconstructing cable-dome structures. The key novelties of the proposed method include four processing stages: (a) point cloud data (PCD) module segmentation; (b) multimember central shrinkage for recognizing the skeleton points; (c) smart multimember recognition; and (d) structural model

reconstruction. The stages of the proposed approach are shown in Fig. 1. First, during the data-preparation step, the proposed module segmentation minimizes the computational effort for further processing. Second, the novel multimember central shrinkage method for the skeleton-point recognition algorithm adopts multi-layering and clustering to modulate point clouds according to the spatial point-cloud features. Central shrinkage is performed by finding the clusters' centroids, which perfectly avoids over shrinkage owing to the changes in the member sections. Thus, the skeleton points for the structural entity are determined. Third, a novel smart multimember recognition algorithm, addressing the multi-section members of a cable dome, is developed. Smart member extraction of member recognition compares the skeletons to a standard model from the design where the skeleton members are recognized. With the proposed method, the accuracy of the member extraction is significantly higher, whereas the computational complexity is very small. Fourth, a cable dome is systematically reconstructed, during which the orientation and position of the skeleton members are calculated. The nodes of the members are then determined at the intersections of the projected lines from the skeleton members. Thus, the module of the cable-dome structure is reconstructed. The final step of the reconstruction is to assemble the segmented modules into a cable-dome entity. It should be noted that a new data structure is formed that includes member types, sectional dimensions, orientations, and positions of members, as well as the node coordinates. Consequently, the geometric deviations of a cable-dome entity can be efficiently computed by comparison with the initial design model.

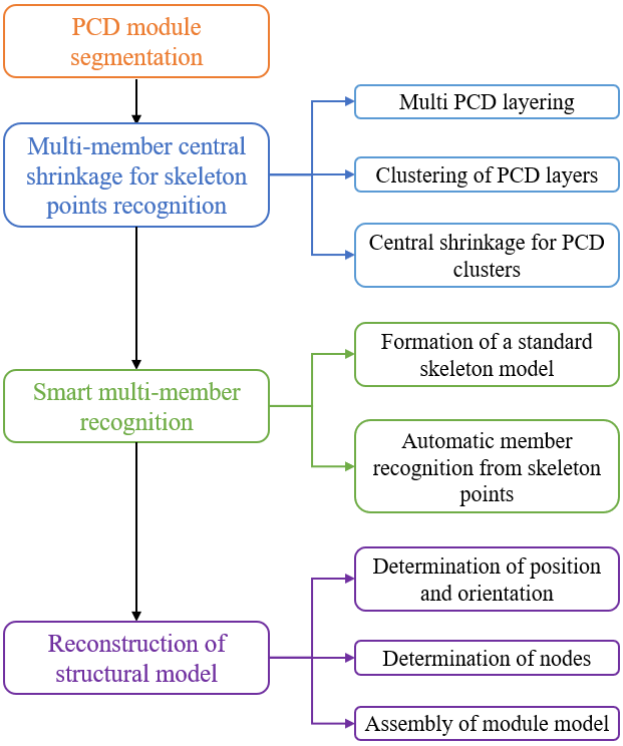


Fig. 1. Flowchart of the proposed MASM method.

This paper first presents an experimental model of a cable dome and the details of the PCD preparation process (Section 2). The theory and implementation of the proposed structural modeling method are described in Sections 3–6. Section 7 discusses the accuracy and efficiency of the proposed structural model. The corresponding geometric deviation monitoring is presented as well. Finally, the conclusions of the study are listed.

2. Data preparation

This section introduces the experimental cable dome, laser scanning devices and procedures, and the preprocessing of the measured data. The structural formation and member geometry of the considered complex spatial structure are described in Section 2.1. The terrestrial laser scanning device and measurement scheme are detailed in Section 2.2. Section 2.3 mainly presents the preprocessing of the structural point-cloud measurements obtained using the 3D laser scanner.

2.1. Structural information

The experimental cable-dome model was a scaled-down ridge-tube cable dome with diagonal struts [4], as shown in Fig. 2. The scale-down factor was 1:10. The diameter and height of the cable dome were 10 m and 1 m, respectively. The cable dome was centrally symmetric and had 24 identical circumferential units. The structure consisted of three circles of ring-hoop cables and four loops of diagonal cables. It had center, ridge, and diagonal struts, with cross-sectional dimensions that varied with respect to the installation location. The dimensions of the ridge struts were Φ30×3, Φ30×3, Φ40×3, Φ40×3, corresponding to the first, second, third, and fourth loops. The dimensions of the diagonal struts were Φ25×3, Φ30×3, and Φ40×3, with respect to the first, second, and third circles.

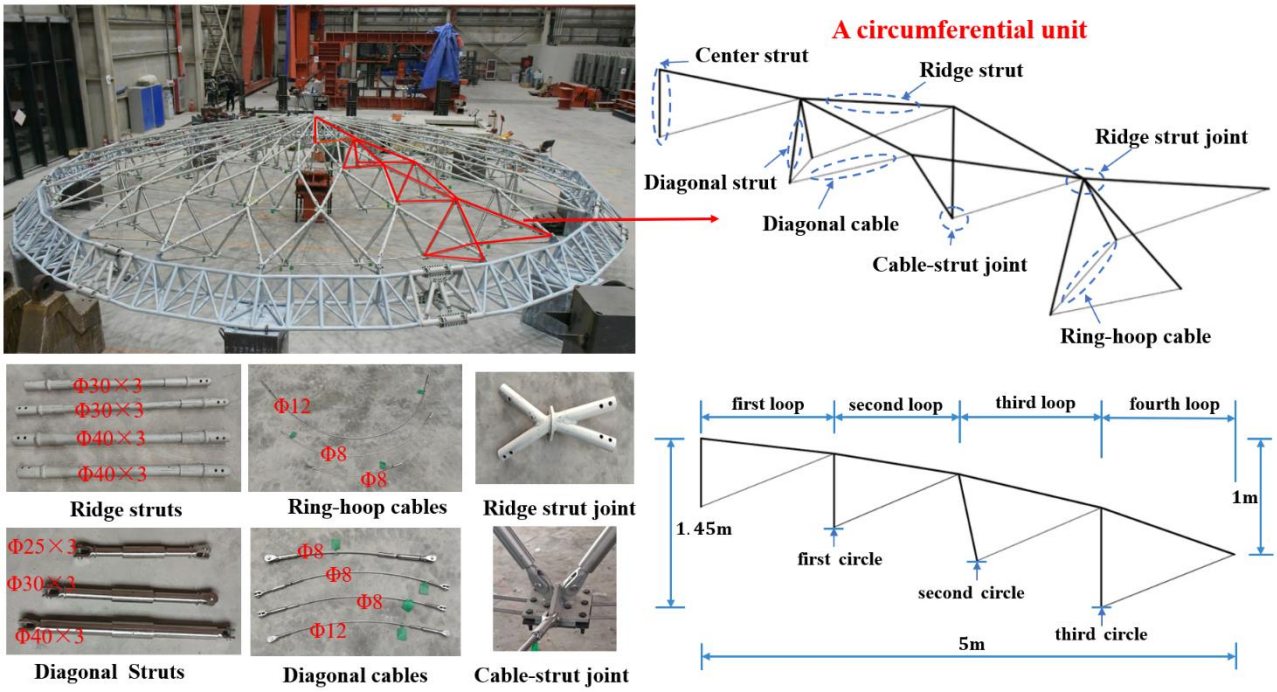


Fig. 2. Structural form and member information.

2.2. 3D laser scanning

The structural PCD was obtained using a RIEGL laser-scanning system. This system encompassed a terrestrial laser scanner RIGEL-VZ1000 (Fig. 3) with PCD preprocessing software RiSCAN Pro. Scanning parameter *Panorama40* mode was used for scanning. The measurement distance varied through the 2.5–1400 m range, covering 360° horizontal measurements. The angular resolution was above 1.8 arcsec. Hence, the measurement accuracy reached 5 mm at 100 m [52].

A meticulous scanning scheme was prepared beforehand to ensure the precision and completeness of the data with minimal redundancy associated with the PCD processing (Fig. 3). Twelve scanning sites with positioning markers were predetermined and positioned clockwise. Two sites inside the structure were designated to capture the interior details of the members, that is, sites 4 and 9, and the remaining sites outside the structure were responsible for recording the overall exterior details [53].

The scanning time for each site was 180 s, the PCD volume was approximately 268 MB. The total scanning time was approximately 2160 s; the total data volume was approximately 3.13 GB. The exported data included the 3D point clouds and RGB values, which were saved in the "rxp" format [52].

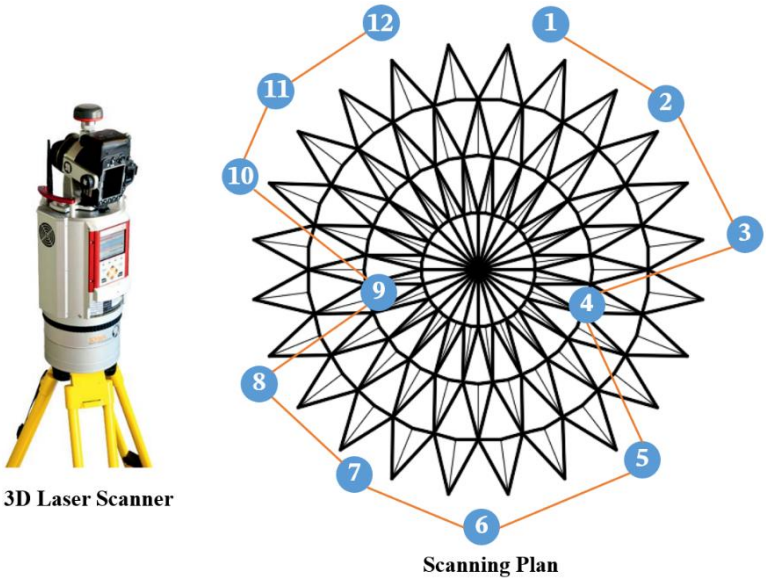


Fig. 3. RIGEL-VZ1000 scanner and corresponding scan locations.

2.3. PCD preprocessing

The exported PCD was processed using the RiSCAN Pro software designed for the RIGEL scanner [54]. The preprocessing included the registration of measurements at 12 sites and elimination of unwanted points and/or noise.

Registration

Registration is a two-step process that is automatically realized in RiSCAN. The initial global registration was conducted on 12-site raw PCD sets that were roughly rotated and displaced by the location of the positioning markers. The initial registration took approximately 320 s, with an accuracy exceeding 3 mm, as shown in Table 1. Finer location registration was then conducted on adjacently measured PCD sets in the *automatic iterative closest point registration* mode. This finer registration took approximately 8 s, and the registration accuracy was within 0.5 mm (Table 1). The final registration accuracy satisfied the further processing requirements. The PCD volume after the registration was 560.42 MB.

TABLE 1.

PCD registration information.

Stage	Registration accuracy	Registration time
Initial global registration	3.874 s	319.2 s
Fine local registration	0.2104 s	8.4 s

Note: The RiSCAN interface provides both registration accuracy and time.

Denoising

Point cloud denoising was performed using the RiSCAN Pro software. Factors such as equipment precision, environmental conditions, and alignment can add significant noise to PCD, directly affecting the modeling accuracy. Statistical filtering was employed to remove noisy data, and the filtering intensity was adjusted by varying both the count of neighboring points and the standard deviation multiplier. The details are listed in Table 2. This procedure significantly reduced the amount of noise and missing data; residual noise was further addressed in the member-recognition step (Fig. 4).

TABLE 2.

PCD denoising information.

Parameter setting		Data volume		Time
Number of neighboring points	6	Pre-denoising	15689612 points (560.42 MB)	28.43 s
Standard deviation multiplier	1.00	Post-denoising	12661192 points (443.03 MB)	

The preprocessed PCD of the entire experimental cable contained more than 12 million points; the corresponding data size was 443.03 MB. The RGB values of the PCD were not required. The PCD coordinates were stored in the "pts" format.

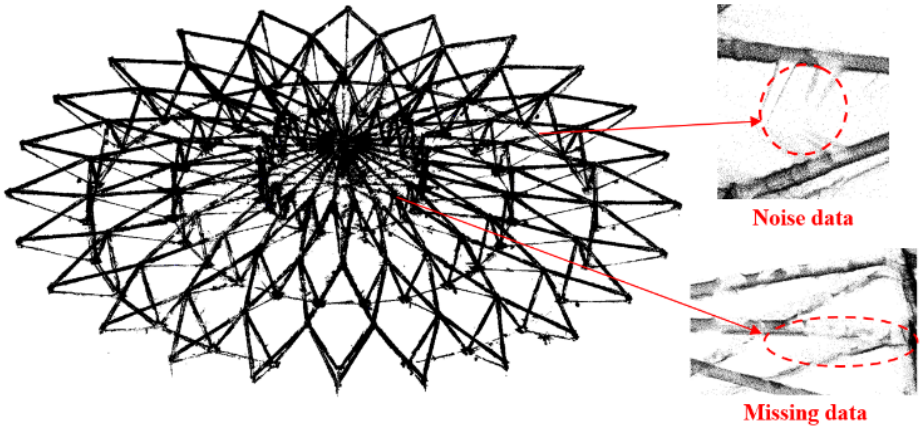


Fig. 4. PCD after denoising.

3. PCD module segmentation

Preprocessed PCD has a large volume, making is inconvenient for structural modeling. A typical cable dome is symmetrical and can be divided into identical modules. The number of modules would depend on the computational efficiency requirements, where one module should contain at least one unit of the structure. The commercial software *CloudCompare* satisfied these segmentation requirements.

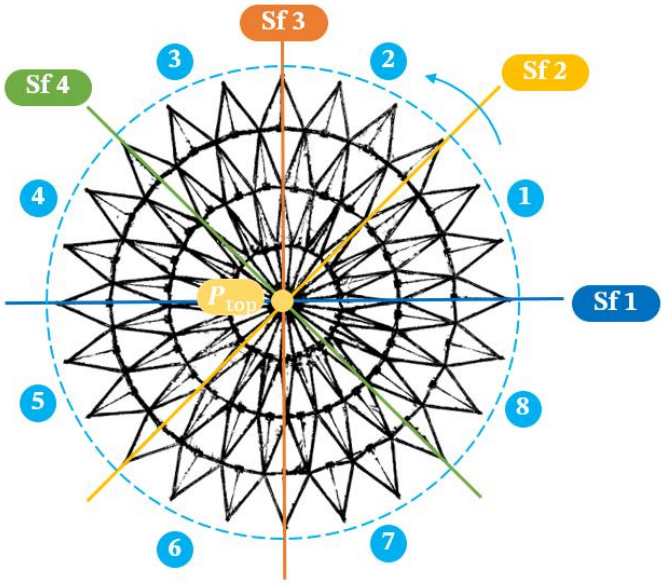


Fig. 5. Module segmentation.

