Automated surface damage detection and quantification in concrete structures using semantic point clouds

Linjie Huang [a,b,c], Gao Fan [a,b,c,*], Jun Li [d,**], Shihong Chen [a,b,c], Zhen Peng [d], Hong Hao [a,d]

[a] Earthquake Engineering Research & Test Center, Guangzhou University, Guangzhou 510006, China
[b] Key Laboratory of Earthquake Resistance, Earthquake Mitigation and Structural Safety, Ministry of Education, Guangzhou University, Guangzhou 510006, China
[c] Guangdong Provincial Key Laboratory of Earthquake Engineering and Applied Technology, Guangzhou University, Guangzhou 510006, China
[d] Center of Infrastructural Monitoring and Protection, School of Civil and Mechanical Engineering, Curtin University, Bentley 6102, Australia

ABSTRACT

Manual inspection of civil infrastructure is increasingly insufficient to meet the growing demands for accuracy and scalability. This paper integrates UAV image segmentation, 3D reconstruction and point cloud processing techniques to propose an automated approach for detecting and quantifying multiclass concrete surface damage. Two innovative methods based on semantic point clouds are sophisticated developed, include a spherical region-based crack width quantification method and a watertight surface mesh based method for determining the area and volume of surface and block damage. Experimental validation on a bridge pier demonstrates the effectiveness and robustness of the proposed approach, achieving precise quantification of a fine and intricate crack, with an average width estimation error of 0.09 mm and a corresponding standard deviation of 0.046 mm. Moreover, the approach surpasses existing techniques by producing high-resolution watertight surface meshes, allowing accurate damage area and volume quantification with average relative errors within 1 % and 3.5 %, respectively, and absolute error within 2.5 cm^2 and 1.5 cm^3, respectively. This innovative approach enhances 3D visualization and perception of structural conditions while providing precise quantitative damage information, significantly improving the efficiency and reliability of structural inspection.

1. Introduction

Effective evaluation of damage conditions in civil infrastructures is crucial for efficient management and maintenance. For decades, regular manual inspection has been the primary on-site method to obtain surface damage information on concrete structures. Nevertheless, this method exhibits significant limitations, including subjective judgment, being time-consuming, requiring substantial manual labor, and entails high risks. Recently, with the rapid development of artificial intelligence (AI) and unmanned aerial vehicle (UAV) photogrammetry, surface inspection of civil infrastructure is gradually evolving towards automation and intelligence.

UAV inspection of civil infrastructures typically generates a large number of images. To process these data efficiently, the development of automated surface damage detection approaches primarily relies on advanced image processing techniques. With advances in computer science, deep learning (DL) has gained rapid development in image detection and is widely investigated for structural inspection tasks in civil engineering. Convolutional Neural Networks (CNNs), featuring local perception and weight sharing, were first employed to determine the existence of cracks in images [1]. For more advanced detection requirement, object detection algorithms featuring bounding boxes generation were used to locate structural damage on input images [2,3]. Besides, end-to-end semantic segmentation to annotate damage in pixel-level have been conducted by using DL models, including FCN [4, 5], U-net [6,7], and Mask R-CNN [8,9]. Recently, Transformer [10,11] architecture was introduced in the designed framework for detecting damages [12,13], showcasing its potential in distinguishing damage from complex backgrounds.

Except effective detection of surface damage, the quantitative assessment of detected damage is also of great significance for understanding the structural health conditions and providing

recommendations for strengthening and repair. Several researches have made efforts to estimate geometrical parameters of cracks from images, including length, width, and orientation [7,14–17]. Miao et al. [14] determined crack width by fitting local skeleton lines using neighboring points of both two crack edges. A perpendicular line of the skeleton line thus was determined, and the distance between its intersections with the crack edges was taken as the crack width. This method smoothed local irregularities but relied on predefined neighborhood parameters. Wang et al. [15] extracted crack skeletons and edges using image processing techniques. Crack width was calculated by projecting normal rays from skeleton points and measuring the distance between their intersections with the crack contours. In most existing approaches, crack width is typically measured as the distance between the pixels where a normal line extending from the crack skeleton intersects its two edge lines. For surface and block damage, Xue et al. [18] developed an inspection approach to detect and quantify water leakage areas in tunnels with varying shapes and environment conditions. Li et al. [19] proposed an quantification approach to measure the actual areas of common concrete surface damage induced by crack, spalling, efflorescence, and hole. While image-based damage quantification approaches can effectively quantify damage from 2D images, they rely on pixel-based measurements and require camera calibration to convert results into real-world units. This process introduces additional sources of error, especially in field environments where lighting, resolution, and shooting distance vary, and becomes increasingly labor-intensive when processing large volumes of images. Moreover, 2D approaches implicitly assume the damage lies on a planar surface, which is often not the case in real-world concrete structures that may exhibit curvature, tilt, or uneven surfaces. These inherent limitations present challenges and inefficiencies in delivering a full-scale and comprehensive assessment of structural heath condition [20,21].

On the other hand, conducting damage detection and quantification from a 3D perspective holds significant promising in addressing the limitations of 2D image-based approaches. Damage detection using LiDAR (Light Detection and Ranging) collected point cloud data has become the mainstream solution in early 3D structure inspections. Point cloud data can provide an efficient and economic 3D representation of multiple types information, such as coordinates, RGB, normal and intensity [22]. Hou et al. [23] applied clustering algorithms to process the intensity and RGB information from LiDAR point clouds to identify various types of textural damage, including tile spalling, metal rusting, and water staining. Kim et al. [24] localized concrete spalls by analyzing the orientation shift of point normal vectors, rendering accuracy in detecting spalls with both depth and width greater than 3 mm. For point cloud-based damage quantification, Erkal and Hajjar [25] quantified cracks in a partially collapsed concrete frame by combining the color and surface normal information, demonstrating its usefulness in estimating crack thickness and length. The results remain dependent on the neighborhood size and threshold selections for normal estimation. Olsen et al. [26] and Kim et al. [27] proposed slice based approaches, which simplify the volume quantification into the sum of the loss of cross-section slices of structure surface, to quantify spalls within the concrete structures with non-flat surfaces. These approaches demonstrate effectiveness in identifying and quantifying concrete surface damage, and show more intuitive presentation of damage than those approaches fully based on images. However, these studies extract points of damage by using handcrafted damage sensitive characteristics based on the inherent properties (e.g., color, normal, and intensity) of point cloud, leading to low detecting efficiency and robustness. Besides, scanning of large structures by LiDAR sensors is time-consuming and terrain-dependent, posing a challenge to its application in large civil infrastructures.

With the development of UAV photography and multi-view 3D reconstruction techniques, producing point cloud of civil infrastructures based on high-quality image sequences captured by UAV equipped camera become a potential economic and flexible alternative to LiDAR [28,29]. Attributed to the flexible nature of UAV image collection and thorough flight path planning, photogrammetric point cloud contains more realistic color and texture than LiDAR point cloud [30,31], providing more detectable details and facilitating its application in concrete surface inspection [32–34]. Kim et al. [35] used the photogrammetric point cloud of a bridge structure to generate a background model for locating and visualizing the image patches with identified cracks, establishing a global mapping for inspection. Khaloo et al. [33] manually inspected damage in a timber truss bridge by using a dense Structure from Motion (SfM) [36] model generated from UAV-captured images.

The photogrammetric point cloud also demonstrates considerable potential in detecting and quantifying damage owing to its dense and realistic surface. Moreover, in contrast to manually identifying damage points from LiDAR point cloud by handcrafted thresholds, it can exploit advanced deep learning based image detection techniques to identify local damage and seamlessly and precisely integrate them into 3D models [16,17,37–40]. Chaiyasarn et al. [39] implemented patch-level crack identification on mosaic images, then mapping the processed images onto a reconstructed mesh model created by SfM to highlight the crack regions in a building. Kalfarisi et al. [40] introduced a framework for crack segmentation and quantification, wherein detected cracks were integrated into a 3D reality mesh model to provide an intuitive and holistic visualization of structural damage. Similarly, Liu et al. [16] introduced an approach that detects cracks and estimates their widths on non-planar concrete surfaces using images, with a 3D triangular mesh model built to precisely localize the detected cracks. In these studies, damage quantification is still predominantly performed at the 2D image level, with 3D models serving to localize and highlight damage regions rather than quantify dimensions on 3D points. As an alternative, some researchers suggest interpolating damage point colors from damage-annotated images to produce a semantic point cloud [21, 41–43]. Zeng et al. [41] proposed a systematic approach encompassing concrete crack detection and localization by incorporating the deep learning based semantic segmentation and 3D reconstruction. The approach realizes precise and efficient determination of crack presence, shape, and location of in a concrete slab through the use of semantic point cloud. Researches using 3D points from semantic point cloud to quantify concrete damage has shown significant potential in reducing errors caused by geometric and perspective distortion, making such approaches more adaptable to damage distributed across multiple surfaces. Deng et al. [21] developed an semantic point cloud based approach for automatic crack length estimation. Binocular camera data were utilized for crack segmentation and Video Simultaneous Localization and Mapping (VSLAM) reconstruction. This approach determines the crack location in overall structure and calculates crack lengths with a mean relative error of 2.97 %. Yuan et al. [43] developed an intelligent framework for concrete crack quantification using an unmanned vehicle-mounted stereo camera system, which employs Convex Hull algorithms to estimate crack volumes within the semantic point cloud. These stereo vision–based approaches rely on fixed baseline distances and have limited fields of view, making them less adaptable to terrain-constrained environments. The aforementioned studies confirm that semantic point cloud–based approaches can overcome the limitations of using image- or point cloud–based techniques alone, enabling more advanced and efficient damage detection and quantification. However, this research direction is still in its early development stage and remains limited in scope and depth: (1) UAVs offer greater positioning flexibility and lower data acquisition costs, but semantic point cloud-based damage quantification using UAV data remains relatively underexplored. (2) There is still a lack of benchmark methodologies for 3D crack width measurement and high-precision area and volume quantification, which hinders the advancement of comprehensive damage quantification technologies.

This paper proposes a semantic point cloud based approach for automatically detecting and quantifying multiclass concrete surface

damage under complex surfaces and contours. The automated approach incorporates UAV photogrammetry data acquisition, DL-based damage segmentation, semantic point cloud generation and point cloud processing for damage detection and quantification. Leveraging semantic point clouds derived from UAV images and 3D reconstruction, the study introduces two key innovations: a spherical region-based crack width quantification method and a watertight surface mesh based method for quantifying the area and volume of surface and block damage. The following sections of this paper are organized as: Section 2 elaborates the methodologies of the developed damage detection and quantification approach based on semantic point clouds. Section 3 introduces the experimental setup and implementation on a scaled concrete bridge pier with various surface damage. Section 4 comprehensively discusses experimental results, validating the effectiveness, robustness, and potential for engineering applications of the approach. Section 5 concludes with the main findings and limitations.

2. Methodology

2.1. Overall framework

The proposed approach for concrete surface damage detection and quantification consists of four procedures: UAV photogrammetry-based data acquisition, DL-based multiclass damage segmentation, semantic point cloud reconstruction, and multiclass damage quantification. As presented in Fig. 1, UAV photogrammetry collects structure surface images and corresponding camera GPS coordinates from multiple views and varying distances along a planned path. The collected images are processed by a trained DL model for multiclass damage segmentation, which provides pixel level damage annotations. Using the original images and their corresponding semantic images, a real-sized semantic point cloud of the target structure can be established by 3D reconstruction algorithms, including SfM [36], scale recovery and Multi-view Stereo (MVS) [44]. This real-sized semantic point cloud offers the global location of damage and point-wised damage segmentation. With this semantic point cloud, the characteristics of damage instances can be quantified by using the proposed damage quantification methods based on point cloud processing algorithms. Notably, the workflow of the proposed approach is fully automatic, integrating image and point cloud processing metric to detect and quantify surface damage, thereby providing reliable detection results from a 2D perspective and global localization and quantification results from a 3D perspective.

2.2. UAV photogrammetry-based data acquisition

UAV photogrammetry is an effective solution for acquiring high-resolution image sequences for structural inspection and 3D reconstruction through close-range shooting. Research on image-based 3D reconstruction and damage detection demonstrates the importance of the UAV flight strategies [16,45]. Flight parameters, such as shooting distance and image overlap, critically influence the quality of UAV images and ensure consistent scale across the image sequence. To optimize image resolution while maintaining UAV flight safety, the shooting distance should exceed 1.00 m and be smallest possible relative to the target surface [16]. Additionally, an image overlap of more than 50 % is essential for accurate 3D reconstruction [16]. Perpendicular shooting is recommended to be maintained to minimize image distortion. Therefore, this paper identifies shooting distance and image overlap as key factors that must be carefully controlled during UAV image acquisition.

2.3. DL-based image segmentation

2.3.1. SegFormer for damage semantic segmentation

Following the UAV image acquisition, DL-based damage segmentation is introduced to extract damage regions at the pixel level, providing semantic images for semantic point cloud reconstruction. Since the semantic segmentation model must identify multiple types of damage from complex backgrounds without cropping of inspection images, it requires a robust capability to processing global contextual information. To this end, SegFormer [46], as illustrated in Fig. 2, is employed. SegFormer is a simple and efficient Transformer-based network [11], featuring a hierarchical Transformer encoder to generate multi-scale feature maps, and a lightweight Multi-Layer Perceptron (MLP) based decoder to detect damage pixels and annotates various types of damage with distinct colors.

2.3.2. Dataset preparation

Deep learning-based damage semantic segmentation serves as a key step in constructing the semantic point cloud of a target structure, primary aimed at enhancing the automation of damage annotation in UAV inspection images. To support this, a customized training dataset is constructed, focusing on representative rather than comprehensive or large-scale samples. It consists of a subset of UAV images captured from various angles and distances, requiring only limited manual annotation while enabling efficient training, thereby significantly reducing the overall annotation workload.
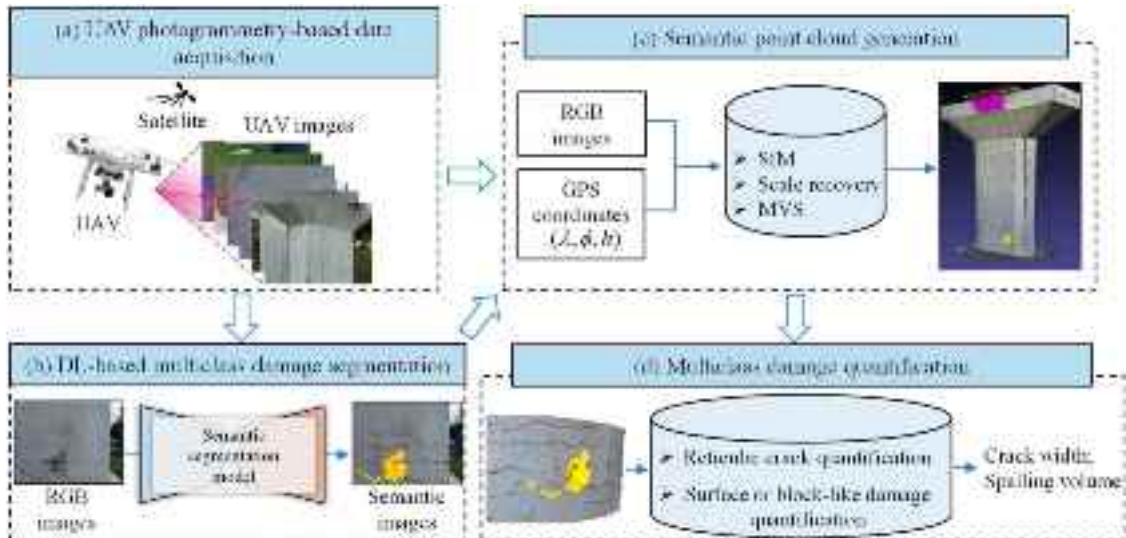

Fig. 1. Overall framework of the proposed damage detection and quantification approach.
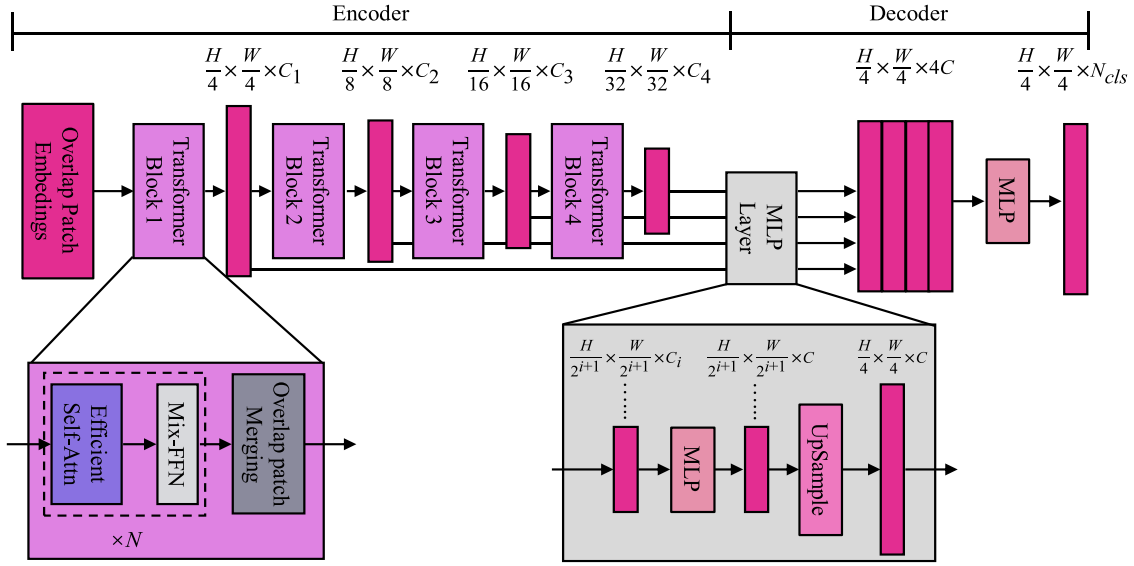
Fig. 2. Structure of SegFormer.

Three of the most commonly observed damage types, including cracks, spalling, and corrosion, are selected as the target categories for semantic segmentation. Ground-truth labels for these categories are manually created using the widely adopted LabelMe annotation tool. Each label is saved as a single-channel grayscale image, in which different grayscale values are assigned to distinguish between damage types: cracks are labeled with a grayscale value of 1, spalling with 2, and corrosion with 3. To facilitate visual interpretation, these grayscale values are further mapped to predefined RGB colors. The RGB color scheme of each defect category is shown in Table 1. These RGB values are also used during the prediction phase to visualize the identified damage regions in the output images, allowing for intuitive differentiation among damage categories.

The dataset is divided into training, validation, and test subsets according to an 8:1:1 ratio. Given the limited number of UAV images, care is taken to maintain uniform viewpoint distribution across subsets. To mitigate potential overfitting during training, data augmentation is employed to rapidly increase training samples. This study applies four common data augmentation techniques: random flipping, random rotation, brightness adjustment, and Gaussian blur, expanding the number of training samples by approximately four times. Details of the applied augmentation constraints are listed clearly in Table 2. To ensure that the augmented images maintain similarity with the original UAV imagery, augmentation is applied with specific constraints, outlined in Table 2. Upon the data augmentation, the training samples are increased by 4 times.

2.3.3. Training configuration

To ensure that the models effectively learned both structural and damage-specific features, all training images are uniformly scaled to a resolution that can be accommodated by the available GPU memory without the need for cropping. This approach preserves the full spatial and contextual information within each image, which is essential for accurate semantic segmentation in structural inspection scenarios. The training process is conducted for a maximum of 200 epochs using a

batch size of 2 images. An initial learning rate of 0.00006 is adopted, and the AdamW optimizer is selected due to its effectiveness in handling weight decay, which is set at 0.01. Cross-Entropy loss function is chosen to supervise training of the damage segmentation model, expressed as

$$L(y, \widehat{y}) = -\frac{1}{N}\sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{C} y_{ij} \cdot \log(\widehat{y}_{ij}) \qquad (1)$$

where N is the number of instances; C is the number of categories; y_i is a one-hot encoded vector representing the ground truth for the i-th instance; $\widehat{y}_i$ is the prediction over categories for the i-th instance. Throughout training, both Loss and Accuracy are systematically recorded every two epochs for the training and validation datasets to monitor model convergence and performance stability. Model performance is quantitatively evaluated using the Intersection-over-Union (IoU) metric, a widely accepted standard for semantic segmentation tasks, which allows for reliable assessment of segmentation accuracy across multiple damage classes.

2.4. Real-size semantic point clouds generation

This section introduces the method for reconstructing real-size semantic point clouds of concrete structures. As shown in Fig. 3, this method incorporates 3D reconstruction algorithms (SfM, scale recovery and MVS) with DL based damage segmentation (explained in Section 2.3). The method relies solely on UAV-captured RGB images and their associated GPS coordinates, which are readily obtainable. The RGB images serve as input for both 3D reconstruction and DL based damage segmentation. The DL model performs pixel-wise classification and assigns class-specific RGB values to crack, spalling, and corrosion areas, resulting in semantic images for use in subsequent 3D reconstruction. It is essential that the colors of damage pixels are clearly distinguishable from other classes and the background. The color scheme used for damage type annotation is defined in Section 2.3.

The 3D reconstruction in this paper is based on the pinhole camera

Table 1
RGB scheme for annotating various damage.

RGB	Damage type		
	Crack	Spalling	Corrosion
[R, G, B]	[128, 0, 0]	[255, 195, 0]	[192, 0, 192]

Table 2
Parameters for training data augmentation.