The experimental cable dome contained 24 structural units uniformly segmented into eight modules, as shown in Fig. 5. The highest point, P_{top}, which is generally at the top of the center strut, was the origin (0,0,0). The axis aligned with P_{top} and perpendicular to the X-Y plane was the Z-axis. A segmentation plane was generated perpendicular to the X-Y plane at 45° increments towards the X-Z plane. Thus, four segmentation planes were generated, i.e., $Sf_i, i \in [1,4]$, and eight modules were obtained. The overall module-segmentation process took approximately 160 s. The module contained approximately 1.4 million points with a maximum volume reaching 59 MB. The data corresponding to the segmented modules were later stored in the form of a KD-tree, which is a k-

dimensional binary tree for efficient PCD storage and searching. The segmentation ensured no loss of information, guaranteeing the entity of structural data. In addition, the PCDs of the ridge struts were confined to the individual modules, eliminating redundancy. However, the data format remained "pts" and only the 3D coordinates of the points were recorded.

4. Multi-member central shrinkage for recognition of skeleton points

Multi-member central shrinkage for recognition of skeleton points is the first step in member recognition; the objective is to simplify the bulky PCD to the skeleton points of members with various cross-sections. The requirement on the multi-member central shrinkage for recognition of skeleton points was to precisely maintain spatial geometric features. The recognition started with multi-PCD layering, in which PCD layers with various member segments were obtained, as described in Section 4.1. Next, PCD clustering was performed for segmentation of different members in different PCD layers, as described in Section 4.2. The skeleton points of the structural modules were then obtained using the central shrinkage algorithm, as described in Section 4.3.

4.1 Multi-PCD layering

Layering seeks to divide the member PCD into multiple segments L_i, as shown in Fig. 6. The division is based on the predetermined radius Δr_s from center P_{top} to the end of the module. Considering the spatial geometric features, the PCD layering algorithm comprised two preservation steps. In the first layering step, the ridge struts were processed, and the PCD was searched using polar coordinates. In the second step of layering, the remaining cables and diagonal struts were considered, and the PCD segments were searched along the Z-axis in the Cartesian coordinate system.

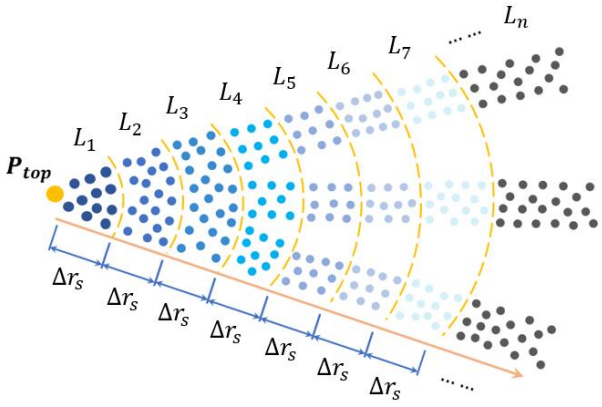


Fig. 6. Multi-PCD layering.

Ridge-strut Layering

The ridge struts were spatially and circumferentially distributed outward by diffusion from the top of the center strut. The total length R^{RS} (Fig. 7(a)) between peak P_{top} and farthest point P_{far} was first calculated according to Eq.

(1). The number of layers N^{RS} was predetermined according to the volume of the module PCD, using the procedure described in Section 4.3. A searching radius increment, Δr_s^{RS}, was calculated using Eq. (2), assuming uniformity.

$$R^{RS} = \sqrt{(x_b - x_t)^2 + (y_b - y_t)^2 + (z_b - z_t)^2} = \sqrt{{x_b}^2 + {y_b}^2 + {z_b}^2} \quad (1)$$

$$\Delta r_s^{RS} = R^{RS}/N^{RS} \quad (2)$$

In the above, the superscript RS refers to the ridge strut; (x_t, y_t, z_t) and (x_b, y_b, z_b) are the coordinates of P_{top} and P_{far}, respectively. The coordinates of P_{top} were (0,0,0).

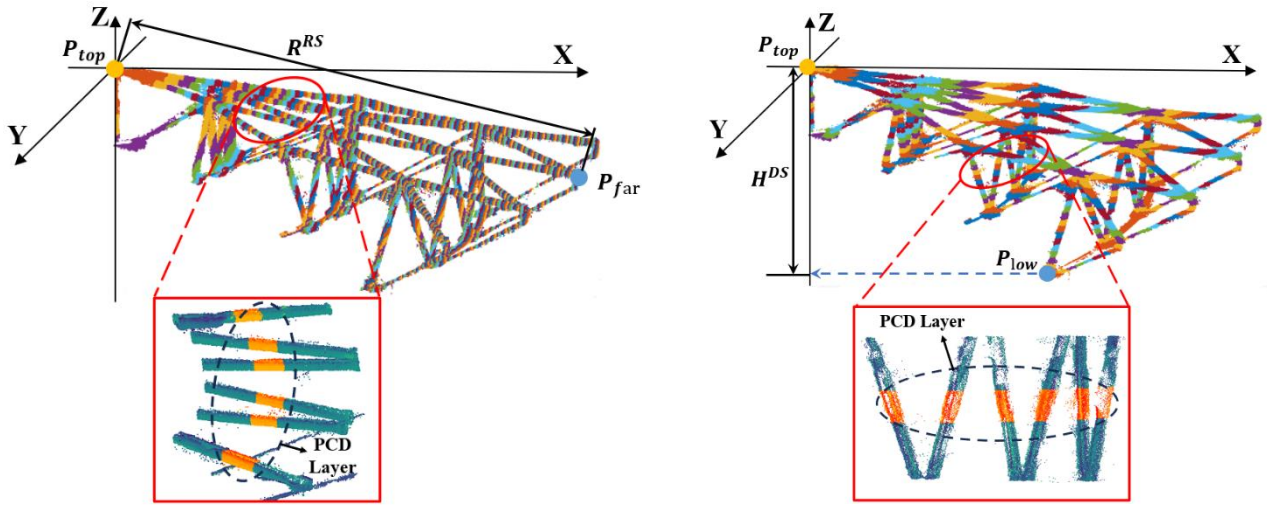
The ridge-strut PCD was layered under a polar coordinate system, where the points of the layers were searched for under radius r_P^{RS}. The value of r_P^{RS} was computed using Eq. (3), which captured the distance between layered points P and P_{top}. It was assumed that points within a specific range of radii, as determined by Eq. (4), belonged to the same layer $\boldsymbol{L_n^{RS}}$, with n representing the n^{th} layer out of N^{RS} layers. The lower and upper bounds, R_{n-1}^{RS} and R_n^{RS}, of L_n^{RS} were calculated according to Eq. (5). The layer-searching procedure continued to the ridge struts until reaching the farthest point P_{far}.

$$r_P^{RS} = \sqrt{(x_P)^2 + (y_P)^2 + (z_P)^2} \quad (3)$$

$$\boldsymbol{L_n^{RS}} = \{P | R_{n-1}^{RS} < r_P^{RS} < R_n^{RS}\}, n \in [1, N^{RS}] \quad (4)$$

$$\begin{cases} R_{n-1}^{RS} = (n-1) \cdot \Delta r_s^{RS} \\ R_n^{RS} = n \cdot \Delta r_s^{RS} \end{cases}, n \in [1, N^{RS}] \quad (5)$$

In the above, r_P^{RS} represents the distance of point P from P_{top}, whereas (x_P, y_P, z_P) represent the coordinates of point P.



(a) Layering for ridge struts (b) Layering for diagonal struts

Fig. 7. Multi-PCD layering.

Diagonal-strut Layering

Diagonal struts were arranged vertically in the structure. The diagonal-strut PCD was obtained by considering the Cartesian coordinate system, where the layer points were searched under heights from the Z-axis projected points to the center P_{top} of the diagonal struts (Fig. 7(b)). The total height, H^{DS}, corresponding to the height between peak P_{top} and lowest projected point P_{low}, was calculated using Eq. (6). The height-search increment Δh_s^{DS} was calculated using Eq. (7).

$$H^{DS} = |z_l - z_t| = |z_l| \tag{6}$$

$$\Delta h_s^{DS} = H^{DS}/N^{DS} \tag{7}$$

In the above, z_l is the z coordinate of P_{low}, and the superscript DS denotes the diagonal strut.

Similar to the ridge-strut layering, the diagonal-strut PCD was layered according to the height h_P^{DS}. The height h_P^{DS}, as the projected distance between points P and P_{top}, was calculated according to Eq. (8). The points of the diagonal struts within a specific range of heights belong to the same layer $\boldsymbol{L_n^{DS}}$, where n represents the n^{th} layer out of N^{DS} layers for the ridge struts (Eq. (9)). The lower and upper bounds, H_{n-1}^{DS} and H_n^{DS}, of L_n^{DS} were calculated according to Eq. (10). The search for layers continued to ridge struts until reaching the lowest point P_{low}.

$$\boldsymbol{L_n^{DS}} = \{P | H_{n-1}^{DS} < h_P^{DS} < H_n^{DS}\}, n \in [1, N^{DS}] \tag{9}$$

$$h_P^{DS} = |z_P - z_t| = |z_P| \tag{8}$$

$$\begin{cases} H_{n-1}^{DS} = (n-1) \cdot \Delta h_s^{DS} \\ H_n^{DS} = n \cdot \Delta h_s^{DS} \end{cases}, n \in [1, N^{DS}] \tag{10}$$

In the above, h_P^{DS} represents the projected distance to P_{top} of point P, and z_P represents the z-coordinate of point P.

Based on the above theory, a multi-PCD layering algorithm was implemented in MATLAB, as shown in Algorithm 1.

Algorithm 1: PCD layering

Input: all ***points***, the total number of layers N^{RS} and N^{DS}

Output: layers $\boldsymbol{L^{RS}}$ and $\boldsymbol{L^{DS}}$

points_RS= ***points_DS*** = ***points***

P_{top}= the peak point in ***points***

P_{far}= the farthest point from P_{top} in ***points***

P_{low}= the lowest point in ***points***

R^{RS}= Distance (P_{top}, P_{far})

$$\Delta r_s^{RS} = R^{RS}/N^{RS}$$

H^{DS}= Projected distance in z-direction (P_{top}, P_{low})

$$\Delta h_s^{DS} = H^{DS}/N^{DS}$$

$\boldsymbol{L_n^{RS}} = \emptyset, n \in [1, N^{RS}]$

$\boldsymbol{L_n^{DS}} = \emptyset, n \in [1, H^{DS}]$

While (size(***points_RS***,1)~=0)

P = the first point in ***points_RS***

r_P^{RS}= Distance (P, P_{top})
Calculate the integer multiple m of r_P^{RS} to Δr_s^{RS}
Add P into $\boldsymbol{L_m^{RS}}$
Delete P from ***points_RS***
end

While (size(***points_DS***,1)~=0)
P = the first point in ***points_DS***
h_P^{DS}= Projected distance in z-direction (P, P_{top})
Calculate the integer multiple m of h_P^{DS} to Δh_s^{DS}
Add P into $\boldsymbol{L_m^{DS}}$
Delete P from ***points_DS***
end

Determination of Parameters

The multilayering algorithm has two parameters, N^{DS} and N^{RS}, which must be determined in addition to the PCD module. The value of N^{DS} corresponds to the number of diagonal-strut layers, whereas N^{RS} corresponds to the number of ridge-strut layers, as mentioned previously. The ratio of N^{RS} to N^{DS} should align with the dimensions of the structure. The height-to-span ratio of an individual module was approximately 1:5, indicating that the corresponding ratio of N^{DS} to N^{RS} was 1:5, as shown in Eq. (11).

$$N^{RS} = 5N^{DS} \tag{11}$$

The next step was to determine the value of N^{DS} that affected the search for the skeleton points. A skeleton point generally requires at least 4000 points/layer density of the PCD for the proper reconstruction of module members. This indicated that N^{DS} should be at least 50, with higher values corresponding to more advantageous outcomes. However, increasing N^{DS} increases the computational complexity of the layering process. The value of N^{DS} therefore was set to the minimum value of 50. Thus, the value of N^{RS} was 250.

4.2. Clustering of PCD layers

The layering points were further clustered, and noise points were eliminated using density-based spatial clustering of applications with noise (DBSCAN) algorithm [55]. The DBSCAN algorithm defines points in a defined circular window as cluster $\boldsymbol{N^K}$, where the density of the points in a cluster is assumed to be uniform (Fig. 8). Subsequently, a uniform cluster can be later shrunk into skeleton points. The densities of the cluster points were used to filter the PCD noise. The clusters formed by the structural members exhibited a high density of points. In contrast, the clusters polluted with noise exhibited a low density of points.

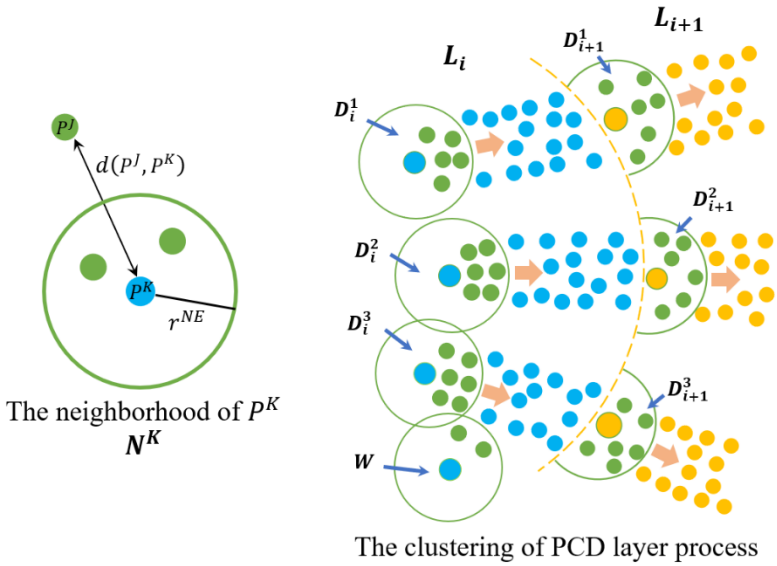


Fig. 8. Clustering of PCD layers.