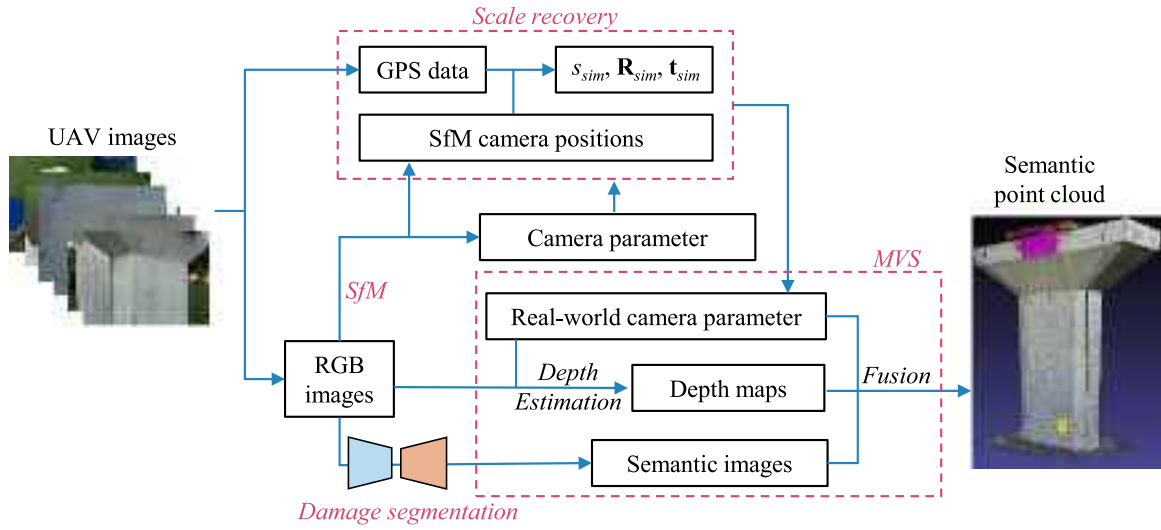| Augmentation methods	Thresholds
Horizontal flipping	Probability: 0.5
Rotation	Degrees: [−20, 20]
Gamma transform	Gamma value: [0.8, 1.2]
Gaussian blur	Blur factor: [0.5, 2]

Fig. 3. Workflow of real-size semantic point cloud generation.

model, which employs camera parameters to project a point $p(x_w, y_w, z_w)$ in 3D world coordinates onto a pixel location $p(u, v)$ in an image, as expressed in

$$d(u,v)\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R^{3\times3} & T^{3\times1} \\ 0^{1\times3} & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = K^{3\times3}E^{3\times4}\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2)$$

where $K^{3\times3}$ represents the camera intrinsic matrix, composed of focal lengths f and the principal point (u_0, v_0); $E^{3\times4}$ is a camera extrinsic matrix, assembled from rotational $R^{3\times3}$ and translational $T^{3\times1}$ matrices; The term $d(u, v)$ denotes the depth of $p(u, v)$, which is the perpendicular distance from the center of camera lens to the point $p(x_w, y_w, z_w)$.

The SfM algorithm, which includes feature extraction and matching, camera parameter estimation, triangulation, and bundle adjustment, is first operated to downsampled RGB images to estimate camera parameters and reconstruct a sparse 3D point cloud. However, the SfM outputs exist in an arbitrary local coordinate system, with unknown measurement unit and scale. Therefore, a scale recovery process is required to transform the outputs into a real-world coordinate system. To achieve this, the UAV recorded GPS locations are matched with the corresponding SfM-estimated camera centers. These matched point pairs are used to compute a similarity transformation, which includes a scale factor s, rotation matrix R, and translation vector t. This transformation is expressed as

$$\mathbf{p}_i^{GPS} \approx s\mathbf{R}\mathbf{p}_i^{SfM} + \mathbf{t}, \ i = 1, ..., N. \quad (3)$$

where: $\mathbf{p}_i^{SfM}$ is the SfM-estimated camera positions, while $\mathbf{p}_i^{GPS}$ denotes the corresponding GPS locations. This transformation is solved using the Umeyama algorithm [47], which minimizes the sum of squared distances between the GPS positions and their corresponding SfM-estimated camera centers and can be expressed as

$$\min_{s,\mathbf{R},\mathbf{t}} \sum_{i=1}^{N} \left\| s\mathbf{R}\mathbf{p}_i^{SfM} + \mathbf{t} - \mathbf{p}_i^{GPS} \right\|^2 \quad (4)$$

Once the similarity transformation parameters (s_{sim}, $\mathbf{R}_{sim}$, $\mathbf{t}_{sim}$) are obtained, all camera poses and 3D points from SfM can be aligned to real-world coordinates. Among these, the transformation of camera poses is more critical and indispensable for downstream MVS tasks, which is expressed as

$$\mathbf{T}_{Real-world} = \begin{bmatrix} \mathbf{R}_{sim}\mathbf{R}_{SfM} & s_{sim}\mathbf{R}_{sim}\mathbf{t}_{SfM} + \mathbf{t}_{sim} \\ 0^{1\times3} & 1 \end{bmatrix} \quad (5)$$

where $\mathbf{T}_{Real-world}$ represents the camera parameters in the real-world coordinate system, while $\mathbf{R}_{SfM}$ and $\mathbf{t}_{SfM}$ represent SfM-estimated rotation matrix and translation vector, respectively.

Following the scale recovery, the MVS algorithm [44] uses the real-world camera parameters, RGB images, and semantic images to perform depth estimation and fusion, ultimately producing a dense, real-size semantic point cloud. In the depth estimation stage, RGB images and real-world camera parameters are used to estimate depth maps from multiple views to obtain accurate and consistent depth information. In the subsequent fusion stage, the algorithm leverages the projection relationships among views and the estimated depth maps to convert semantic image pixels into 3D point coordinates, effectively mapping annotated damage into 3D space. This process enables the integration of multi-view depth and semantic information into a unified coordinate system, producing a dense semantic point cloud with realistic surface texture and spatially accurate annotated damage. Notably, this is achieved without the need for physical targets or manual 3D annotation, significantly improving efficiency and automation. To prepare the clear semantic point cloud for damage quantification, irrelevant background information, such as trees, roads, and adjacent buildings reconstructed alongside the target structure, is manually removed from the 3D model.

2.5. Multiclass damage quantification methods

Though the damage instances are annotated in semantic point clouds, the points within a point cloud are unorganized and unordered, necessitating adequate point cloud processing for damage quantification. In this Section, advanced automated damage quantification methods, which effectively integrate various point cloud processing algorithms, are developed to compute physical sizes of damage with complex contours and surfaces.

2.5.1. Damage instance extraction

To derive the quantification parameters of each individual damage instance, damage points are extracted from the semantic point cloud and categorized into distinct damage classes and instances. The key procedures include extracting damage points of each class based on RGB color intervals, detecting and filtering outliers, and clustering damage instances, as shown in Fig. 4. The RGB color of each 3D point is generated through a weighted averaging process of corresponding pixel values from multiple views, as implemented in MVS. The weights are influenced by several factors, including the angle between the camera's optical axis and the 3D point's normal, and the local depth estimation errors. Viewpoints that are closer to the normal direction typically
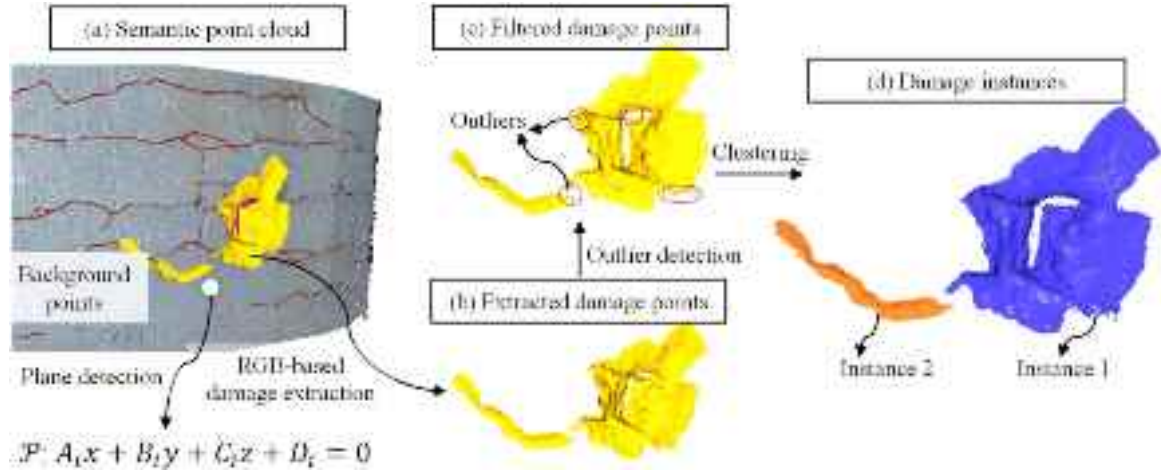
Fig. 4. Procedures of damage instance extraction.

receive larger weights, whereas viewpoints with significant depth errors are suppressed. Given this, the color of damage points may slightly differ from the RGB values in the semantic images, and the final 3D point color cannot be accurately determined by a simple arithmetic mean of the corresponding pixel values. Therefore, damage points for each class are sifted and grouped into a point set *DP* based on empirically determined RGB color intervals, rather than strict analytical derivation. This process can be expressed as follows

$$DP = \{x | x \in PC_{sem}, R_1 \leq R(x) \leq R_2, G_1 \leq G(x) \leq G_2, B_1 \leq B(x) \leq B_2\} \tag{6}$$

where PC_{sem} is the data set of semantic point cloud; $R(x)$, $G(x)$, and $B(x)$ are functions that return the red, green, and blue color values of the point x, respectively; (R_1, R_2), (G_1, G_2), and (B_1, B_2) are the threshold values of specific intervals for red, green, and blue channels, respectively. To improve robustness, each damage class is assigned a clearly distinguishable color, with broad RGB ranges defined to ensure reliable point classification. These intervals are experimentally validated to ensure insensitivity to minor color blending or illumination variation during 3D reconstruction, as detailed in the Table 3.

The extracted damage points often include outliers that can affect the accuracy of the subsequent quantitative estimations. To address this, the Statistical Outlier Removal (SOR) algorithm from the Open3D library [48] is employed to detect and remove these outliers. This algorithm evaluates the local density around each point and removes those that deviate significantly from their neighbors. Two parameters control this process: (a) the number of neighbors, and (b) the standard deviation threshold. Identifying and filtering noise in damage point clouds is inherently challenging and somewhat subjective, particularly for cracks,

as the density of damage points varies significantly across different regions. In this study, relatively conservative thresholds are used preserve more data for subsequent quantification while still effectively removing isolated and clearly erroneous points, rather than establishing a universal threshold applicable to all cases. The parameter values used for this denoising process are provided in Table 3.

Afterward, the filtered damage points can be organized into distinct damage instances through clustering, considering their spatial relationships within a 3D coordinate system. In this study, Ordering Points to Identify the Clustering Structure (OPTICS) algorithm [49] is employed to cluster the damage points into multiple 3D damage instances. This density-based clustering algorithm effectively identifies clusters of varying densities and shapes without needing to specify the number of clusters in advance. Parameters including minimum neighbors, maximum distance, and minimum samples are adjusted moderately, as detailed in Table 3. As a result, it enables damage instance extraction with minimal manual intervention, facilitating effective damage quantification.