The clustering process was conducted from the nearest layer to the farthest layer, considering the distances to P_{top} as shown in Algorithm 2. In layer L_i, the clustering first sorted the points by distance to P_{top}, from smallest to largest. Thus, point P_i^K, nearest to P_{top} , was selected as the starting point, with i representing the layer order and K representing the cluster order in L_i. The distance $d(P_i^K, P_i^J)$, between points P_i^K and any other point P_i^J, was then calculated according to Eq. (12). The search radius, r^{NE}, was 1.5 × the maximum diameter of struts that all cross-sectional PCD of members covered. Cluster $\boldsymbol{N_i^K}$ was formed by points at distances $d(P_i^K, P_i^J)$ within the search radius r^{NE}, as captured by Eq. (13), and thereafter, the search continued to the remaining PCD in L^i where new point P_i^{K+1}, nearest to P_{top}, was selected, and new cluster $\boldsymbol{N_i^{K+1}}$ was formed. The procedure was stopped when all the layers were searched.

$$d(P_i^K, P_i^J) = \sqrt{(x_i^K - x_i^J)^2 + (y_i^K - y_i^J)^2 + (z_i^K - z_i^J)^2}, J, K \in [1, M_K] \tag{12}$$

$$\boldsymbol{N_i^K} = \{P_i^J | d(P_i^K, P_i^J) \leq r^{NE}\} \tag{13}$$

In the above, (x_i^K, y_i^K, z_i^K) and (x_i^J, y_i^J, z_i^J) represent the coordinates of points P^K and P^J, whereas M_K denotes the overall number of PCDs in layer L_i.

The overall number M_D of clusters was then considered, where the threshold value M_{thr} was $M_K/60$, according to Eq. (14). For $M_D < M_{thr}$, the points in $\boldsymbol{N_i^K}$ were considered noise points $\boldsymbol{W}$ and were directly deleted. Otherwise, the points were considered valid member points and a PCD cluster $\boldsymbol{D_i^K}$ was formed for further analysis. The search was repeated until the nearest new P^K in the layer was found, excluding points in $\boldsymbol{N_i^K}$ from the previous search.

$$\begin{cases} \boldsymbol{W} = \boldsymbol{N}_i^K & M_D < M_{th} \\ \boldsymbol{D}_i^K = \boldsymbol{N}_i^K & M_D > M_{th} \end{cases} \tag{14}$$

Based on the above theory, the clustering of PCD layers was implemented using MATLAB, as shown in Algorithm 2.

Algorithm 2: Clustering of PCD layers

Input: all points in $\boldsymbol{L_i}$, the threshold number M_{thr}, the search radius r^{NE}
Output: the cluster $\boldsymbol{D^{clus}}$

While (size ($\boldsymbol{L_i}$,1)~=0)
  P^K = the first point in $\boldsymbol{L_i}$
  $\boldsymbol{N_k}(P^K) = \emptyset$
  Find out Distance from point P^K to each of the surrounding points P^J, $J \in [1, \text{size}\ (\boldsymbol{L_i}, 1)]$
  for i = 1: size($\boldsymbol{L_i}$, 1)
    if Distance (P^K, P^J)< r^{NE}
      Add P^J into $\boldsymbol{N_k}(P^K)$
    end
  end
  if the number in $\boldsymbol{N_k}(P^K) > M_{thr}$
    $\boldsymbol{N_k}(P^K)$ is considered as $\boldsymbol{D^{clus}}$
  else
    $\boldsymbol{N_k}(P^K)$ is considered as noise point
  end
  Delete P^K from $\boldsymbol{L^{RS}}$
end

Using PCD layering and clustering, the PCD clusters of members with different orientations were extracted uniformly and efficiently.

4.3. Central shrinkage for PCD clusters

The skeleton points were determined by centrally shrinking the PCD member clusters $\boldsymbol{D_i^k}$, as described in detail in this section. As shown in Fig. 9, the differently colored points represent the corresponding PCD member clusters $\boldsymbol{D_i^k}$. Red points represent the target skeleton points.

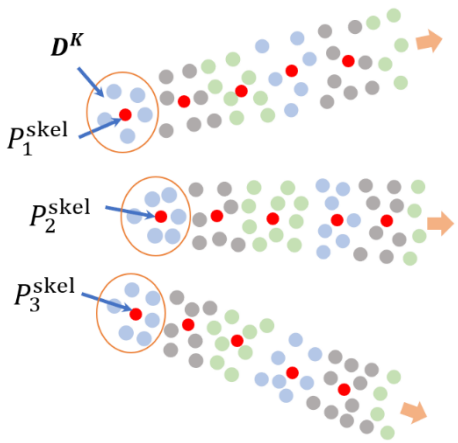


Fig. 9. Central shrinkage of PCD clusters $\boldsymbol{D_i^K}$.

The point densities of the member clusters were similar. Thus, centroids were obtained by averaging the coordinates of the points in a cluster, as shown in Eq. (15). The shrinking window was applied sequentially through the PCD member cluster $\boldsymbol{D_i^k}$ to recognize the skeleton point P^{skel} for each PCD cluster. The obtained skeleton points were aligned, yielding the skeleton-point model $\boldsymbol{S_{sm}}$. The ridge- and diagonal-strut skeleton points are shown in orange and blue, respectively. Sparse points were observed on the cables, as discussed in Section 5.

$$\begin{cases} x^{sm} = \dfrac{1}{M^D}\sum_1^{M^D} x_t \\ y^{sm} = \dfrac{1}{M^D}\sum_1^{M^D} y_t, t \in [1, M^D] \\ z^{sm} = \dfrac{1}{M^D}\sum_1^{M^D} z_t \end{cases} \tag{15}$$

In the above, (x^{sm}, y^{sm}, z^{sm}) are the coordinates of point P^{sm}, (x_t, y_t, z_t) are the coordinates of a point in a PCD member cluster $\boldsymbol{D_i^k}$, and M^D is the number of points in the PCD cluster.

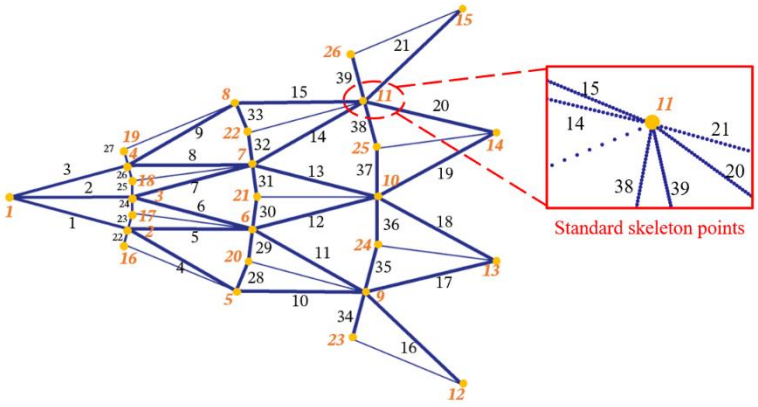
Compared with the original PCD, the data volume after the skeleton-point recognition method was effectively reduced to 0.37% of the original volume. Most importantly, the key geometric features of all the members were preserved. Further development was conducted to automatically recognize members from skeleton models.

5. Smart multi-member recognition

The skeleton model obtained in the previous section maintained geometric integrity with spatial features. Therefore, automatic multi-member recognition was conducted to efficiently obtain the geometric information of structural members for structural model construction. Recognition was performed in two steps: the formation of a standard skeleton model and automatic multimember recognition, as described in Sections 5.1 and 5.2, respectively.

5.1. Formation of a standard skeleton model

A standard skeleton model, developed from the design structure using MATLAB, was required as a reference for multi-member recognition. The design structure was first segmented into the corresponding modules. The standard node information and node adjacency matrix (INAM) of the design module were identified, as shown in Figs. 10(a) and 10(b), respectively.



(a) Standard node information.

	1	2	3	⋯	9	10	11	⋯	24	25	26
1	0	1	2	⋯	0	0	0	⋯	0	0	0
2	1	0	0	⋯	0	0	0	⋯	0	0	0
3	2	0	0	⋯	0	0	0	⋯	0	0	0
⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋱	⋮	⋮	⋮
9	0	0	0	⋯	⋮	⋮	⋮	⋯	35	0	0
10	0	0	0	⋯	⋮	⋮	⋮	⋯	0	37	0
11	0	0	0	⋯	⋮	⋮	⋮	⋯	0	28	39
⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋱	⋮	⋮	⋮
24	0	0	0	⋯	35	0	0	⋯	0	0	0
25	0	0	0	⋯	0	37	38	⋯	0	0	0
26	0	0	0	⋯	0	0	39	⋯	0	0	0

Node IDs

Member IDs

(b) The improved node adjacency matrix.

Fig. 10. Standard skeleton model.

The standard node information includes both the node ID number and corresponding coordinates of the design module. By contrast, the node adjacency matrix captures the connectivity between the nodes of the design modules. The pivot rows and columns represent the node ID. The other entries in the matrix represent the member IDs that connect the corresponding nodes.

The INAM is a symmetric and hollow matrix; that is, all diagonal elements are zero, indicating that nodes are not self-connected. A standard skeleton model, $\boldsymbol{S_{st}}$ was formed based on the INAM design. The members and connections were assumed to be perfect (without defects). Interpolation was conducted among the nodes, to form the skeleton points of the members. The density of $\boldsymbol{S_{st}}$ was 200 points/member, as estimated from the number of member skeleton points in $\boldsymbol{S_{sm}}$. The $\boldsymbol{S_{st}}$ data contained both the member IDs and spatial coordinates, which were used as references for member recognition in the following section.

5.2. Automatic multi-member recognition from skeleton points

An automatic multi-member recognition algorithm was developed using an iterative closest point (ICP) method, in which the surface formed by the skeleton model $\boldsymbol{S_{sm}}$ was compared with that of the standard model $\boldsymbol{S_{st}}$. The errors between the geometric features of the two surfaces were minimized such that a transform matrix, including rotations and displacements, for $\boldsymbol{S_{st}}$ was obtained. Subsequently, $\boldsymbol{S_{st}}$ was registered. The recognition of the members of $\boldsymbol{S_{sm}}$ was performed automatically based on the spatial position of the members of $\boldsymbol{S_{st}}$.

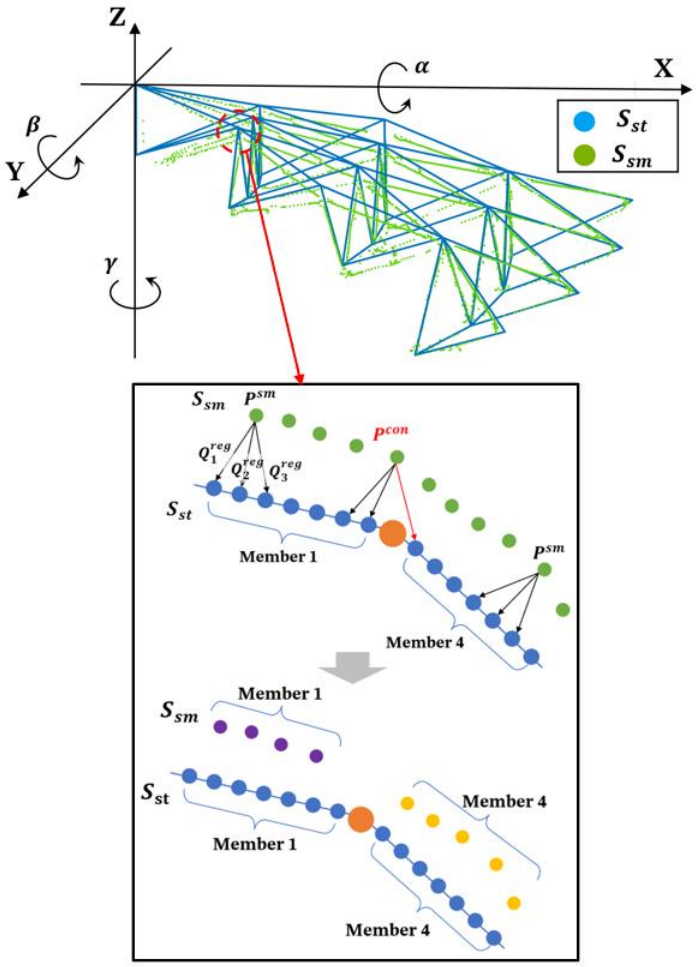


Fig. 11. Registration of skeleton models $\boldsymbol{S_{st}}$ and $\boldsymbol{S_{sm}}$.

Registration

The sets $\boldsymbol{S_{st}}$ and $\boldsymbol{S_{sm}}$ were first sorted using KD-Tree algorithms for efficient computation. A K-nearest neighbor search was implemented, where comparison pairs of points between $\boldsymbol{S_{sm}}$ and $\boldsymbol{S_{st}}$ were formed, (P^{sm}, Q^{st}), as shown in Fig. 11. The K-nearest neighbor method used the Euclidean distance as the target function in Eq. (16), to optimize the data registration of $\boldsymbol{S_{st}}$, that is, the rotation matrix $\boldsymbol{R^{REG}}$ and translation vector $\vec{\boldsymbol{t}}^{\boldsymbol{REG}}$. The rotation matrix $\boldsymbol{R^{REG}}$ was a 3 × 3 matrix, with three angles α, β, γ rotating around the X, Y, Z axes, as captured by Eq. (17). The translation vector $\vec{\boldsymbol{t}}^{\boldsymbol{REG}}$ was a 3 × 1 vector with three displacements, t_x, t_y, t_z along the X, Y, Z axes from the origin, as captured by Eq. (18). The optimization was repeated approximately 50 times and the registration error was constrained to within 0.2 mm. The computation time was constrained to within 0.2 s. Finally, points Q^{st} in $\boldsymbol{S_{st}}$ were adjusted to the coordinate system of $\boldsymbol{S_{sm}}$ by rotation and translation, according to Eq. (19).

$$E\left(\boldsymbol{R^{REG}}, \vec{\boldsymbol{t}}^{\boldsymbol{REG}}\right) = \operatorname{argmin}\|\boldsymbol{S_{sm}} - \boldsymbol{S}_{st}\|_2 = \operatorname{argmin} \sum_{i=1}^{M^{skel}} \left[P^{sm} - \left(\boldsymbol{R^{REG}} \cdot Q^{st} + \vec{\boldsymbol{t}}^{\boldsymbol{REG}}\right)\right]^2 \quad (16)$$

$$\boldsymbol{R^{REG}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$$\vec{\boldsymbol{t}}^{\boldsymbol{REG}} = [t_x \quad t_y \quad t_z] \tag{18}$$

$$Q^{reg} = \boldsymbol{R^{REG}} \cdot Q^{sm} + \vec{\boldsymbol{t}}^{\boldsymbol{REG}} \tag{19}$$

In the above, M^{skel} denotes the number of skeleton points in $\boldsymbol{S_{sm}}$, Q^{reg} represents a registered point in $\boldsymbol{S_{st}}$, and $\boldsymbol{sm}$ represents the skeleton model.