Spatial information is also crucial for quantifying damage instances. Random Sample Consensus (RANSAC) [50] algorithm is used to fit reference planes based on the flat concrete surface where the damage instances are located, as expressed in

$$\mathscr{P} = \{\mathscr{P}_i = \{(x, y, z) \in \mathbb{R}^3 | A_i x + B_i y + C_i z + D_i = 0\} | i = 1, \ldots, n\} \tag{7}$$

The use of $\mathscr{P}$ provides essential orientation and position information for quantifying damage instance, facilitating the identification of crack width points and the generation of a watertight surface mesh.

2.5.2. Crack width quantification method

Cracks are among the most common types of damage in concrete structures, severely threatening structural safety. Most crack quantification methods have concentrated on simple, well-defined line-shape cracks. However, complex and reticular cracks, characterized by their intricate network of branches and intersections, frequently occur in actual inspections and pose substantial challenges in recognizing and assessing individual crack instances. This paper proposes a spherical region-based width quantification method for evaluating complex crack instances, effectively integrating skeleton points, point pairs, and boundary points, as shown in Fig. 5.

For quantifying crack width, it is crucial to determine the center coordinates and boundary points of a crack. This study employs Laplacian-Based Contraction (LBC) algorithm [51] to generate skeleton points at center of a crack instance, allowing estimation of the width at each skeleton point. Although the spatial coordinates of skeleton points are known after extraction, their connectivity remains undefined, especially in the case of complex cracks with branching or intersections,

Table 3
Parameters and thresholds for damage instance extraction.

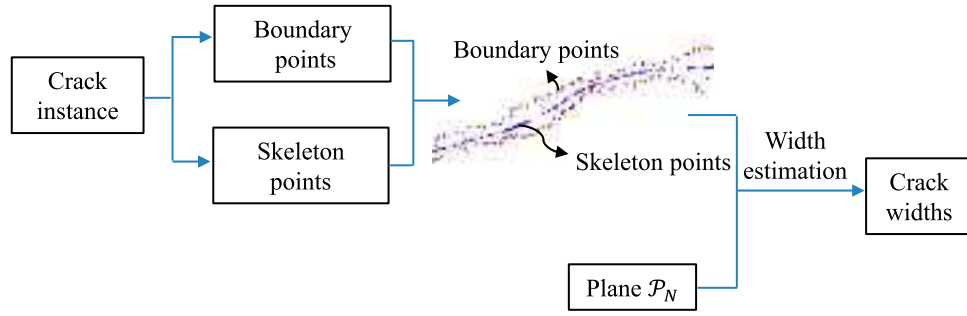| Method	Parameter	Threshold		
		Crack	Spalling	Corrosion
Damage point selection by RGB intervals	$[R_1, R_2]$	[99,128]	[197,255]	[138,202]
	$[G_1, G_2]$	[0,65]	[98,198]	[0,60]
	$[B_1, B_2]$	[0,56]	[0,50]	[50,196]
SOR outlier processing	Number of neighbors	40	20	100
	Standard deviation ration	3	5	1
OPTICS clustering	Min neighbors	/	30	10
	Max distance (m)	/	0.005	0.005
	Min samples of a cluster	/	100	30

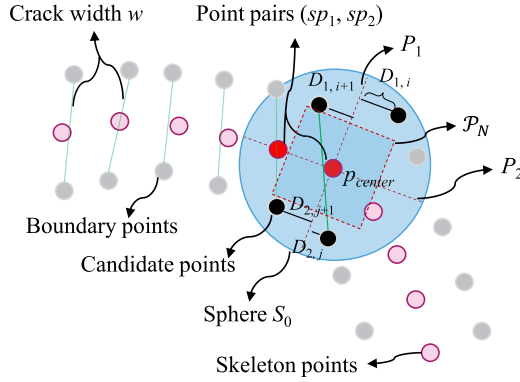**Fig. 5.** Key procedures for quantifying crack widths.

Fig. 6. Schematic diagram of the spherical region-based crack width estimation.

where proximity-based pairing may lead to erroneous or duplicate connections. As a result, it cannot be guaranteed that each skeleton point is always paired with its optimal neighbor for accurate plane estimation, nor that each point is used only once in the computation. To address this, a minimum spanning tree (MST) algorithm [52] is applied, treating each skeleton point as a node and using Euclidean distances as edge weights. This algorithm processes the skeleton points as unique point pairs (sp_1, sp_2) by minimizing the total connection cost, facilitating systematic crack width estimation at each skeleton point. Boundary points on either sides of a crack are detected and grouped into a point set *BP* by using the function named *compute_boundary_points()* from the Open3D library [48], providing edge information necessary for measuring the crack width.

In estimating crack width by the proposed spherical region-based width quantification method, the first point sp_1 serves as the center p_{center} of a sphere S_0 given radius r, as shown in Fig. 6. Boundary points within this spherical region are selected to form a candidate point set *CP* for the calculation of the crack width at the current location, as expressed in

$$CP = \{x | x \in BP, \|x - sp_1\|_2 < r\} \tag{8}$$

The radius r is fixed during the width estimation process, which is set to 2 mm in this study. S_0 serves to constrain the selection of crack boundary points, thereby reducing interference from adjacent branches or noise and improving computational efficiency. Since the accuracy mainly depends on the density and distribution of boundary points,

radius r is not a sensitive parameter. It is manually set based on the approximate maximum crack width observed from the boundary points. Once r is sufficiently large to span across the crack, the method can identify candidate point pairs regardless of crack width.

To prevent estimating crack width using candidate points on the same side of boundary, two perpendicular planes, P_1 and P_2, are set. Specifically, P_1 is perpendicular to the vector v from sp_1 to sp_2, and passes through p_{center}, which can be expressed as

$$P_1 : [n_1 | D_1] = [A_1 \quad B_1 \quad C_1 | D_1] = [v | - v \cdot p_{center}^\top] \tag{9}$$

To determine the appropriate surface plane for each crack region, the projected distance between the center p_{center} and each plane in the set $\mathscr{P}$ is calculated. The nearest plane corresponding to the minimum distance is selected and defined as $\mathscr{P}_N$. This process can be expressed as

$$\mathscr{P}_N = \underset{\mathscr{P}_i:[n_{0,i}|D_{0,i}]\in\mathscr{P}}{argmin} \frac{|n_{0,i} \cdot p_{center} + D_{0,i}|}{||n_{0,i}||} \tag{10}$$

Subsequently, Plane P_2 is constructed to be perpendicular to both $\mathscr{P}_N$ and P_1, and it passes through the p_{center}. Utilizing the normal vectors of the known planes $n_{0,N}$ and n_1, along with the p_{center}, P_2 can be expressed as

$$P_2 : [n_2 | D_2] = [A_2 \quad B_2 \quad C_2 | D_2]$$
$$= [n_{0,N} \times n_1 | - (n_{0,N} \times n_1) \cdot p_{center}^\top] \tag{11}$$

Once P_1 and P_2 are determined, the candidate points are divided into two point sets, DP_1 and DP_2, according to their locations relative to P_2. The distances between the points in sets DP_1 and DP_2 and plane P_1 are computed, denoted as D_1 and D_2, as exhibited in Fig. 6. The two points for which the sum of vertical distances is minimized are selected as width points, and the crack width w is calculated as the Euclidean distance between these width points, which can be expressed as

$$WP = \{(x_1, x_2) | (i*, j*) = \underset{i,j}{argmin}(D_{1,i} + D_{2,j}), x_1 = DP_{1,i*} \in DP_1, x_2 = DP_{2,j*} \in DP_2\},$$
$$w = \| x_1 - x_2 \|_2 \tag{12}$$

2.5.3. Area and volume quantification methods

For quantitative analysis of surface or block damage, such as corrosions and spalls, this paper proposes automatic damage quantification methods based on watertight surface mesh models, to precisely estimate the area and volume of damage with complex patterns, as shown in Fig. 7. Raw damage instances often exhibit large concave openings respect to the flat structure surface, complicating the generation of reliable damage surface meshes. To address this, the projection of raw
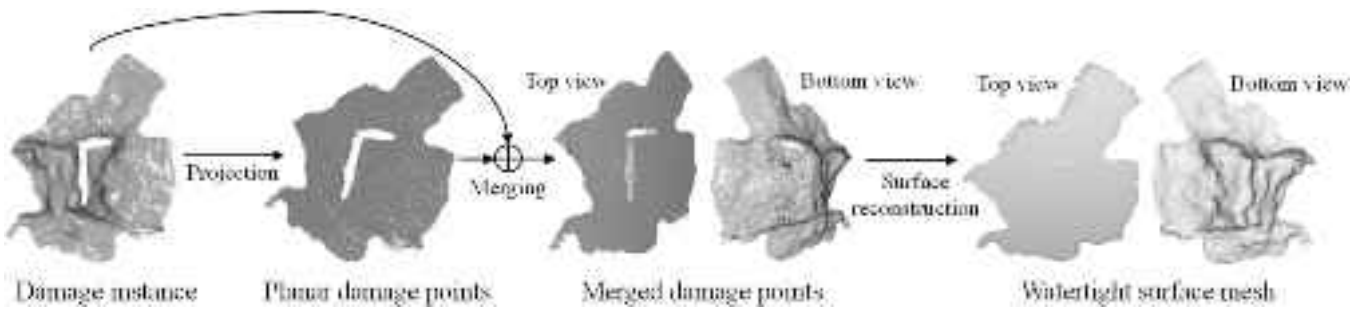
Fig. 7. Technical details of watertight surface mesh generation.

damage points on plane $\mathscr{P}_N$, named planar damage points, are firstly generated. The determination of plane $\mathscr{P}_N$ here follows the method used in the crack quantification section, but the Euclidean distance is calculated based on the centroid of the damage instance rather than the skeleton point p_{center}. Using Poisson reconstruction [53] with these planar damage points, a Poisson surface mesh model of the projection of damage instance on plane $\mathscr{P}_N$ is established. The area of the damage instance is then calculated by summing the areas of all small triangular elements in the surface mesh model.

For volume quantification, the planar damage points and the spatial damage instance need to be merged into a unified point set for subsequent surface mesh reconstruction. Since the projection plane $\mathscr{P}_N$ is fitted using points from the flat concrete surface where the damage is located, the projected planar damage points and the original surface points lie on the same plane, with boundaries directly aligned. Therefore, the two point clouds can be directly merged by loading them into the same 3D space without additional positional adjustments. Due to the significant holes and complex shapes of the damage instances, generating an accurate watertight surface mesh model is challenging. To achieve more robust surface reconstruction, the damage points set is processed by using Poisson reconstruction and hole filling [54] algorithms collaboratively. The Poisson reconstruction algorithm, known for its smooth surface reconstruction and ability to better estimate local details, is responsible for generating an initial triangular surface mesh model, referred to as the Poisson surface mesh. The density threshold of the Poisson algorithms is chosen empirically and is set to 0.001, which robustly generates surface meshes with sufficient reconstruction detail while balancing the computational cost. Since holes or gaps may remain after Poisson reconstruction, a hole-filling algorithm is applied to detect boundary loops and interpolate missing regions using moving least squares. When all holes in the Poisson surface mesh are filled, a closed watertight surface mesh model of a damage instance can be obtained. Notably, the direction information of the points is utilized to distinguish the internal and external regions of the surface. It is necessary to verify and adjust the normals for both the planar points and damage instances in advance, ensuring all normals are oriented toward the interior of the merged damage points. Finally, the volume of the closed mesh model is measured by using PyMeshLab [55] library, achieving volume quantification for the damage instance. Table 4 provides a unified summary of

the algorithms used in each step, along with their corresponding objectives.

3. Experiments verification

To demonstrate the performance of the proposed damage detection and quantification approach, an experiment is conducted using a 1/10 scale bridge pier located on the campus of Guangzhou University. The bridge pier comprises a trapezoidal pier cap and a rectangular pier shaft, as shown in Fig. 8. Key dimensions of the pier are measured: the total height of the pier H is 2.5 m and the pier shaft height h is 1.75 m. The length L and width W of the top of the pier cap are 2.35 m and 0.7 m, respectively. The cross section of the bottom of pier cap is the same as the pier shaft, with a length l of 1.1 m and a width w of 0.35 m. Various types of damage, including crack, spalling, and corrosion, are observed at the bottom of the pier shaft.

To acquire detailed images of the pier, a DJI Phantom 4 RTK quadrotor UAV with a monocular camera is used. The camera, featuring a 24 mm lens and a 20-megapixel resolution, is mounted on a gimbal that allows for an adjustable pitch angle ranging from −90–30 degrees, providing a capability to capture clear images of the pier surface from different viewpoints. During data acquisition, the UAV is controlled to circumnavigate the pier, capturing images from varying altitudes, as depicted in Fig. 8. Due to nearby plants and billboards obstructing the field of view and restricting flight space, the shooting distances are adjusted between 1.00 and 5.00 m to ensure high-quality images are acquired while maintaining operational safety. To guarantee the accuracy of 3D reconstruction, an overlap of over 50 % between consecutive images is maintained throughout the collection of pier images. Because the test scenario involves a campus-based bridge pier, the imaging and data acquisition conditions are relatively controllable. To obtain sufficiently dense points for damage measurement, the images of damaged areas are captured in closer proximity and with higher overlap, thereby providing more detailed and diverse viewpoints. Given the small size of the pier, 412 UAV images are collected to cover all necessary viewpoints, as illustrated in Fig. 9. To ease the computational burden of DL model training and 3D reconstruction, the raw images are downscaled from 4864 × 3864 pixels to 1600 × 1200 pixels.

To train a damage segmentation model, a dataset of images depicting

Table 4
Algorithms used in area and volume quantification.

|	Step	Algorithm/Software used	Objective
Area estimation	Planar damage points generation	Points Projection onto plane $\mathscr{P}_N$	Convert raw damage points into a planar representation
	Mesh generation	Poisson reconstruction	Generate a Poisson surface mesh of the planar damage instance
	Area calculation	PyMeshLab library: *get_geometric_measures()* function	Compute the area of the mesh model
Volume estimation	Points merger	Point cloud merging	Merging of planar and spatial damage points
	Mesh generation	Poisson reconstruction and hole-filling algorithms	Generate a closed watertight mesh representation
	Normal adjustment	Open3D library: *orient_normals_towards_camera_location()* function	Correct normal vectors oriented toward the interior
	Volume calculation	PyMeshLab library: *get_geometric_measures()* function	Compute the volume of the closed watertight surface mesh model
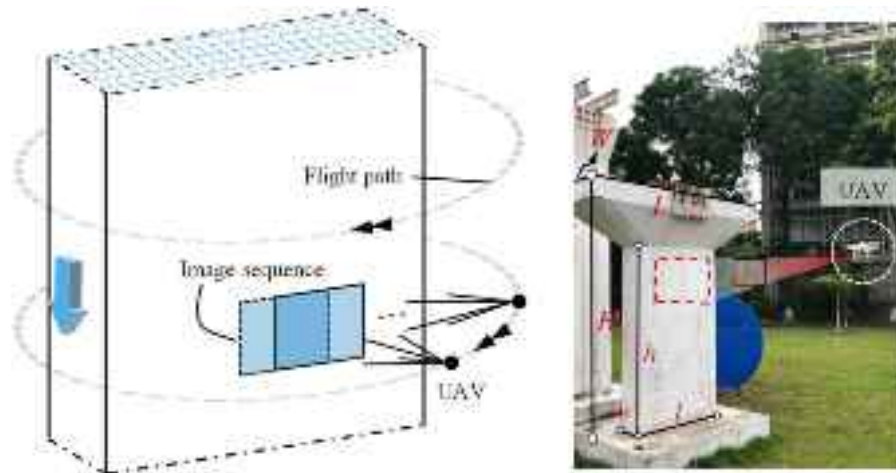
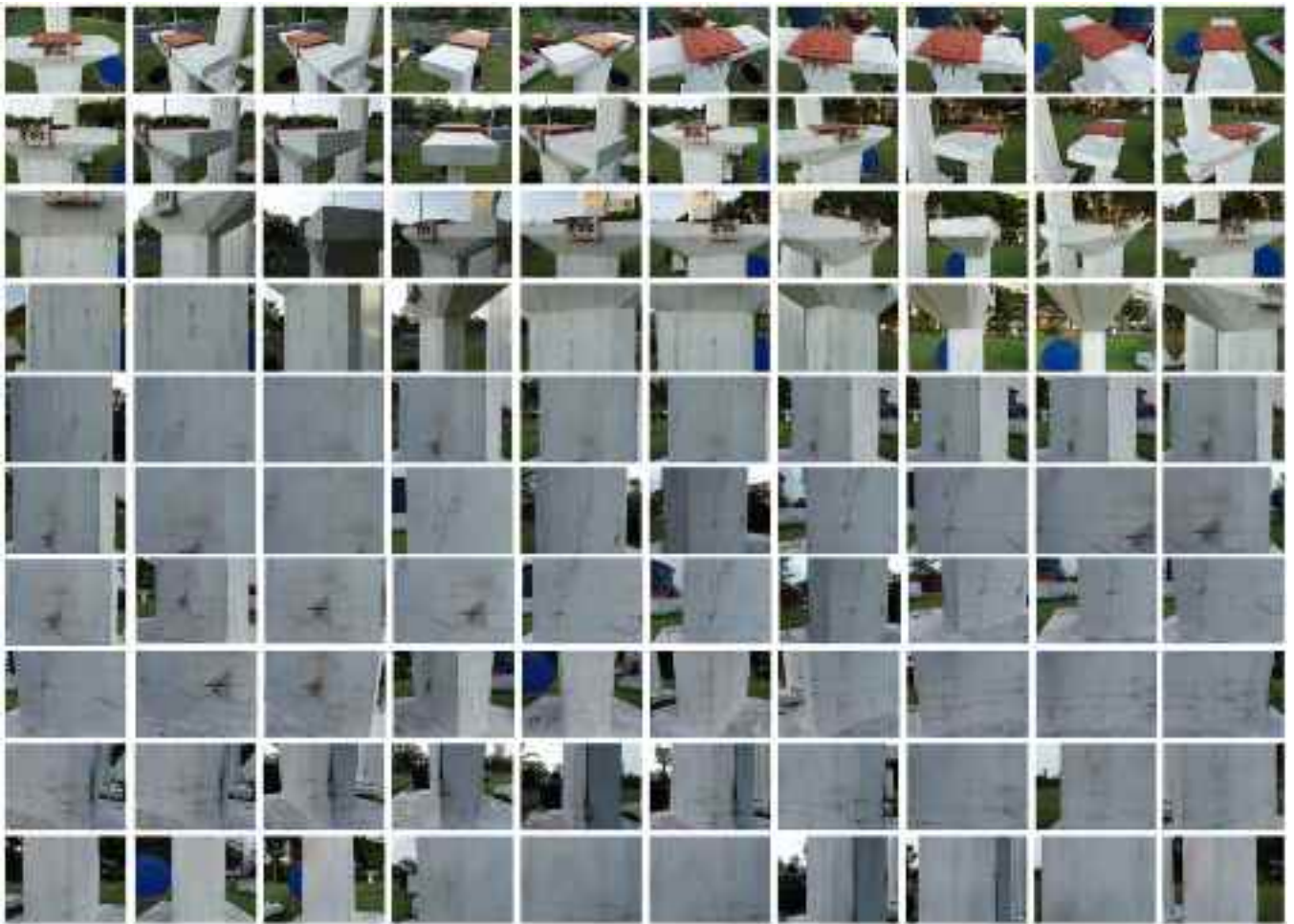**Fig. 8.** UAV image acquisition of the bridge pier.

Fig. 9. Examples of UAV collected images.

typical damage are required. In this study, 127 images comprehensively covering pier and structural damage are selected to maximize sample diversity by minimizing repeated viewpoints. For pier damage, pixel-wise annotations include three damage types: crack, spalling, corrosion. Fig. 10 illustrates the bridge pier damage samples after augmentation. The augmentations generated 404 additional images, bringing the total number of training images to 505 when combined with the

original 101 training images. To provide a comprehensive evaluation, two classical semantic segmentation networks, DeepLabV3 + and U-Net, are employed for comparison with SegFormer. To ensure a fair and consistent comparison, all three models are trained using identical experimental configurations and hardware settings. The model training program codes with the DL framework PyTorch.

To evaluate the effectiveness of Poisson reconstruction for volume

Fig. 10. Several augmentation examples of the damage image.

Table 5
Schemes and thresholds used in the comparative experiment.

Algorithm	Parameter	Threshold
Ball-Pivoting	Radius (mm)	$[d, 2d]$
		$[2d, 4d]$
Alpha Shape	Alpha value	0.01
		0.03

quantification, a comparative experiment is conducted using Ball-Pivoting [56] and Alpha Shape [57] algorithms generated watertight surface mesh models as alternatives to Poisson reconstruction. Table 5 details the schemes and parameters of these volume quantification methods. The average distance d between each point and its nearest neighbor in a crack is used to set thresholds for the Ball-Pivoting algorithm. Thresholds for Alpha Shape are chosen empirically. For a more comprehensive analysis, each surface reconstruction algorithm is tested twice with varying parameter thresholds. All computation implementations are performed on a computing platform with the following specifications: 13th Intel(R) Core (TM) i7–13700K@3.40 GHz, 64 GB RAM, and a single NVIDIA GeForce RTX 4090 GPU.

To validate the effectiveness and reliability of the developed damage detection and quantification approach, manual measurements of damage instances are conducted to provide reference values. These measurements include crack width and spalling volume. For crack width measurement, 10 crack measurement points are carefully selected through inspecting both the structural surface on-site and its corresponding point cloud, ensuring that the crack features are clearly distinguishable in both contexts. For each measurement point, three-dimensional coordinates are determined relative to a fixed reference point P_R located at the surface of the structure, as illustrated in Fig. 11. This shared reference system is applied to locate the corresponding points in the point cloud model, ensuring that manual and automated measurements are conducted at exactly the same spatial locations. A scale-reading microscope (0.05–4 mm range) is used to measure the width at each point, with each measurement repeated three times to reduce operational deviations.