Recognition

Recognition was performed after the registration completed, as shown in Fig. 11. The members of the two nodes in $\boldsymbol{S_{st}}$ were determined in advance. As shown in Fig. 11, skeleton point P^{sm} in $\boldsymbol{S_{sm}}$ was first paired with the nearest point Q_1^{reg} in $\boldsymbol{S_{st}}$ $\left(P^{sm}, Q_1^{reg}\right)$. Two additional points, Q_2^{reg} and Q_3^{reg} adjacent to point Q_1^{reg} in $\boldsymbol{S_{st}}$ were then extracted from the set. If the two points were marked with the same member IDs, the skeleton point P^{sm} was considered a member point with the corresponding member ID. However, if one of the points (Q_2^{reg} or Q_3^{reg}) was found to be the skeleton point of another member, the skeleton point was considered to be the point near connection P^{con}, and was deleted thereafter.

Algorithm 3: Automatic multi-member recognition from skeleton points

Input: all skeleton points in $\boldsymbol{S_{sm}}$, all skeleton points $\boldsymbol{S_{st}}$, where skeleton-point information in $\boldsymbol{S_{st}}$ includes 3D coordinates and the number of the member to which it belongs.

Output: the members skeleton points $\boldsymbol{S_k^{mem}}$

$\boldsymbol{S_k^{mem}}$=∅, $k \in [1, \text{Number of members}]$

While (size ($\boldsymbol{S_{sm}}$,1)~=0)

P^{sm} = the first point in $\boldsymbol{S_{sm}}$

Find out the three nearest points $Q_1^{reg} - Q_3^{reg}$ in $\boldsymbol{S_{st}}$ corresponding to P^{sm}; the corresponding member number of $Q_1^{reg} - Q_3^{reg}$ are $M_1^{reg} - M_3^{reg}$

if M_1^{reg}=M_2^{reg}=M_3^{reg}=k

Add P^{sm} into $\boldsymbol{S_k^{mem}}$

else

consider P^{sm} as the skeletal point around the nodes

end

Delete P^{sm} from $\boldsymbol{S_{sm}}$

end

This step was performed using MATLAB. The resulting member skeleton points $\boldsymbol{S_k^{mem}}$ were stored in the matrix format in preparation for subsequent member reconstruction. The data volume was approximately 150 KB.

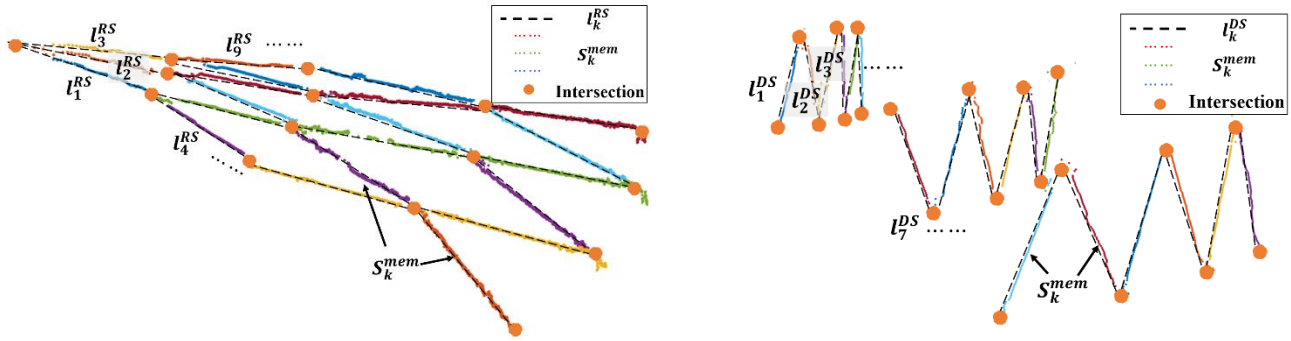
6. Reconstruction of the structural model

Member recognition yielded the skeleton points of members $\boldsymbol{S_k^{mem}}$ that were used to reconstruct the cable-dome model. The first step was to determine the position and orientation of the members for fitting member nodes, as

described in Section 6.1. The nodes connecting the members were obtained from a projection-reprojection member, as described in Section 6.2. The entire cable-dome model was then reconstructed sequentially, as described in Section 6.3. This processing stage was accomplished using MATLAB, in which the data format was inherited from the previous step.

6.1. Determination of position and orientation

The skeleton points of the recognized members were scattered in a structured space that was first linearly aligned, as shown in Fig. 12. The initial bending defects of a member do not affect the overall performance of the structural entity [57]. Therefore, it was assumed that member skeleton $\boldsymbol{S_k^{mem}}$ could be fitted by straight line $\boldsymbol{l_k^{mem}}$ in space (Eq. (20)). Line $\boldsymbol{l_k^{mem}}$ mainly comprised spatial position, $\boldsymbol{P_k^{mem}}$, and direction vector $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ with Δ_k a linear parameter determining the corresponding member's length. The calculations of position $\boldsymbol{P_k^{mem}}$ and direction vector $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ are discussed below.



(a) Ridge struts. (b) Diagonal struts.

Fig. 12. Member skeleton points.

$$\boldsymbol{l_k^{mem}} = \boldsymbol{P_k^{mem}} + \vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}} \cdot \Delta_k, k \in [1, \text{Number of members}] \quad (20)$$

In the above, **mem** indicates either a ridge or diagonal strut. $\boldsymbol{P_k^{mem}}$ represents the spatial position of skeleton member $\boldsymbol{l_k^{mem}}$, whereas $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ is the direction vector of line $\boldsymbol{l_k^{mem}}$. The parameter Δ_k represents the distance parameter for the linear interpolation, $\Delta_k \in \left[-\frac{|\boldsymbol{l_k^{mem}}|}{2}, \frac{|\boldsymbol{l_k^{mem}}|}{2}\right]$, and $|\boldsymbol{l_k^{mem}}|$ is the length of the corresponding member.

Position of a skeleton member

Position $\boldsymbol{P_k^{mem}}$ was defined as the centroid of skeleton member $\boldsymbol{S_k^{mem}}$. It should be noted that the point density of a skeleton member was uniform following the previous processing. Therefore, the centroid coordinates were simply the average of the skeleton member points in a member, as expressed by Eq. (21).

$$\boldsymbol{P_k^{mem}}:\begin{cases}\bar{x}_k^{mem}=\dfrac{1}{M_k^{mem}}\sum_{i=1}^{M_k^{mem}}x_{k,t}^{mem}\\ \bar{y}_k^{mem}=\dfrac{1}{M_k^{mem}}\sum_{i=1}^{M_k^{mem}}y_{k,t}^{mem}\,,t\in[1,M_k^{mem}]\\ \bar{z}_k^{mem}=\dfrac{1}{M_k^{mem}}\sum_{i=1}^{M_k^{mem}}z_{k,t}^{mem}\end{cases}\tag{21}$$

In the above, $(\bar{x}_k^{mem},\bar{y}_k^{mem},\bar{z}_k^{mem})$ and $\left(x_{k,t}^{mem},y_{k,t}^{mem},z_{k,t}^{mem}\right)$ are the coordinates of the centroid and points of skeleton member $\boldsymbol{S_k^{mem}}$, respectively. The skeleton member $\boldsymbol{S_k^{mem}}$ comprised the total number M_k^{mem} of skeleton points.

Orientation of skeleton member $\boldsymbol{S_k^{mem}}$

The spatial direction vector $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$, of line $\boldsymbol{l_k^{mem}}$, was determined using the singular value decomposition (SVD) method. The set $\boldsymbol{S_k^{mem}}$ was first normalized to improve the fit accuracy [56]. The values of σ_k^x, σ_k^y, σ_k^z, were calculated as the base point of the normalization in Eq. (22). The coordinates of the skeleton points in a member were then sequentially normalized using Eq. (23). Thus, a normalized member skeleton, $\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$, was obtained for further analysis.

$$\begin{cases}\sigma_k^x=\sqrt{\dfrac{\sum_1^{M_k^{mem}}\left(x_{k,t}^{mem}-\bar{x}_k^{mem}\right)^2}{M_k^{mem}}}\\ \sigma_k^y=\sqrt{\dfrac{\sum_1^{M_k^{mem}}\left(y_{k,t}^{mem}-\bar{y}_k^{mem}\right)^2}{M_k^{mem}}}\\ \sigma_k^z=\sqrt{\dfrac{\sum_1^{M_k^{mem}}\left(z_{k,t}^{mem}-\bar{z}_k^{mem}\right)^2}{M_k^{mem}}}\end{cases}\tag{22}$$

$$\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}}\begin{cases}\hat{\boldsymbol{x}}_{\boldsymbol{k,t}}^{\boldsymbol{mem}}=(x_{k,t}^{mem}-\bar{x}_k^{mem})/\sigma_k^x\\ \hat{\boldsymbol{x}}_{\boldsymbol{k,t}}^{\boldsymbol{mem}}=(y_{k,t}^{mem}-\bar{y}_k^{mem})/\sigma_k^y\\ \hat{\boldsymbol{x}}_{\boldsymbol{k,t}}^{\boldsymbol{mem}}=(z_{k,t}^{mem}-\bar{z}_k^{mem})/\sigma_k^z\end{cases}\tag{23}$$

In the above, $\left(\hat{x}_{k,t}^{mem},\hat{y}_{k,t}^{mem},\hat{z}_{k,t}^{mem}\right)$ are the coordinates of a point in $\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$.

The normalized skeleton member set, $\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$, was assumed factorizable by left singular matrix **U**, singular values matrix $\sum$, and right singular matrix $\boldsymbol{V}$, as shown in Eq. (24). The vector $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ was then operated by $\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem^T}}\cdot\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ that provided a symmetric property. The $\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem^T}}\cdot\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ product was expanded as shown in Eq. (25). The pivot unit vector $\vec{\boldsymbol{v}}_{\boldsymbol{1}}$ of $\boldsymbol{V}$ represents the major direction of the aligned points, and is denoted as $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$. Based on the orthogonality, both sides of Eq. (25) were simultaneously multiplied by $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$, yielding Eq. (26). The right-hand side of the equation was shifted to the left-hand side, as shown in Eq. (27). The determinant of $\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem^T}}\cdot\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}}-\lambda_1\cdot\boldsymbol{I}$ was then solved where the largest eigenvalue λ_1 corresponding to the unit vector $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ was found (Eq. (28)). The result λ_1 was substituted into Eq. (27), yielding vector $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$.

Because the vector $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ was a non-zero vector, λ_1 was obtained by solving the determinant that allowed to solve for $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ in Eq. (28).

$$\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}}{}_{M_k^{mem}\times 3} = \mathbf{U}_{M_k^{mem}\times M_k^{mem}} \cdot \sum_{M_k^{mem}\times 3} \cdot \boldsymbol{V}^{\boldsymbol{T}}{}_{3\times 3} \tag{24}$$

$$\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem^T}} \cdot \widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}} = \boldsymbol{V} \cdot \boldsymbol{\Sigma^2} \cdot \boldsymbol{V^T} = \begin{bmatrix} \vec{\boldsymbol{v}}_1 \\ \vec{\boldsymbol{v}}_2 \\ \vec{\boldsymbol{v}}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \lambda_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_3 \end{bmatrix} [\vec{\boldsymbol{v}}_1 \quad \vec{\boldsymbol{v}}_2 \quad \vec{\boldsymbol{v}}_3] \tag{25}$$

$$\left(\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem^T}} \cdot \widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}}\right) \cdot \vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}} = \begin{bmatrix} \vec{\boldsymbol{v}}_1 \\ \vec{\boldsymbol{v}}_2 \\ \vec{\boldsymbol{v}}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \lambda_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_3 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \lambda_1 \cdot \vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}} \tag{26}$$

$$\left(\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem^T}} \cdot \widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}} - \lambda_1 \cdot \boldsymbol{I}\right) \cdot \vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}} = \vec{\mathbf{0}} \tag{27}$$

$$\left|\widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem^T}} \cdot \widehat{\boldsymbol{S}}_{\boldsymbol{k}}^{\boldsymbol{mem}} - \lambda_1 \cdot \boldsymbol{I}\right| = 0 \tag{28}$$

Position $\boldsymbol{P}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ and orientation $\vec{\boldsymbol{v}}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ were found. The fit line $\boldsymbol{l}_{\boldsymbol{k}}^{\boldsymbol{mem}}$ was then interpolated using the linear parameter in Eq. (20) for the skeleton members. The nodes of the skeleton structural cable dome were then determined based on the member lines, as described below.

6.2. Determination of nodes

The structural modeling of a cable dome requires reconstructed multi-section members connected by joints. Considering that joints are represented by nodal points in a general structural analysis, it was assumed that the joints of a cable dome in structural modeling could be simplified as nodes. However, the nodes could not be determined directly from the intersection of the member lines, because these lines did not spatially intersect (Fig. 13). Instead, an intersection was observed when these spatial lines were projected onto a common plane. A projection-reflection method for finding a node was proposed. The x and y coordinates nodes were determined from the intersection of the projected lines. The z-coordinates of the nodes were determined using the reprojection method.

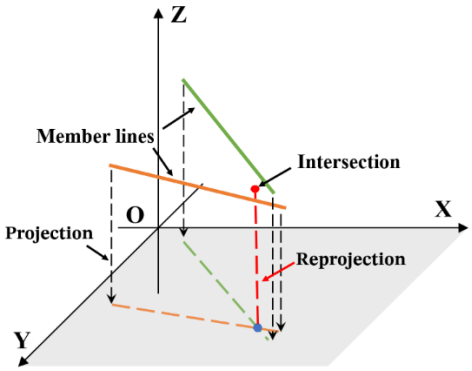


Fig. 13. The intersection of member lines.

The nodes were categorized into three types based on the connection relationship of the lines: 1) a two-member node (TMN) refers to a node that connects two members, 2) a multi-member node (MMN) refers to a node that

connects multiple members, and 3) a single-member node (SMN) refers to a node that connects multiple members, as shown in Fig. 14.

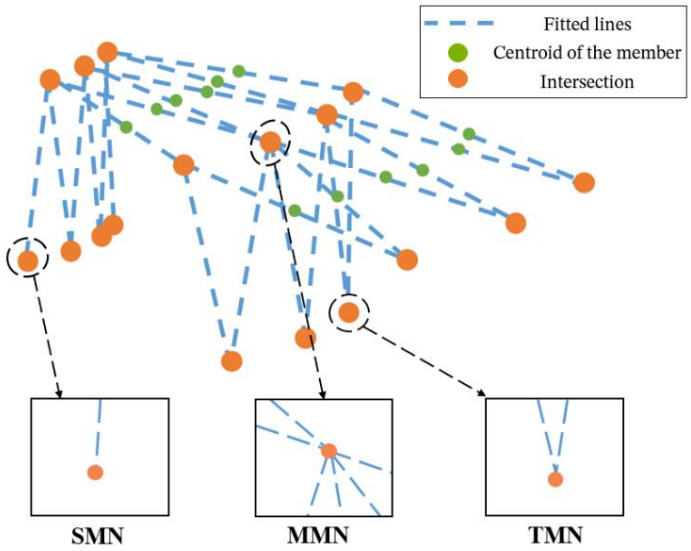


Fig. 14. Classification of nodes.

Two-member nodes (TMNs):

A TMN, P_{TMN}, connected two member lines, denoted by $\boldsymbol{l_a}$ and $\boldsymbol{l_b}$, respectively. $\boldsymbol{l_a}$ and $\boldsymbol{l_b}$ were represented mathematically by Eq. (27a) and Eq. (27b), where the coefficients were obtained as explained in the previous section. The two lines were first projected onto the $Z = 0$ plane, $\bar{\boldsymbol{l}}_{\boldsymbol{a,XY}}$ and $\bar{\boldsymbol{l}}_{\boldsymbol{b,XY}}$ (Eq. (28a) and Eq. (28b)). It was assumed that the two projected lines intersected (Eq. (29)). Thus, the linear parameter $\Delta_a = \Delta_b = \Delta^{NID}$ was derived, as shown in Eq. (30).

$$\boldsymbol{l_a} = \boldsymbol{P_a^{mem}} + \vec{\boldsymbol{v}}_{\boldsymbol{a}}^{\boldsymbol{mem}} \cdot \Delta_a = \begin{bmatrix} \bar{x}_a^{mem} \\ \bar{y}_a^{mem} \\ \bar{z}_a^{mem} \end{bmatrix} + \vec{\boldsymbol{v}}_{\boldsymbol{a}}^{\boldsymbol{mem}} \cdot \Delta_a \tag{27a}$$

$$\boldsymbol{l_b} = \boldsymbol{P_b^{mem}} + \vec{\boldsymbol{v}}_{\boldsymbol{b}}^{\boldsymbol{mem}} \cdot \Delta_b = \begin{bmatrix} \bar{x}_b^{mem} \\ \bar{y}_b^{mem} \\ \bar{z}_b^{mem} \end{bmatrix} + \vec{\boldsymbol{v}}_{\boldsymbol{b}}^{\boldsymbol{mem}} \cdot \Delta_b \tag{27b}$$

$$\bar{\boldsymbol{l}}_{\boldsymbol{a,XY}} = \bar{\boldsymbol{P}}_{\boldsymbol{a}}^{\boldsymbol{mem}} + \overline{\vec{\boldsymbol{v}}_{\boldsymbol{a}}^{\boldsymbol{mem}}} \cdot \Delta_a = \begin{bmatrix} \bar{x}_a^{mem} \\ \bar{y}_a^{mem} \end{bmatrix} + \overline{\vec{\boldsymbol{v}}_{\boldsymbol{a}}^{\boldsymbol{mem}}} \cdot \Delta_a \tag{28a}$$

$$\bar{\boldsymbol{l}}_{\boldsymbol{b,XY}} = \bar{\boldsymbol{P}}_{\boldsymbol{b}}^{\boldsymbol{mem}} + \overline{\vec{\boldsymbol{v}}_{\boldsymbol{b}}^{\boldsymbol{mem}}} \cdot \Delta_b = \begin{bmatrix} \bar{x}_b^{mem} \\ \bar{y}_b^{mem} \end{bmatrix} + \overline{\vec{\boldsymbol{v}}_{\boldsymbol{b}}^{\boldsymbol{mem}}} \cdot \Delta_b \tag{28b}$$

$$\bar{\boldsymbol{l}}_{\boldsymbol{a,XY}} = \bar{\boldsymbol{l}}_{\boldsymbol{b,XY}} \tag{29a}$$

$$\begin{bmatrix} \bar{x}_a^{mem} \\ \bar{y}_a^{mem} \end{bmatrix} + \overline{\vec{\boldsymbol{v}}_{\boldsymbol{a}}^{\boldsymbol{mem}}} \cdot \Delta_a = \begin{bmatrix} \bar{x}_b^{mem} \\ \bar{y}_b^{mem} \end{bmatrix} + \overline{\vec{\boldsymbol{v}}_{\boldsymbol{b}}^{\boldsymbol{mem}}} \cdot \Delta_b \tag{29b}$$

$$\Delta^{NID} = \frac{\bar{x}_a^{mem} - \bar{x}_b^{mem}}{\overline{\vec{\boldsymbol{v}}_{\boldsymbol{b}}^{\boldsymbol{mem}}} - \overline{\vec{\boldsymbol{v}}_{\boldsymbol{a}}^{\boldsymbol{mem}}}} \tag{30}$$

In the above, Δ_a, Δ_b, and Δ^{NID} represent the linear parameters of the two lines and node, respectively. NID represents the node ID, as described in the INAM of a structural module.

The intersection coordinates P_{TMN} were calculated as follows: The values of x_{ab} and y_{ab} were calculated using Eqns. (31) and (32), whereas the value of the linear parameter, Δ^{NID}, was obtained from Eq. (30) . The dimensional differences between the two members were assumed negligible. Therefore, z_{ab} was obtained by averaging the z values corresponding to l_a and l_b.

$$x_{ab} = \bar{x}_a^{mem} + \vec{\boldsymbol{v}}_{\boldsymbol{a}}^{\boldsymbol{mem}} \cdot \Delta^{NID} \tag{31}$$

$$y_{ab} = \bar{y}_a^{mem} + \vec{\boldsymbol{v}}_{\boldsymbol{a}}^{\boldsymbol{mem}} \cdot \Delta^{NID} \tag{32}$$

$$z_{ab} = \frac{1}{2}\sum z_i \,, i = a \text{ or } b \tag{33}$$

where $z_a = \bar{z}_a^{mem} + \vec{\boldsymbol{v}}_{\boldsymbol{a}}^{\boldsymbol{mem}} \cdot \Delta^{NID}, z_b = \bar{z}_b^{mem} + \vec{\boldsymbol{v}}_{\boldsymbol{b}}^{\boldsymbol{mem}} \cdot \Delta^{NID}$

Multi-member nodes (MMNs):

For the MMN, the node coordinate-fitting process resembles that of the TMN. The members connected by the node were grouped two by two, with a total of $M^{NID} = M^{con} \cdot (M^{con} + 1)/2$ groups, with M^{con} representing the number of members. The coordinates of P_1–$P_{M^{NID}}$ of the groups were obtained by solving them using the projection-reflection method. For example, P_1 connected two member lines $\boldsymbol{l_i}$ and $\boldsymbol{l_j}$. The coordinates x_{ij}, y_{ij}, z_{ij} of P_1 were found by solving for the linear parameter Δ^{NID} (Eqns. (34a–34d)). The final coordinates were the centers of P_1–$P_{M^{NID}}$, obtained using Eq. (35).

$$\Delta^{NID} = \frac{\bar{x}_i^{mem} - \bar{x}_j^{mem}}{\vec{\boldsymbol{v}}_{\boldsymbol{j}}^{\boldsymbol{mem}} - \vec{\boldsymbol{v}}_{\boldsymbol{i}}^{\boldsymbol{mem}}} \tag{34a}$$

$$x_{ij} = \bar{x}_i^{mem} + \vec{\boldsymbol{v}}_{\boldsymbol{i}}^{\boldsymbol{mem}} \cdot \Delta^{NID} \tag{34b}$$

$$y_{ij} = \bar{y}_i^{mem} + \vec{\boldsymbol{v}}_{\boldsymbol{i}}^{\boldsymbol{mem}} \cdot \Delta^{NID} \tag{34c}$$

$$z_{ij} = \frac{1}{2}\sum z_k \,, k = i \text{ or } j \tag{34d}$$

where $z_i = \bar{z}_i^{mem} + \vec{\boldsymbol{v}}_{\boldsymbol{i}}^{\boldsymbol{mem}} \cdot \Delta^{NID}, z_j = \bar{z}_j^{mem} + \vec{\boldsymbol{v}}_{\boldsymbol{j}}^{\boldsymbol{mem}} \cdot \Delta^{NID}$

$$\begin{aligned} x_M &= \frac{1}{M^{NID}}\sum_1^{M^{NID}} x_i \\ y_M &= \frac{1}{M^{NID}}\sum_1^{M^{NID}} y_i, i \in [1, M^{NID}] \\ z_M &= \frac{1}{M^{NID}}\sum_1^{M^{NID}} z_i \end{aligned} \tag{35}$$

In the above, (x_M, y_M, z_M) and (x_i, y_i, z_i) are the coordinates of the final intersection and the intersection obtained in a group, respectively. The parameter M represents the member IDs intersected by the node.

Single-member nodes (SMNs):

The SMN is typically located at the edge of the structural module. The calculation of the SMN is generally performed after those of the other nodes. Therefore, the positions of the connecting members and nodes adjacent to the SMN were used to extrapolate the SMN coordinates using Eq. (36).

$$\begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \begin{bmatrix} \bar{x}_s^{mem} \\ \bar{y}_s^{mem} \\ \bar{z}_s^{mem} \end{bmatrix} + \vec{\boldsymbol{v}}_{\boldsymbol{s}}^{\boldsymbol{mem}} \cdot \Delta^{NID}, \Delta^{NID} = \frac{\bar{x}_s^{mem} - x_M}{\vec{\boldsymbol{v}}_{\boldsymbol{s}}^{\boldsymbol{mem}}} \tag{36}$$

In the above, x_S, y_S, z_S are the coordinates of P_s. $[\bar{x}_s^{mem}, \bar{y}_s^{mem}, \bar{z}_s^{mem}]$ and $\vec{\boldsymbol{v}}_{\boldsymbol{s}}^{\boldsymbol{mem}}$ are the central points and directional vectors of the line intersecting the node, respectively. The parameter S represents the line ID. The parameter x_M represents the x coordinate of the node at the other end of the line.

6.3. Reconstruction of a cable dome

The reconstruction of the cable dome was initiated by sorting the data structures of the modules (Table 3). The IDs of the skeleton members and connecting nodes followed those of the standard model $\boldsymbol{S}_{\boldsymbol{st}}$ from the INAM in Fig. 10(b). Nodes *i* and *j* indicate the coordinates of the nodes of the corresponding members that were later used in the module assembly. The class, cross-section, and length of the columns can be later used in structural modeling and future applications.

Table 3

The data structure of the module model.

Member ID	Nodes ID[a]	Node i (m)			Node j (m)			Class[b]	Cross-section[c] (mm)	Length (mm)
		x	*y*	*z*	*x*	*y*	*z*			
1	1/2	0	0	0	1.1880	0.1583	-0.1826	L	Φ30×3	1212.3
2	1/3	0	0	0	1.1435	0.4822	-0.1954	L	Φ30×3	1256.3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
11	6/9	2.3580	0.6161	-0.4089	3.6593	0.4399	-0.6574	L	Φ40×3	1336.5
12	6/10	2.3580	0.6161	-0.4089	3.4254	1.3631	-0.6439	L	Φ40×3	1323.8
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
35	9/24	3.6593	0.4399	-0.6574	3.5571	0.9164	-1.4542	L	Φ40×3	934.0
36	10/24	3.4254	1.3631	-0.6439	3.5571	0.9164	-1.4542	L	Φ40×3	934.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
60	24/25	3.5571	0.9164	-1.4542	3.0587	1.7491	-1.4277	C	Φ12	970.4
61	25/26	3.0587	1.7491	-1.4277	2.4837	2.5106	-1.4473	C	Φ12	954.2

a: Node ID represents the global Node ID at the two ends of the members: node a/node b.

b: The member form consists of straight lines (L) and curves (C) dictating the respective member-modeling methods.

c: The cross-section of struts are traffic circles, with the dimension of Φ radius of the outer circle × tube thickness. The cross-sections of the cables are circles with dimensions of the radius of the circle.

As the previous processing was conducted on the segmented modules sequentially, the processed modules were assembled where the edge nodes of the modules were registered, as shown in Fig. 15. The assembly was done in two stages: 1) a coarse global assembly and 2) a fine local assembly. The coarse assembly simply registered Node 1 of

all the structural modules together. However, other edge nodes may not coincide with each other because of the manual module segmentation. The shared nodes, for example, Node 16 in Module 1 and Node 19 in Module 8, were combined by averaging the coordinates of the pairs of nodes. This was performed on individual modules until the cable-dome skeleton model was obtained.

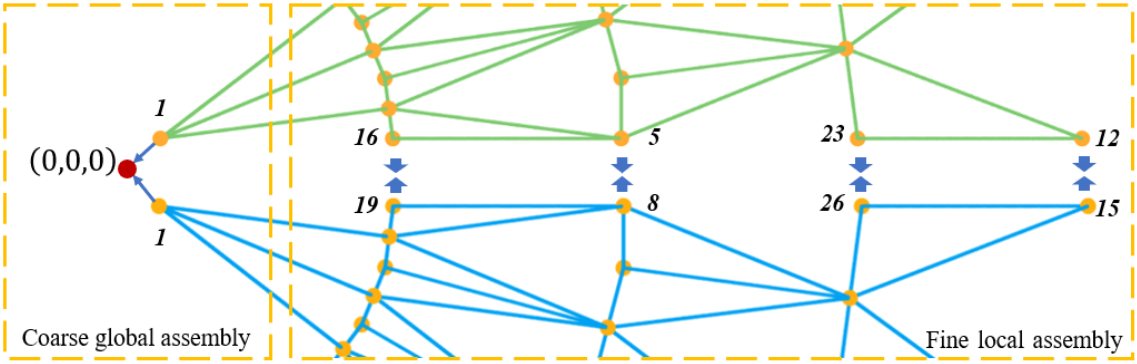


Fig. 15. Coarse global assembly and fine local assembly.