The complex geometry of spall presents significant challenges for accurately determining volume through direct dimensional measurement. Ref. [58] validated the accuracy of a concrete loss volume estimation by measuring the volume of water used to fill the spalling region. While straightforward and cost-effective, this approach requires the damaged surface to face upward, making it inapplicable in this study due to the fixed pose of the pier. Hence, a clay drainage method for volume measurement is employed to quantify concrete loss by converting solid volume measurements into water volume measurements. Firstly, a solid model of local concrete loss is created by continuously filling the loss area with oil-based clay until the surface level reaches the surrounding surface level. The experimental setup and procedures are illustrated in Fig. 12. To reduce experimental error, the experiment is repeated three times by different experimenters. After each filling, the oil-based clay models are carefully detached from the filled areas. To ensure accuracy and prevent volume changes due to drying-induced deformation, a drainage experiment is conducted immediately after obtaining the clay models, as shown in Fig. 13. The specific material parameters of the clay are not obtained. However, it is observed to have a density greater than water and a slightly oily surface, suggesting possible hydrophobic properties. To further minimize the risk of water absorption affecting the volume measurement, the clay models are pre-soaked before the drainage experiments to fully saturate their surfaces. This precaution aims to stabilize the material and reduce potential measurement errors. Due to the difficulty of manually measuring surface area with such complex geometric shapes, the surface and block damage areas are measured from the real-size point cloud by manually selecting polygonal regions. The process is repeated three times to minimize the impact of operational deviations.

Given that the bridge pier provides only one representative spalling instance, two additional samples are introduced to validate the accuracy and reliability of the proposed automated volume measurement approach. The spalling samples are manually created on two concrete slabs. As shown in Fig. 14, both slabs have a width W_{cs} and thickness T_{cs} of 300 mm and 100 mm, respectively. The spalling area on Slab 1 has an approximate maximum width of 120 mm, while the spalling area on Slab 2 is approximately 60 mm wide. A total of 38 and 64 images at a resolution of 4032×3024 pixels are collected for Slab 1 and Slab 2, respectively. The semantic point clouds were generated follow the same procedure as the bridge pier.

4. Results and discussions

4.1. Semantic point cloud reconstruction for the bridge pier

This section thoroughly presents and discusses the main results obtained from the experimental validation of the proposed approach. From Fig. 15, it can be observed that the training and validation curves of both Loss and Accuracy for the three models remain highly consistent throughout the training process, with only limited fluctuations in the validation curves at a few epochs. Among them, SegFormer and U-Net exhibit less fluctuation in the validation curves compared to DeepLabV3plus, indicating a better learning of damage features. The model accuracies on both the training and validation datasets remain at a high level and consistently improve over the epochs, mirroring the trend observed in the loss curves. In the end of training, the peak validation accuracies of all three models achieved impressive results, with each exceeding 0.98. SegFormer completes the entire training process in

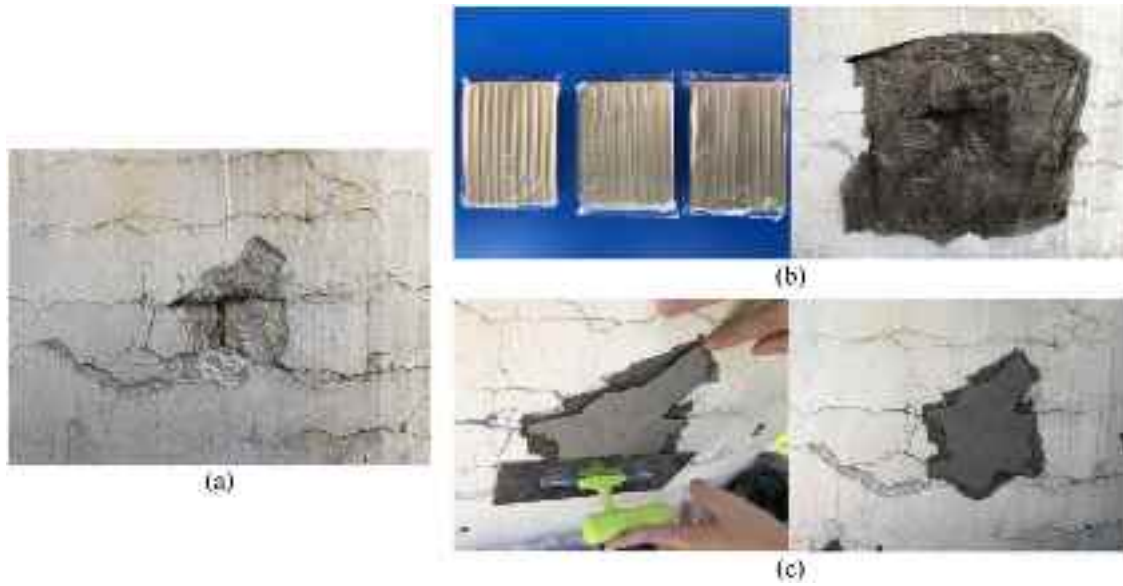**Fig. 11.** Measurement points for crack widths.

Fig. 12. Volume filling of a concrete spall: (a) Spalling region; (b) industrial oil-based clay and plastic film, and (c) volume filling.
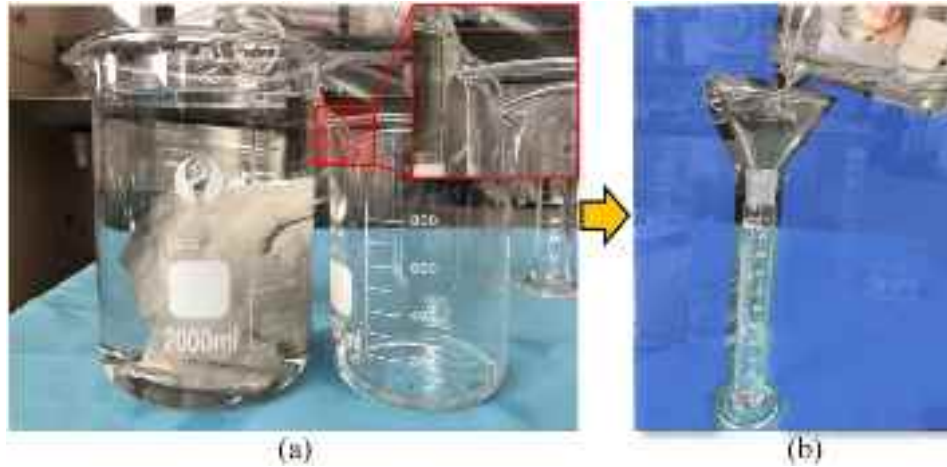

Fig. 13. Drainage experiment: (a) Experimental instruments; (b) Drainage process of oil-based clay.

Fig. 14. Concrete slabs with spalling damage: (a) Slab 1 with large a spall; (b) Slab 2 with a small a spall.

approximately 6 h, saving about 40 % of training time compared to the other two networks, showcasing efficient computation due to its lightweight architecture. Therefore, both the accuracy and efficiency of SegFormer in multiclass damage segmentation are exceptional. The parameters of the optimally performing three models are saved for further validation.

The capability of the trained models for multiclass damage segmentation is further validated using the testing dataset. As shown in Table 6, SegFormer achieved the best results across all three types of

damage, with IoU scores exceeding 97 % for both corrosion and spalling segmentation. Although crack detection remains the most challenging task, SegFormer still outperforms DeepLabV3plus and U-Net in this category, demonstrating its superior ability to capture subtle and fragmented damage patterns.

In fact, the segmentation accuracy for cracks is significantly lower than that for the other two types of damage. Unlike surface-like damage with rich texture features that support stable learning, fine cracks are typically narrow, intricate, and reticulate, making them highly sensitive
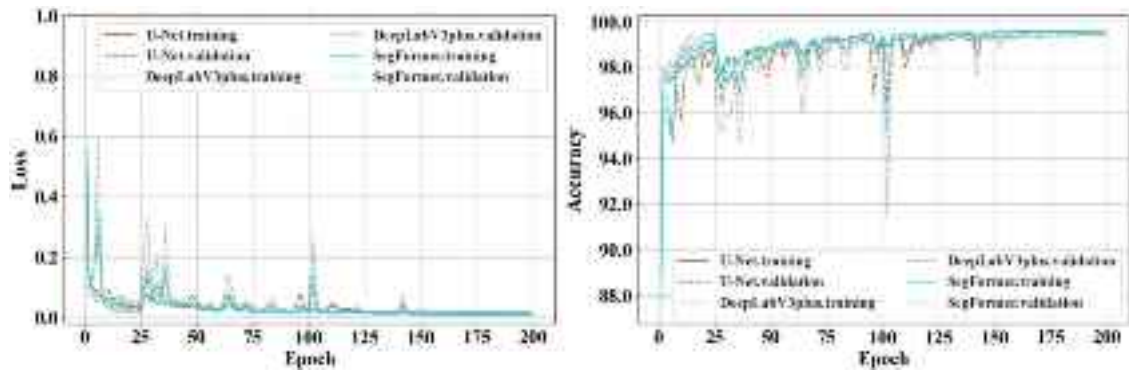
Fig. 15. Loss and Accuracy curves of training and validation.

Table 6

Damage segmentation performance of the trained models.

Network	Segmentation IoU		
	Corrosion	Crack	Spalling
U-Net	0.9403	0.6110	0.8933
DeepLabV3 +	0.9300	0.5969	0.8842
SegFormer	0.9828	0.7572	0.9721

to small deviations in contour prediction. Minor discrepancies between predicted and ground truth masks can significantly reduce IoU scores, even if the predicted shape is visually acceptable. Cracks also often exhibit low contrast and weak texture cues, especially under complex lighting or surface conditions, limiting the performance of RGB-based segmentation models. Additionally, ambiguous crack boundaries make annotation challenging. Ground truth labels may not precisely match actual contours, and while models may extract more accurate edge features, such misalignments still lead to IoU inconsistencies.

To intuitively validate the segmentation results, Fig. 16 presents several examples of damage segmentation produced by these three models. The enlarged views highlight that the damage regions annotated by SegFormer are more closely aligned with the ground truth compared to the results from DeepLabV3plus and U-Net, indicating the most accurate damage segmentation. DeepLabV3plus and U-Net face challenges in detecting fine and reticulate cracks, with their predictions showing noticeable discontinuities and fragmentation. In contrast, SegFormer, benefiting from its superior global feature extraction capability, effectively reduces these omissions and successfully detects cracks
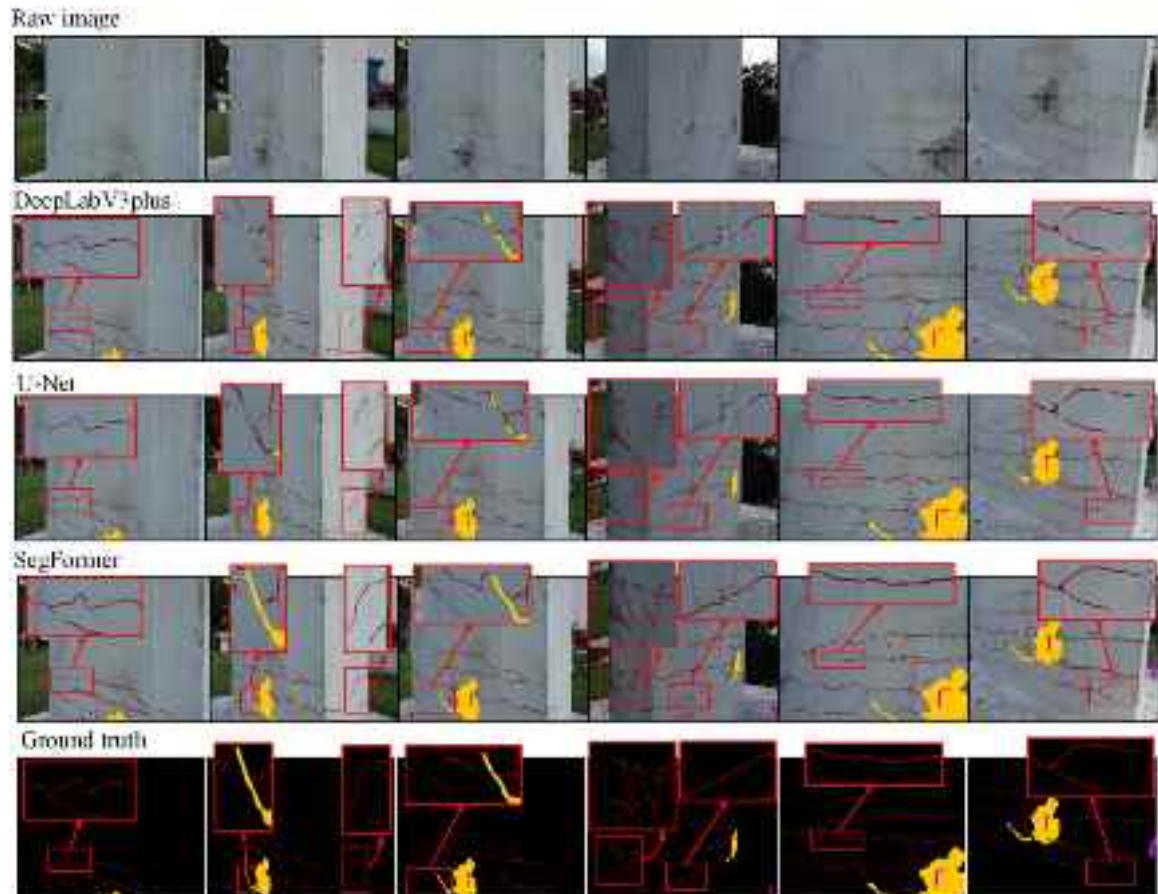

Fig. 16. Visualizations of damage segmentation for the bridge pier.

with the highest level of completeness. These visualization results of test set demonstrate the powerful capability of SegFormer in segmenting complex damage. Although its quantitative IoU scores for cracks are not particularly high, the practical segmentation quality is sufficient to support downstream tasks such as semantic point cloud reconstruction and crack width measurement.

The optimal SegFormer model is used to label all images involved in the 3D reconstruction. In fact, these UAV-captured images, sourced from various outdoor perspectives and positions, present some natural variations in viewing angles and lighting conditions. Despite these variations, damage identification in each image is remains effective and yields results largely consistent with those shown in Fig. 16. Most of the damage pixels in the region are accurately identified, with only a few areas with less distinct features experiencing some degree of missed detection. Additionally, while variations in angle and illumination could cause slight discrepancies in damage recognition across different images, the SfM reconstruction process inherently balances the colorization of point clouds from multiple viewpoints. As a result, the final semantic point cloud can still be effectively reconstructed, providing a robust and accurate representation of the detected damage, as shown in Fig. 17.

The 3D reconstruction of the semantic point cloud for this bridge pier takes approximately 86 min. From Fig. 17, it can be observed that the coverage of the target object from numerous viewpoints demonstrates that a sufficient number of feature points are extracted for the 3D reconstruction of the pier. The generated semantic model showcases realistic textures and intuitive semantic information, enabling global localization and automatic quantification of damage. While some regions with weak or uniform textures (such as clean concrete surfaces) lead minor holes or sparse points during reconstruction, the presence of surface damage typically enhances local texture variation. This texture enrichment leads to improved reconstruction quality in damaged areas compared to intact regions. By observing the enlarged view of multiclass damage region at the bottom of pier, the three types of damage are represented by distinct colors: crack in red, spalling in yellow, and corrosion in magenta. These colors closely align with those shown in the images, indicating that the damage segmentation results are effectively integrated into the semantic point cloud. The clear distinction between different types of damage and structural surface, along with the dense aggregation of damage points, highlight a continuous and complete damage representation, enhancing the accuracy of subsequent quantitative analyses. Although this study adopted close-range imaging to ensure reconstruction quality, the proposed framework is not limited to such settings. In practice, when UAVs cannot approach the surface, sufficient detail can still be obtained through optimized flight paths with higher overlap or by using high-resolution or zoom cameras. Advances in reconstruction algorithms are also expected to mitigate the impact of increased imaging distance. Therefore, as long as the semantic point cloud retains adequate detail in the damage regions, the method remains applicable for quantitative defect detection and measurement.

To further investigate the robustness of semantic segmentation, the SegFormer is evaluated under several more challenging conditions, including very close distances, surface shadows, and nighttime lighting. The testing results are shown in Fig. 18. It can be observed that while damage segmentation accuracy decreases under these challenging scenarios, the model still detects cracks with distinct geometric morphology. However, challenges such as the misclassification of few stains as spalling and background complexity caused by varying illumination conditions can adversely affect damage detection, particularly when the environmental conditions deviate significantly from those encountered during training. These observations underscore the importance of incorporating environment-specific data when aiming for reliable performance in real-world applications. While this study primarily focuses on verifying the feasibility and effectiveness of the proposed method for damage quantification, expanding the dataset to cover more diverse scenarios and improving the model's generalization capabilities represents an important direction for future work.

4.2. Multiclass damage quantification

4.2.1. Damage instances extraction

Fig. 19 displays the results of damage point extraction from the semantic point cloud, where crack, spalling, and corrosion are identified based on their RGB value intervals. Statistical data reveals 10845 points for crack, 19067 points for spalling, and 1717 points for corrosion, demonstrating that point clouds generated from standard UAV-captured images using SfM-MVS reconstruction are sufficient to support surface damage quantification. Outliers are observed near the damage points, primarily due to environmental noise, segmentation errors, and feature matching inaccuracies during the generation of semantic point cloud. For the test samples, the crack contours are highly complex, with most cracks being narrow and surrounded by fine surface scratches and stains on the concrete. These factors make precise manual verification to denoising challenging. As shown in Fig. 19(b), the detected small crack segments primarily appear at the crack ends and are sparsely distributed. Their removal has minimal influence on the overall width quantification of the main crack region. Therefore, this suggests that most outliers are accurately detected and removed using the configured parameters of the SOR algorithm, resulting in cleaned and reliable damage instances.

The OPTICS clustering effectively identifies two distinct spalling clusters and a corrosion cluster, as shown in Fig. 19 (c) and (d). The larger spalling cluster, characterized by its complex concave and convex surface and irregular holes caused by exposed rebar, represents typical
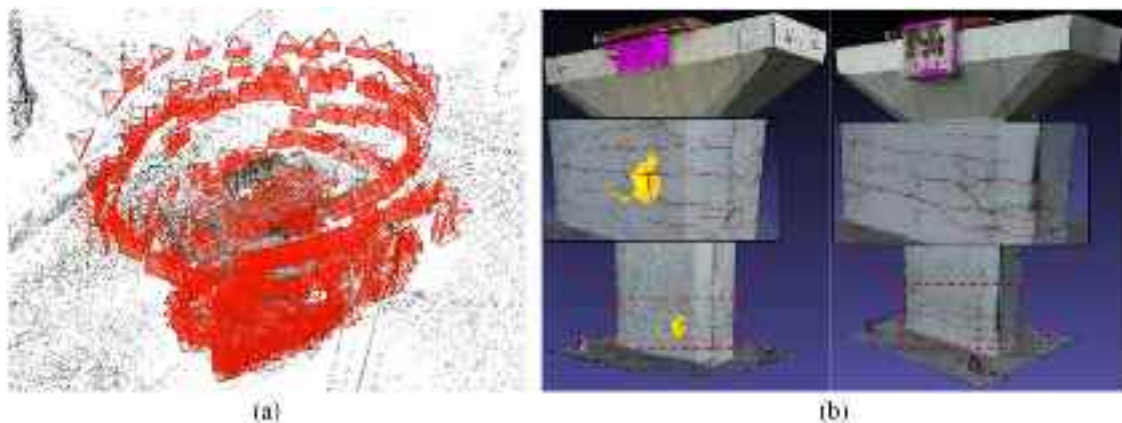

Fig. 17. 3D reconstruction of the semantic point cloud of the pier: (a) Sparse scene reconstruction from SfM process; (b) Semantic point cloud model after background removal.

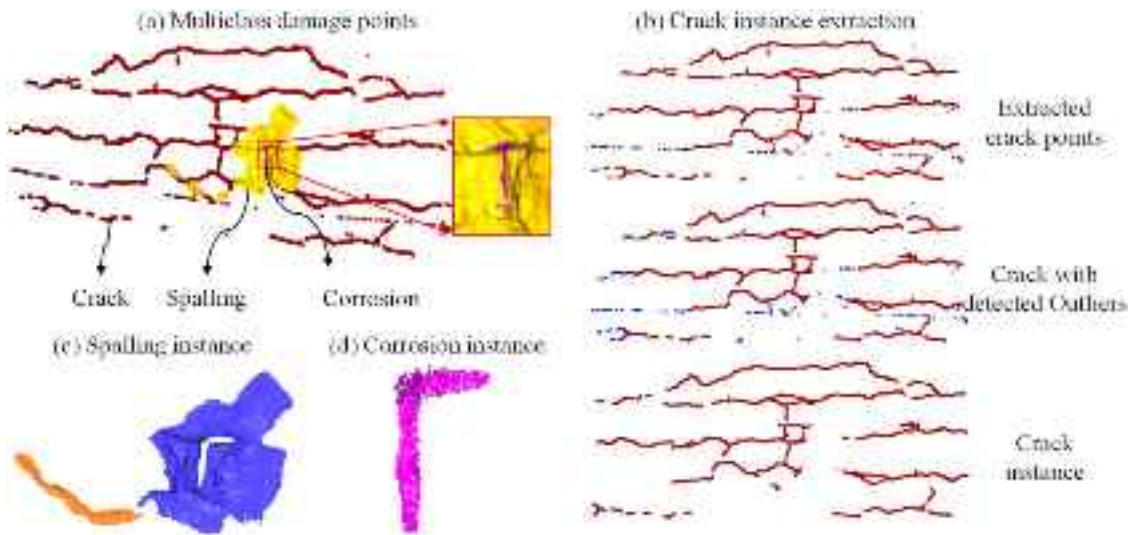**Fig. 18.** Damage segmentation visualizations under challenging situations.