The reconstructed structural model is shown in Fig. 16. The ridge struts, diagonal struts, diagonal cables, and ring-loop cables are colored green, blue, orange, and yellow, respectively, according to the structured data. Evidently, the structural model captures the actual geometric features of the cable-dome structure. The structured data of the structural model can be easily used for further development of numerical simulations or building information modeling (BIM) design.

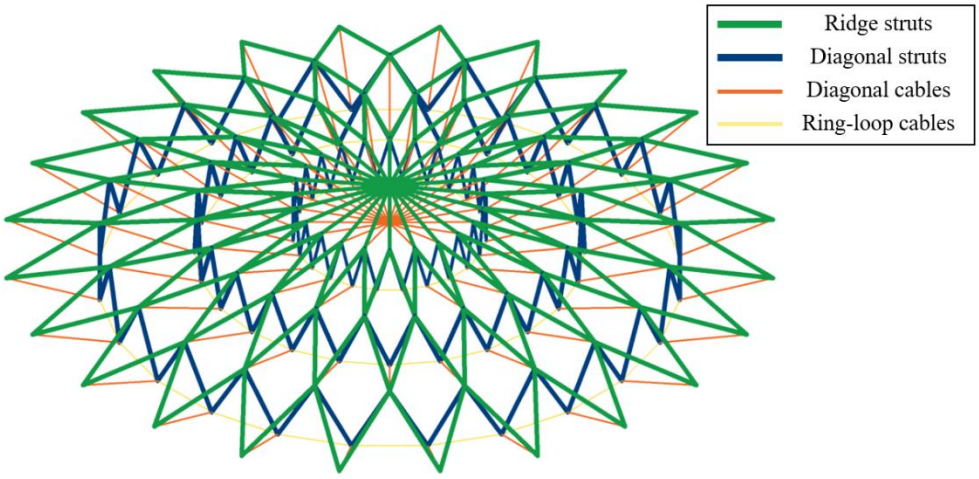


Fig. 16. The cable-dome structure model.

7. Discussion

This section first validates the proposed MASM method in terms of the processing accuracy and computational efficiency; this is done in Section 7.1. In Section 7.2, the applications of the method to structural deviation inspection and monitoring are demonstrated in detail. The novel aspects of the proposed MASM method are discussed in detail in Section 7.3. Section 7.4 discusses the limitations of the proposed method that require future development.

435 **7.1. Validation of the MASM method**

436 The MASM method was validated in terms of both the reconstruction accuracy and computational efficiency,
437 which are technical concerns in general structural modeling that uses laser measurements.

438 Accuracy validation

439 Accuracy validation amounted to evaluating differences between the nodes of the MASM method-based model
440 and those of the "ground truth" model, because nodes are important for structural analysis [58]. In addition, a true
441 reference model was manually built according to the preprocessed PCD cable dome, as shown in Fig. 17, using the
442 commercial software Geomagic Wrap [59]. First, the members' PCDs were manually and visually recognized and
443 then segmented from member to member. The segmented PCD members were sequentially fitted with the
444 corresponding 3D shapes, using the RANSAC function in the software. The centerlines of the members were then
445 plotted, the intersections of which were defined as reference nodes. Thus, the integrity of the reference nodes
446 constituted the true reference model. Constructing the true reference model required substantial manual work and
447 was time-consuming. The modeling was primarily used to validate the accuracy of the MASM method-based model.

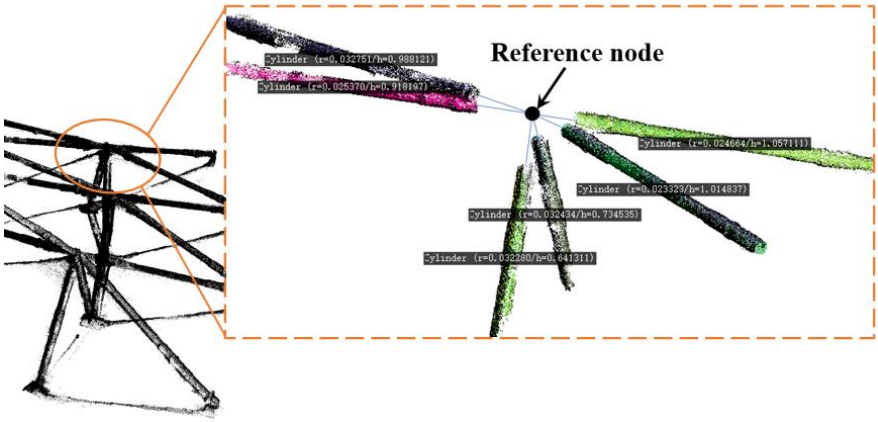


Fig. 17. Reference point extraction.

448 Accuracy validation then started by comparing the nodes of the MASM method-based model to those of the
449 reference model (Eq. (37)). The fit error δ_{MASM} was defined as the 2-norm distance between P_{MASM} and P_{ref},
450 corresponding to the nodes of the MASM method-based model and the reference model, respectively.

$$\delta_{MASM} = \left\| P_{ref} - P_{MASM} \right\|_2 = \sqrt{\left(x_{ref} - x_{MASM}\right)^2 + \left(y_{ref} - y_{MASM}\right)^2 + \left(z_{ref} - z_{MASM}\right)^2} \tag{37}$$

451 In the above, δ_{MASM} represents the fit error of P_{MASM}. $(x_{ref}, y_{ref}, z_{ref})$ and $(x_{MASM}, y_{MASM}, z_{MASM})$ are the
452 coordinates of P_{ref} and P_{MASM}, respectively.

The fit errors of the nodes were statistically analyzed, and the results are shown in Fig. 18. The histogram shows the distribution of the fit errors. The probability density function (PDF) followed a Gaussian distribution with a mean of 1.384 mm and standard deviation of 0.39 mm. The 95% confidence interval (CI) was [1.227, 1.534]. Considering the span of the cable dome, the accuracy of the MASM method-based model varied within 0.02%. According to the technical specifications for space-frame structures [60], a typical constructional deviation is in the 35–185 mm range, which is 10 times greater than the fit errors in the present study. The signal-to-noise ratio (SNR) was sufficiently high, such that the errors did not impact the monitoring and analysis of constructional deviations[61]. Thus, the accuracy of the MASM method was satisfactory.

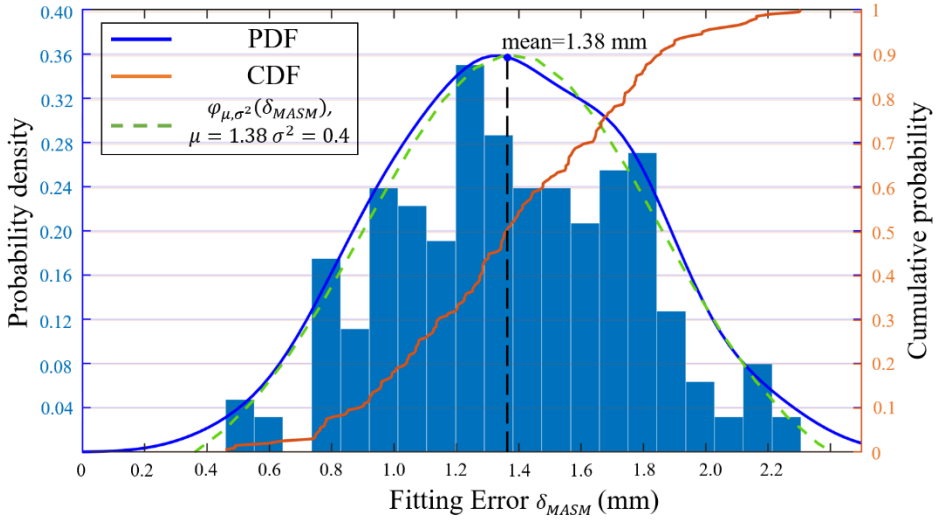


Fig. 18. Node-fitting errors.

Efficiency validation

The proposed MASM method targets the automatic structural modeling of a multimember-section cable dome based on the PCD laser measurements. Unfortunately, none of these structural modeling methods are applicable to cable-dome structures. Reference to efficiency validation allows to select the most updated structural modeling method for steel structures with comparable computational complexity [39]. Runtime tests were conducted for both structural modeling methods to demonstrate the computational efficiency of the proposed MASM method. The runtime tests were performed on a personal computer equipped with an AMD Ryzen7 5800H @3.20GHz processor. The runtimes of the structural modeling methods were recorded using the tic-toc function in MATLAB (Table 4).

PCD structural modeling generally includes two stages: 1) skeleton extraction and 2) model reconstruction. However, the proposed MASM method implemented the module segmentation of the cable-dome PCD according to the features of the structural configuration in advance. Module segmentation divided the cable-dome PCD into eight structural modules, which took approximately 160.45 s. The skeleton extraction took approximately 3.28 s for the structural module and approximately 21.18 s for the total 8 pieces of modules. The reconstruction stage took

approximately 4.63 s for the module and 37.97 s for the entity. The total runtime was 221.95 s for data containing approximately 12,661,192 points.

The reference BIM reconstruction method (BRM) processed steel structures with a volume of approximately 22,213,417 points. It utilized Laplacian-based central skeleton contraction and the rolling sphere algorithm to extract the skeleton of the model, which took approximately 25,342.69 s. The reconstruction of the 3D model required approximately 3,589 s to achieve the entire structural model.

TABLE 4

SDT modeling time.

Methods	**Stages**	**Time**	
MASM[a]	S1: Module segmentation	160.45 s	
	S2: Skeleton extraction	Module	Entity
	Multi-member central shrinkage for skeleton-point recognition	2.87 s	18.75 s
	Smart multi-member recognition	0.41 s	2.43 s
	Subtotal of S2	3.28 s	21.18 s
	S3: Reconstruction of structural model	Module	Entity
	Determinations of positions, orientations, and nodes	4.63 s	37.97 s
	Reconstruction of a cable dome	2.35 s	
	Subtotal of S3	40.32 s	
	Total	221.95 s	
BRM[b][39]	S1: Skeleton extraction		
	Laplacian-based central skeleton contraction	18342.58 s	
	Central axis candidate extraction	5891.21 s	
	Central axis refinement	948.78 s	
	Regional growing-based central axis segmentation	160.12 s	
	Subtotal of S1	25,342.69 s	
	S2: Reconstruction of structural model		
	PCD segmentation	356.15 s	
	Central axis curve estimation	2873.21 s	
	3D modeling	359.67 s	

	Subtotal of S2	3,589.03 s
	Total	28931.72 s
MASM v.s. BRM[d] (%)		0.14%

Notes: a. MASM represents the automatic structural modeling method.

b. BRM represents the BIM reconstruction method.

d. Comparison of the runtimes of the two methods was performed using the same amount of data. $Ratio = (Time\ of\ MASM/Time\ of\ BRM) \times 100\%$.

It was observed that the runtimes associated with the extraction of members were 2.43 s and 160.12 s, respectively, for the MASM and BRM methods. The runtimes associated with the reconstructions of structural models were 2.35 s and 359.67 s, for the MASM and BRM methods, respectively. Although the MASM method featured an additional module-segmentation stage, this additional step greatly improved the computational efficiency of the latter two stages. Consequently, the total performance time of the MASM method was only 0.14% that of the BRM method. Thus, the efficiency of the MASM method was demonstrated.

7.2. Application to deviation monitoring

Currently, construction deviation monitoring is considered the most important aspect of any structural project. Structural modeling using 3D laser measured PCD provides better access to deviation monitoring than manual or local measurements. The previous sections described an analysis-ready cable-dome model with structured data. The cable-dome model was aligned with the design model using the center struts of the two structural models. The deviations can be determined from the distances between the node pairs, using Eq. (38).

$$D^{NID} = \left\| P_{design} - P_{MASM} \right\|_2 = \sqrt{\left(x_{design} - x_{MASM}\right)^2 + \left(y_{design} - y_{MASM}\right)^2 + \left(z_{design} - z_{MASM}\right)^2} \tag{38}$$

In the above, P_{design} and P_{MASM} are the nodes from the design model and those fitted using the SDT modeling method, respectively. D^{NID} represents the deviation at a node from P_{design}. $\left(x_{design}, x_{design}, x_{design}\right)$ and $\left(x_{MASM}, x_{MASM}, x_{MASM}\right)$ are the coordinates of P_{design} and P_{MASM}, respectively.

The comparison results are presented in Fig. 19. The upper cable dome is a design model in which the nodes are colored in orange. The lower cable dome is the MASM cable-dome model, in which the nodes are colored in red. Differences between red and orange nodes are shown in the bottom mesh plot. The maximum deviation was 33.2 mm and the average was approximately 21.7 mm. This was attributed to the deviation of the actual construction from the ideal state, in which the lift to the ridge struts through tensioning was affected by the stiffness of the members.

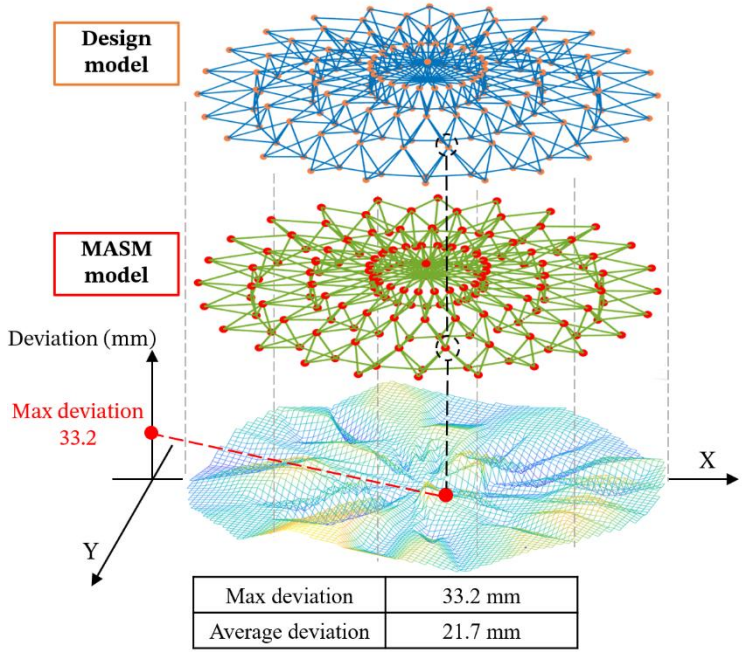


Fig. 19. Deviation of nodes.

7.3. Novel aspects of the MASM method

Structural modeling methods for laser-based PCD generally involve two major steps: 1) skeleton extraction and 2) structural reconstruction. However, traditional structural modeling methods generally have high computational complexity, owing to the large volume of the PCD data associated with a structural entity. The cable-dome structure in this study was formed using identical structural units. Therefore, the large amount of the PCD data associated with the cable dome can be segmented into small PCD modules, as in Stage 1 of the MASM method. Module segmentation is simple, and can be implemented in most commercial software. However, segmentation applies to most steel structures with identical structural units, which effectively decreases the computation complexity associated with further processing, as listed in Table 4.

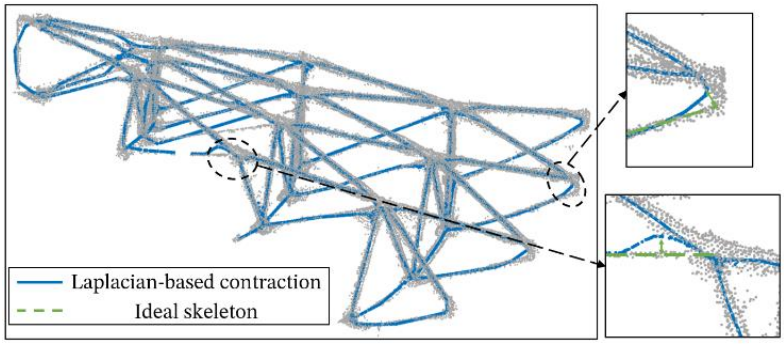
The novel aspect of the MASM method is not module segmentation but the development of skeleton-processing algorithms for cable domes with various cross-section members. The benefits of the developed algorithms for various-section cable domes were demonstrated by comparing with existing popular methods.

Comparison of Skeleton-point Recognition Methods

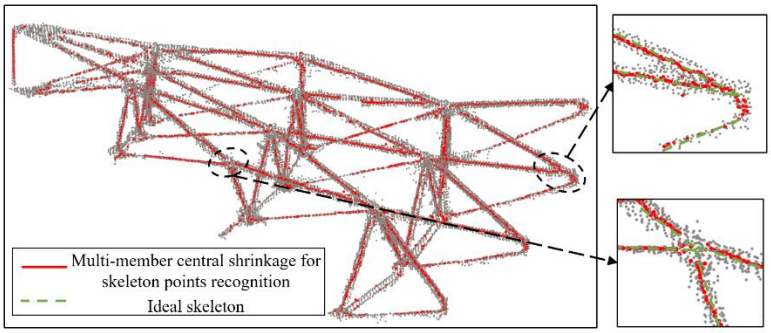
Skeleton-point recognition is a key step in which the central points of the members are recognized from volumetric PCDs. The MASM method proposes a novel algorithm called multimember central shrinkage for skeleton-point recognition, to denoise and extract skeleton points from unstructured multi-sectional cable-dome PCDs. This algorithm processes a layered PCD, in which the centroids of the clusters are identified sequentially. Thus, the centroids of the clusters are considered as skeleton points, as described in Section 4.3.

By contrast, general skeleton-point recognition uses a Laplacian-based skeleton contraction method [45]. The Laplacian matrix comprises the neighborhood of the target point based on the distance relationship. The eigenvalues of the Laplacian matrix are obtained from eigen-decomposition, where the small and large eigenvalues are separated into two groups. Points with small eigenvalues are considered contraction points and discarded. Conversely, points with large eigenvalues are considered attraction points and are retained. The procedure is repeated until the error is minimized.

The two algorithms for skeleton-point recognition were applied to cable-dome module PCDs, and the superiority of the proposed multi-member central shrinkage algorithm was validated. The Laplacian-based skeleton-contraction algorithm performed poorly in the corner regions, as shown in Fig. 20(a). This discrepancy arose from the variation in the cross-sectional dimensions of the members, where the Laplacian-based skeleton-contraction method allowed only uniform contractions. The proposed multi-member central shrinkage method performed well, and the algorithm was implicitly adapted to geometric variations (Fig. 20(b)).



(a) Skeleton points from Laplacian-based contraction method.



(b) Results of multi-member central shrinkage for skeleton-point recognition method.

Fig. 20. Comparison of skeleton-point recognition methods.

The computational times and accuracies are compared in detail in Table 5. The Laplacian-based skeleton contraction required longer time than the multi-member central shrinkage method, with 90.34 s against 2.87 s per structural module. The accuracy, noise, and distortion of the skeleton points were statistically analyzed. These results demonstrate that the noise ratios for Laplacian-based skeleton contraction and multi-member central shrinkage for

skeleton-point recognition methods were similar, at 1.74% and 1.55%, respectively. However, the distortion of the Laplacian-based skeleton-contraction method was much larger than that of the multi-member central shrinkage method for skeleton-point recognition (11.74% vs. 3.16%).

Thus, the proposed multi-member central shrinkage method for skeleton-point recognition is more effective than existing methods, when processing cable domes with members of various cross-sections.

TABLE 5

Comparison of skeleton-extraction methods.

Method	**Time**[a]	**Shrinkage accuracy**			
		Noise point	Percentage[b]	Distortion point	Percentage[c]
Multi-member central shrinkage for skeleton-point recognition	20.05 s	65	1.74%	118	3.16%
Laplacian-based skeleton contraction	90.33 s	332	1.55%	2514	11.74%

Notes: Module 1 PCD were used for both methods in the comparison process.

b: Comparison of the number of noise-related skeleton points with the total number of skeleton points. Percentage = $(number\ of\ noise\ skeleton\ points\ /number\ of\ total\ skeleton\ points) \times 100\%$.

c: Comparison of the number of distortion-related skeleton points with the total number of skeleton points. Percentage = $(number\ of$ distortion $skeleton\ points\ /number\ of\ total\ skeleton\ points) \times 100\%$.

Comparison of Member-recognition Methods

Member recognition is another important step in the skeleton-extraction process, in which members are automatically recognized for future reconstruction. The MASM method uses a novel smart multi-member recognition algorithm that automatically recognizes multi-section members composed of skeleton points according to a standard model. The algorithm was compared to a general member-recognition algorithm, the region-growing method [51]. The region-growing method applies a test to the curvatures and angles of a random point and its neighbors in the structural module. Thresholds are set for the curvatures and angles. Points within the boundaries defined by the thresholds are assumed to belong to a member. The procedure is iterated until all points are processed.

Similar to the Laplacian-based skeleton-contraction algorithm, the region-growing method did not perform well on cable domes with various section members, with respect to the automatic recognition of members (Fig. 21). The fixed threshold was not adequate for dealing with intersections of multiple members.

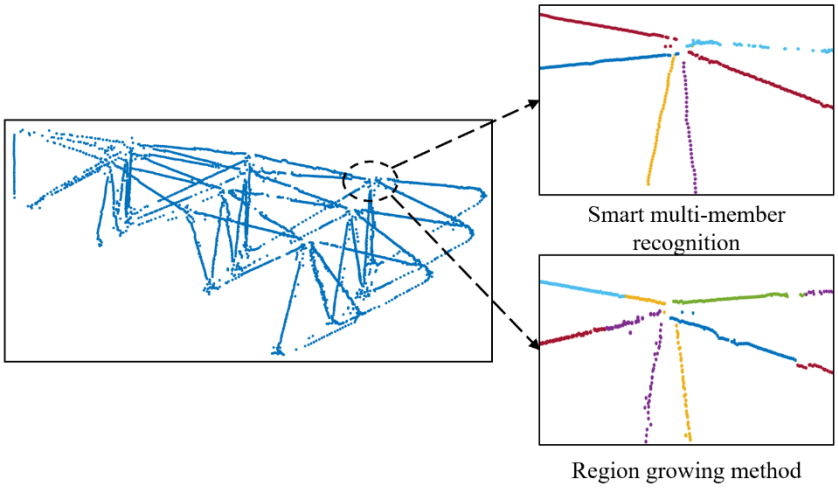


Fig. 21. Comparison of member-recognition methods.

The computational times and accuracies of the two algorithms were compared and are summarized in Table 6. For the region-growing method, most members were not fully recognized. The correctness of the recognition was low, with only four out of the 39 members perfectly recognized. However, the smart multi-member recognition method yielded satisfactory results. The false positive rate was 7.69%. The runtimes of the two methods were also compared. Taken together, the results suggest that the smart multi-member recognition method was more efficient than the existing region-growing method.

TABLE 6

Comparison of member-recognition methods.

Methods	**Time**[a]	**Recognition error**	
		Wrong Members	Percentage[b]
Smart multi-member recognition	0.41s	3	7.69%
Region growing	2.57s	35	89.74%

Notes: Module 1 PCD were used for both methods in the comparison process.

b: Comparison of the wrong members with the members' number. Percentage $= (number\ of$ wron$g\ members\ /number\ of\ total\ members) \times 100\%$.

The proposed MASM method not only encapsulates key algorithms for processing but also provides a systematic automatic cable-dome structural modeling approach that integrates processing algorithms adapted to cable-dome structures. The novel aspects of the algorithm and methods, and the potential benefits for applications, were validated.

7.4. Limitations of the MASM method

The proposed MASM method is applicable to cable-dome structural modeling with volumetrically measured PCD. Although this method is novel and has important advantages, it is limited by several factors, as discussed below.

The accuracy of the MASM method is affected by the measurement quality. Multi-member central shrinkage requires a uniform PCD density for a single member. In other words, the algorithm performs sub-optimally when missing points occur in the members. We note that PCD repair is always required during the data preparation stage, implying the need for additional manual effort.

Module segmentation was conducted manually, and structural configurations and manual tuning had substantial effects. Module segmentation requires the structural PCD to exhibit a repetitive formation of identical structural units, such as a ring-symmetric cable dome. Manual experience determines the number of segmentations required for a structural entity. Proper segmentation is advantageous for better processing efficiency. The number of module segmentations should be optimized.

Conversely, the MASM method utilizes *CloudCompare* and MATLAB software to reconstruct the cable-dome model. The capacity of the specific computational platform is another concern. The PCD size reaches 4×10^8 points, equivalent to a 9 GB storage volume for *CloudCompare*. The MASM algorithm is constrained by MATLAB resources, and this can be assessed using the WHOS function, which estimates the amount of data stored during the analysis. By testing, the maximum storage capacity of MATLAB was determined as 2.88×10^{14}; consequently, the maximum PCD could reach 10^{12}. In addition, owing to the dimensionless calculation procedure, this method has no constraints on the size of the structural members.

The final factor is the accuracy of the MASM method. This study demonstrated that the accuracy of the MASM method reached 1/ 4,334.63 of the structural span. This indicates that the MASM method does not accommodate highly geometry-sensitive monitoring of targets. This method is also not suitable for member-level inspection, where the accuracy is generally within 1 mm [61].

8. Conclusions

A high-fidelity structural model of the cable-dome structure was developed using the proposed MASM method based on 3D laser scanning. The as-measured cable-dome model was used to efficiently and accurately monitor the entire structural deviation. The conclusions are as follows:

1. The MASM method applies module segmentation to a cable-dome PCD entity based on the geometric features of the structural configuration. The additional processing stage reduces the computational complexity more efficiently than direct processing associated with the general method.
2. The MASM method uses two novel algorithms to automatically extract the skeleton and recognize members; these algorithms are the multi-member central shrinkage for skeleton-point recognition and smart multi-

member recognition. The proposed algorithm can accommodate cable domes with various cross-sectional members. The accuracy and efficiency were validated by comparisons with popular algorithms.

3. The MASM method is a systematic structural modeling method that is applicable to general symmetric cable domes with various types and dimensions of members. The reconstruction of the cable-dome model is elaborated from members to modules and eventually to the entity. The accuracy and efficiency of the MASM method were validated.
4. The limitations of the MASM method were discussed in detail. Module segmentation was constrained by manual operations and structural configurations. The best measurement accuracy was 0.45 mm, heavily depending on the PCD density. It was articulated that the computational capacity of the MASM method is constrained by the software used.

This study proposed a technique for reconstructing a structural model from a large volume of PCD. Reconstructed structural models are likely to enable automatic deviation monitoring. In future studies, the established model will be considered for the SDT establishment and structural performance predictions, where numerical simulations will involve high-fidelity SDT models.

Acknowledgment

This study was supported by the Key Program of the National Natural Science Foundation of China (grant number: 52130809).

References

[1] D. Geiger, A. Stefaniuk, D. Chen, The design and construction of two cable domes for the Korean Olympics, Spatial, Lattice and Tension Structures: Proceedings of The International Association for Shell and Spatial Structures (IASS) International Symposium, Osaka 1986. pp. 265-272, Retrieved from http://www.columbia.edu/cu/gsapp/BT/DOMES/SEOUL/s-desig.html (accessed on 13 June, 2024).

[2] M. P. Levy, The Georgia Dome and beyond: achieving Lightweight-Longspan structures, Spatial, Lattice and Tension Structures: Proceedings of The International Association for Shell and Spatial Structures (IASS) International Symposium, Atlanta 1994. pp. 560-562, Retrieved from https://cedb.asce.org/CEDBsearch/record.jsp?dockey=0086962 (accessed on 13 June, 2024).

[3] J. Q. Ge, A. L. Zhang, X. G. Liu, G. J. Zhang, X. B. Ye, S. Wang, X. C. Liu, Analysis of tension form-finding and whole loading process simulation of cable dome structure, Journal of Building Structures. 33 (2012), pp. 1-11, https://doi.org/10.14006/j.jzjgxb.2012.04.002.

[4] A. L. Zhang, G. H. Shangguan, Y. X. Zhang, M. Zou, J. Wang, Experimental study on static performance of fully assembled ridge-tube threading cable with annular-struts cable dome, Engineering Structures. 288 (2023), pp. 116194, https://doi.org/10.1016/j.engstruct.2023.116194.

[5] H. J. Schek, The force density method for form finding and computation of general networks, Computer Methods in Applied Mechanics and Engineering. 3 (1974), pp. 115-134, https://doi.org/10.1016/0045-7825(74)90045-0.

[6] J. Guo, M. Zhu, Negative gaussian curvature cable dome and its feasible prestress design, Journal of Aerospace Engineering. 29(3) (2016), pp. 0893-1321, https://doi.org/10.1061/(ASCE)AS.1943-5525.0000585.

[7] J. K. Zhang, K. Sun, Construction process simulation of cable dome, Applied Mathematics and Mechanics. 94-96 (2011), pp. 750-754, 10.4028/www.scientific.net/AMM.94-96.750.

[8] Z. H. Chen, Y. Li, X. Y. Yan, H. B. Liu, Q. Ma, S. Y. Lou, Experimental research and simulation analysis of a new tension construction forming method for cable dome structure, Spatial Structures. 25(3) (2019), pp. 51-59, https://doi.org/10.13849/j.issn.1006-6578.2019.03.051.

[9] D. A. Gasparini, P. C. Perdikaris, N. Kanj, Dynamic and static behavior of cable dome model, Journal of Structural Engineering. 115(2) (1989), pp. 363-381, https://doi.org/10.1061/(ASCE)0733-9445(1989)115:2(363).

[10] M. L. Zhu, Y. F. Peng, W. N. Ma, J. M. Guo, J. Y. Lu, Artificial neural network-aided force finding of cable dome structures with diverse integral self-stress states-framework and case study, Engineering Structures. 285 (2023), pp. 116004, https://doi.org/10.1016/j.engstruct.2023.116004.

[11] A. L. Zhang, G. H. Shangguan, Y. X. Zhang, M. Zou, C. Y. Luo, Structural behavior of a fully assembled cable-piercing ridge-tube cable dome with modular assembly ring truss, Journal of Constructional Steel Research. 211 (2023), pp. 108193, https://doi.org/10.1016/j.jcsr.2023.108193.

[12] L. M. Zhang, W. J. Chen, S. L. Dong, Manufacture error and its effects on the initial pre-stress of the Geiger cable domes, International Journal of Space Structures. 21(3) (2006), pp. 141-147, https://doi.org/10.1260/026635106779380467.

[13] L. M. Chen, H. Deng, Y. H. Cui, S. L. Dong, Y. Y. Zhou, Theoretical analysis and experimental study on sensitivity of element-length error in cable-strut tensile structures, Advances in Structural Engineering. 19(9) (2016), pp. 1463-1471, https://doi.org/10.1177/1369433216643245.

[14] L. X. Wang, H. B. Liu, Z. H. Chen, F. Zhang, L. L. Guo, Combined digital twin and hierarchical deep learning approach for intelligent damage identification in cable dome structure, Engineering Structures. 274 (2023), pp. 115172, https://doi.org/10.1016/j.engstruct.2022.115172.

[15] L. Sanhudo, N. M. Ramos, J. P. Martins, R. M. Almeida, E. Barreira, M. L. Simoes, V. Cardoso, A framework for in-situ geometric data acquisition using laser scanning for BIM modelling, Journal of Building Engineering. 28 (2020), pp. 1010 73, https://doi.org/10.1016/j.jobe.2019.101073.

669 [16] L. Šiaudinytė, K. T. V. Grattan, Uncertainty evaluation of trigonometric method for vertical angle calibration of the total
670 station instrument, Measurement. 86 (2016), pp. 276-282, https://doi.org/10.1016/j.measurement.2015.10.037.
671 [17] T. D. Moshood, Ja. Rotimi, W. Shahzad, J. A. Bamgbade, Infrastructure digital twin technology: A new paradigm for future
672 construction industry, Technology in Society. 77 (2024), pp. 102519, https://doi.org/10.1016/j.techsoc.2024.102519.
673 [18] A. Sharifi, A. T. Beris, A. S. Javidi, M. S. Nouri, A. G. Lonbar, M. Ahmadi, Application of artificial intelligence in digital
674 twin models for stormwater infrastructure systems in smart cities, Advanced Engineering Informatics. 61 (2024), pp. 102
675 485, https://doi.org/10.1016/j.aei.2024.102485.
676 [19] T. W. Kang, Y. Mo, A comprehensive digital twin framework for building environment monitoring with emphasis on real
677 -time data connectivity and predictability, Developments in the Built Environment. 17 (2024), pp. 100309, https://doi.org/
678 10.1016/j.dibe.2023.100309.
679 [20] R. Chacón, H. Posada, C. Ramonell, M. Jungmann, T. Hartmann, R. Khan, R. Tomar, Digital twinning of building
680 construction processes. Case study: a reinforced concrete cast-in structure, Journal of Building Engineering. 84 (2024), pp.
681 108522, https://doi.org/10.1016/j.jobe.2024.108522.
682 [21] W. M. Kritzinger, M. Karner, G. Traar, J. Henjes, W. Sihn, Digital twin in manufacturing: a categorical literature review
683 and classification, International Federation of Automatic Control (IFAC)-PapersOnLine. 51(11) (2018), pp. 1016-1022,
684 https://doi.org/10.1016/j.ifacol.2018.08.474.
685 [22] B. R. Barricelli, E. Casiraghi, D. Fogli, A survey on digital twin: definitions, characteristics, applications, and design
686 implications, Institute of Electrical and Electronics Engineers (IEEE) Access. 7 (2019) pp. 167653-167671,
687 https://doi.org/10.1109/ACCESS.2019.2953499.
688 [23] Y. M. Ye, Q. Yang, J. G. Zhang, S. H. Meng, J. Wang, A dynamic data driven reliability prognosis method for structural
689 digital twin and experimental validation, Reliability Engineering & System Safety. 240 (2023), pp. 109543, https://doi.org
690 /10.1016/j.ress.2023.109543.
691 [24] V. Vanessa, J. Handibry, F. Henry, Technologies for digital twin applications in construction, Automation in Constructio
692 n. 152 (2023), pp. 104931, https://doi.org/10.1016/j.autcon.2023.104931.
693 [25] T. Zhou, S. M. Hasheminasab, A. Habib, Tightly-coupled camera/LiDAR integration for point cloud generation from
694 GNSS/INS-assisted UAV mapping systems, International Society for Photogrammetry and Remote Sensing (ISPRS)
695 Journal of Photogrammetry and Remote Sensing. 180 (2021), pp. 336-356, https://doi.org/10.1016/j.isprsjprs.2021.08.020.
696 [26] C. P. Chao, C. Y. Wang. Design of constrained dynamic path planning algorithms in large-scale 3D point cloud maps for
697 UAVs, Journal of Computational Science. 67 (2023), pp. 101944, https://doi.org/10.1016/j.jocs.2023.101944.
698 [27] Z. H. Huang, D. Li. A 3D reconstruction method based on one-dimensional galvanometer laser scanning system, Optics
699 and Lasers in Engineering. 170 (2023), pp. 107787, https://doi.org/10.1016/j.optlaseng.2023.107787.

[28] T. Srimontriphakdi, P. Mahasuwanchai, C. Athisakul, S. Leelataviwat, K. Klaycham, N. Poovarodom, N. Magteppong, S. Chucheepsakul, A non-contact approach for cable tension evaluation based on 3D laser scanning data and nonlinear finite element analysis, Measurement. 212 (2023), pp. 112680, https://doi.org/10.1016/j.measurement.2023.112680.

[29] D. B. Zhang, T. Q. Liu, S. H. Li, H. Q. Luo, G. Z. Ding, Z. H. Su, C. F. Xie, S. J. Zhang, P. Feng, Three-dimensional laser scanning for large-scale as-built surveying of 2022 Beijing Winter Olympic Speed Skating Stadium: a case study, Journal of Building Engineering. 59 (2022), pp. 105075, https://doi.org/10.1016/j.jobe.2022.105075.

[30] X. X. Li, E. Deng, Y. W. Wang, Y. Q. Ni, 3D laser scanning for predicting the alignment of large-span segmental precast assembled concrete cable-stayed bridges, Automation in Construction. 155 (2023), pp. 105056, https://doi.org/10.1016/j.autcon.2023.105056.

[31] X. C. Wei, J. S. Fan, Y. F. Liu, J. X. Zhang, X. G. Liu, S. Y. Kong, Automated inspection and monitoring of member deformation in grid structures, Computer-Aided Civil and Infrastructure Engineering. 37(10) (2021), pp. 1277-1297, https://doi.org/10.1111/mice.12766.

[32] Bentley, URL: https://www.bentley.com/software/bentley-descartes/ (accessed on 13 June, 2024).

[33] Trimble, URL: https://cn.geospatial.trimble.com/products-and-solutions/trimble-realworks (accessed on 13 June, 2024).

[34] Leica, URL: https://www.leica-geosystems.com.cn/ (accessed on 13 June, 2024).

[35] Z. L. Ma, S. L. Liu, A review of 3D reconstruction techniques in civil engineering and their applications, Advanced Engineering Informatics. 37 (2018), pp. 163-174, https://doi.org/10.1016/j.aei.2018.05.005.

[36] I. Heinz, F. Härtl, C. Fröhlich, Semi-automatic 3D CAD model generation of as-built conditions of real environments using a visual laser radar, Proceedings 10th Institute of Electrical and Electronics Engineers (IEEE) International Workshop on Robot and Human Interactive Communication, Roman 2001. pp. 400-406, https://doi.org/10.1109/ROMAN.2001.981937.

[37] F. Bosche, C. T. Haas, Automated retrieval of 3D CAD model objects in construction range images, Automation in Construction. 17(4) (2008), pp. 499-512, https://doi.org/10.1016/j.autcon.2007.09.001.

[38] D. S. Li, J. P. Liu, L. Feng, Y. Zhou, H. T. Qi, Y. F. Chen, Automatic modeling of prefabricated components with laser-scanned data for virtual trial assembly, Computer-Aided Civil and Infrastructure Engineering. 36(4) (2020), pp. 453-471, https://doi.org/10.1111/mice.12627.

[39] J. Liu, L. Fu, G. Cheng, Automated BIM reconstruction of full-scale complex tubular engineering structures using terrestrial laser scanning, Remote Sensing. 14(7) (2022), pp. 1659, https://doi.org/10.3390/rs14071659.

[40] Y. H. Jin, W. H. Lee, Fast cylinder shape matching using random sample consensus in large scale point cloud, Applied Sciences. 9(5) (2019), pp. 974, https://doi.org/10.3390/app9050974.

[41] M. N. Xu, L. M. Sun, Y. F. Liu, B. L. Li, M. Zhou, X, Chen, Member separation and deformation recognition of spatial grid structures in-service, Engineering Structures. 304 (2024), pp. 117642, https://doi.org/10.1016/j.engstruct.2024.117642.

[42] J. Lee, H. Son, C. Kim, C. Kim, Skeleton-based 3D reconstruction of as-built pipelines from laser-scan data, Automation in Construction. 35 (2013), pp. 199-207, https://doi.org/10.1016/j.autcon.2013.05.009.

[43] G. Castellazzi, C. Carloni, F. Ubertini, An innovative numerical modeling strategy for the structural analysis of historical monumental buildings, Engineering Structures. 132 (2017), pp. 229-248, https://doi.org/10.1016/j.engstruct.2016.11.032.

[44] A. Tagliasacchi, H. Zhang, D. Cohen-Or, Curve skeleton extraction from incomplete point cloud, Association for Computing Machinery transactions on graphics. 28(3) (2009), pp. 1-9, https://doi.org/10.1145/1576246.1531377.

[45] H. Huang, S. H. Wu, D. Cohen-Or, M. L. Gong, H. Zhang, G. Q. Li, B. Q. Chen, L1-medial skeleton of point cloud, Association for Computing Machinery (ACM) Transactions on Graphics. 32(4) (2013), pp. 1-8, https://doi.org/10.1145/2461912.2461913.

[46] Y. F. Liu, S. Cho, B. F. Spencer Jr., F. ASCE, J. S. Fan, Concrete crack assessment using digital image processing and 3D scene reconstruction, Journal of Computing in Civil Engineering. 30(1) (2016), pp. 04014124, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000446.

[47] E. Ozbay, A. Cinar, Z. Guler, A hybrid method for skeleton extraction on Kinect sensor data: combination of L1-median and Laplacian shrinking algorithms, Measurement. 125 (2018), pp. 535-544, https://doi.org/10.1016/j.measurement.2018.05.029.

[48] Y. Q. Liu, Justin K.W. Yeoh, Automated crack pattern recognition from images for condition assessment of concrete structures, Automation in Construction. 128 (2021), pp. 103765, https://doi.org/10.1016/j.autcon.2021.103765.

[49] Y. J. Yan, J. F. Hajjar, Automated extraction of structural elements in steel girder bridges from laser point clouds, Automation in Construction. 125 (2021), pp. 103582, https://doi.org/10.1016/j.autcon.2021.103582.

[50] P. Tang, D. Huber, B. Akinci, R. Lipman, A. Lytle, Automatic reconstruction of as-built building information models from laser-scanned point clouds: a review of related techniques, Automation in Construction. 19(7) (2010), pp. 829-843, https://doi.org/10.1016/j.autcon.2010.06.007.

[51] A.V. Vo, L. Truong-Hong, D.F. Laefer, M. Bertolotto, Octree-based region growing for point cloud segmentation, International Society for Photogrammetry and Remote Sensing (ISPRS) Journal of Photogrammetry and Remote Sensing. 104 (2015), pp. 88-100, https://doi.org/10.1016/j.isprsjprs.2015.01.011.

[52] RIGEL-VZ1000, URL: http://www.ilidar.com/product/3.html (accessed on 13 June, 2024).

[53] H. Ma, A. L. Zhang, X. Zhao, Y. X. Zhang, G. Du, Research on digital twin technology for large-span steel structures based on 3D laser scanning, 13th Pacific Structural Steel Conference, Chengdu 2023. pp. 102, Retrieved from https://www.researchgate.net/publication/381403230 (accessed on 13 June, 2024).

[54] RiSCAN PRO 2.0, URL: http://www.riegl.com/products/software-packages/riscan-pro/ (accessed on 13 June, 2024).

[55] M. Ester, H. P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland Oregon 1996. pp. 226-231, https://dl.acm.org/doi/10.5555/3001460.3001507.

[56] L. A. Shalabi, Z. Shaaban, B. Kasasbeh, Data mining: a preprocessing engine, Journal of Computational Science. 62(4) (2006), pp.735-739, https://www.researchgate.net/publication/26445674.

[57] N. Fumio, H. Wibisono, Influential mode of imperfection on carrying capacity of structures, Journal of Engineering Mechanics. 115 (1989), pp. 2150-2165, https://doi.org/10.1061/(ASCE)0733-9399(1989)115:10(2150).

[58] Ansys, URL: http://www.ansys.com/ (accessed on 13 June, 2024).

[59] Geomagic Wrap, URL: http://www.geomagic.mairuan.com/ (accessed on 13 June, 2024).

[60] C. W. Tian, L. K. Fei, W. X. Zheng, Y. Xu, W. M. Zuo, C. W. Lin, Deep learning on image denoising: an overview, Neural Networks. 131 (2020) pp. 251-275, https://doi.org/10.1016/j.neunet.2020.07.025.

[61] Technical specification for space frame structures, China Architecture &Building Press, 2010, pp. 56, Retrieved from https://www.nssi.org.cn/nssi/front/77628516.html (accessed on 13 June, 2024).