Fig. 19. Extracted damage instances from semantic point cloud.

structural spalling damage. The corrosion instance of exposed rebar is accurately identified, while other smaller corrosion points are deemed negligible and thus ignored in subsequent analyses. Consequently, three damage instances, including the reticular crack, larger spall and corrosion of an exposed rebar are used for validating the proposed methods for crack width, spalling and corrosion volume and area quantification. Three plane equations are fitted to the flat concrete surfaces where these instances occur, providing a set of candidate reference planes for optimal selection in each case. The parameters of these plane equations are listed in Table 7.

4.2.2. Reticular crack quantification

Fig. 20 presents the results of width quantification for the reticular crack. The topological characteristics of the detected boundary points and generated skeleton points are closely resembled the crack instance,

highlighting the powerful capability of the proposed method in processing cracks with complex geometry. Three examples of detailed crack quantifications are visualized in the enlarged views. It can be observed that the crack points furthest from the centers are accurately identified as boundary points. The skeleton points are generated near the centers, demonstrating a good alignment with the boundary points at corresponding locations. In Fig. 20, most of green lines correctly connect the appropriate boundary points on both sides of the crack, even in cases where the boundaries are very close to each other, with many crack widths calculated at less than 1 mm. This demonstrates that the proposed spherical region-based method is capable of accurately estimating widths of tiny cracks. Additionally, reliable width estimations are achievable even in areas with fewer boundary points, highlighting superior robustness and adaptability of the proposed method.

To validate the accuracy of estimated crack widths, manual measurements are conducted, as shown in Fig. 21. It should be noted that measuring such tiny crack width is challenging, even manually. Table 8 presents a comparison between the manual and automated methods for measuring crack widths. Each manual measurement value in Table 8 represents the average of three repeated measurements to account for potential human error. The average estimated crack widths are consistently larger than the manually measured values. This overestimation is likely attributable to that the narrowness of cracks, which leads the manual delineation of crack contours in training dataset preparation

Table 7
Coefficients of plane equations fitted to pier surfaces.

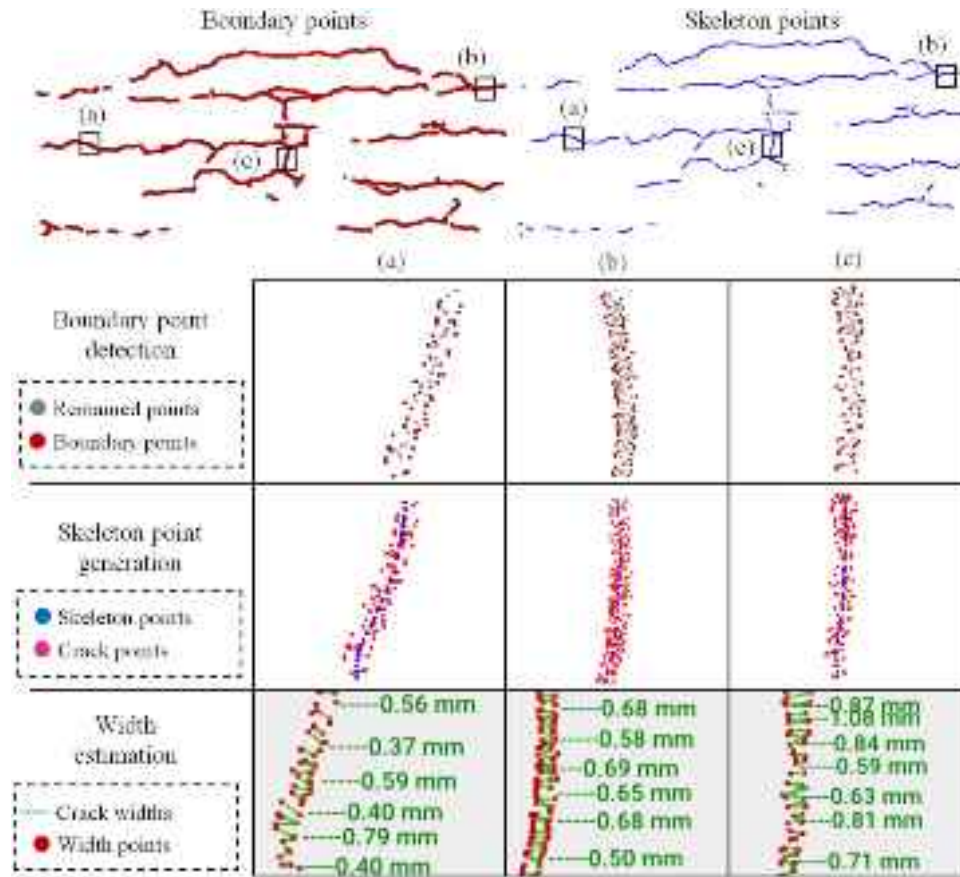| Plane ID	Coefficients			
	A	*B*	*C*	*D*
1	0.94	0.27	0.19	0.43
2	0.85	0.47	0.21	0.75
3	0.70	0.66	0.25	0.96

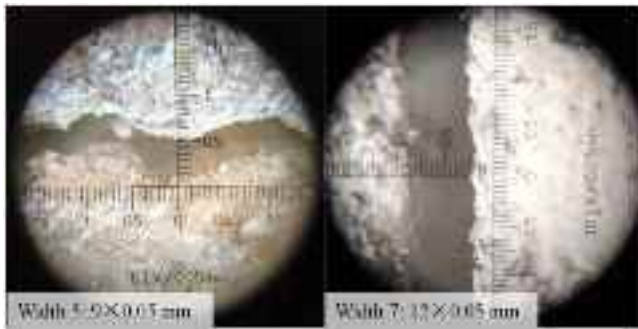Fig. 20. Width quantification results of the reticular crack.

Fig. 21. Manual measurement of crack width by scale-reading microscope.

Table 8
Comparison of manual and automated measurements.

ID	Manual (mm)	Automated (mm)	Absolute Error (mm)	Relative Error (%)
1	0.45	0.46	0.01	3.0
2	0.60	0.68	0.08	12.8
3	0.80	0.95	0.15	19.2
4	0.80	0.83	0.03	3.3
5	0.45	0.57	0.12	27.4
6	0.65	0.74	0.09	14.4
7	0.60	0.68	0.08	12.8
8	1.45	1.60	0.15	10.3
9	0.65	0.77	0.12	17.9
10	0.60	0.71	0.11	18.3
Average	0.705	0.799	0.09	13.9
Standard deviation	0.287	0.311	0.046	7.4

particularly difficult. In many cases, the annotated boundaries of cracks extend slightly into adjacent concrete regions. This may introduce a slight overestimation bias into the model training process, resulting in predicted crack widths being larger than the true values. Crack ID 5 presented the narrowest width, making it more difficult to annotate accurately. Additionally, the reconstructed crack points in this region are relatively sparse, and the line connecting the two width points deviates more substantially from the crack skeleton normal.

To further assess the consistency of both methods, the standard deviations of average crack widths across the 10 measurement points are calculated. The manual method exhibited a standard deviation of 0.287 mm, while the automated method shows a slightly higher value of 0.311 mm. The minor difference indicates a comparable level of precision between the two approaches. Moreover, the difference in mean values between the two methods is found to be less than 0.1 mm, with a corresponding standard deviation of 0.046 mm, suggesting that the automated measurement errors remain within an acceptable range and that the method possesses strong potential for replacing manual crack width measurement in practical applications. The validation results confirm the effectiveness of the crack quantification method in estimating small crack widths, highlighting its potential for practical engineering applications. Due to obstacles around the bridge pier, it is difficult to conduct more comprehensive resolution analyses in this study. Most images of the cracked regions of the pier are captured at approximately 50–80 cm from the structure, within which the method reliably measures cracks with a minimum width.

4.2.3. Volume and area quantification of spall and corrosion damage

To validate the necessity of each procedure of the proposed method in generating watertight surface mesh, an ablation study is conducted under four different cases, with the results visualized in Fig. 22. Without
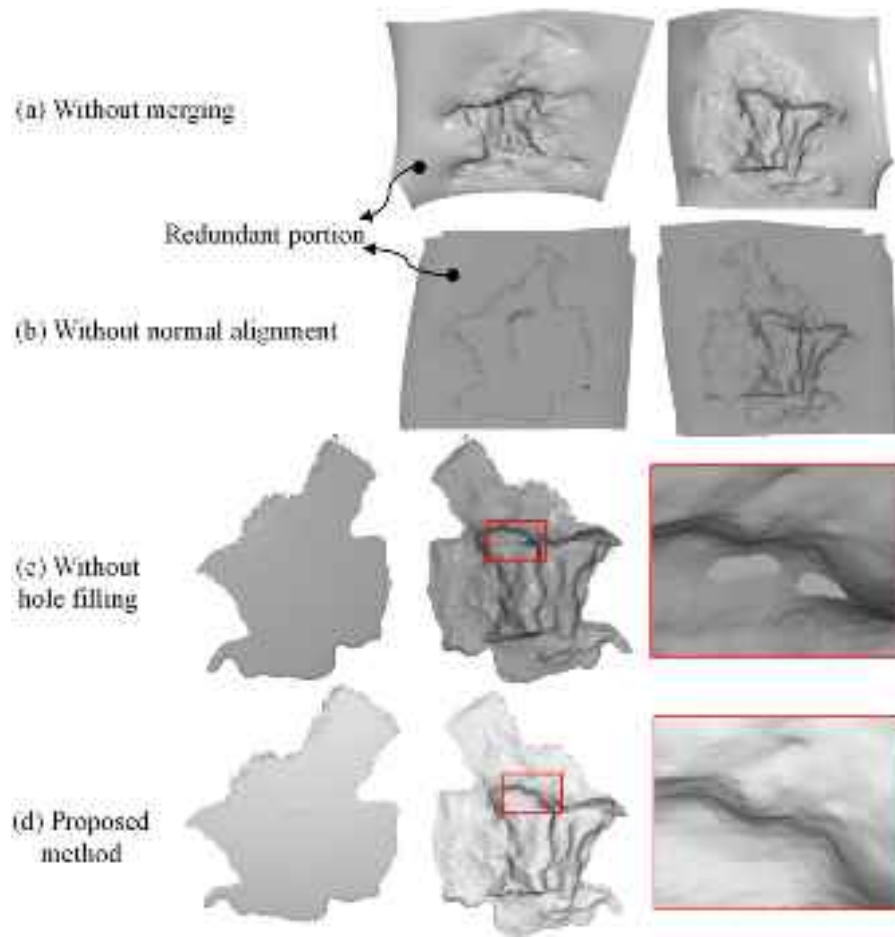
Fig. 22. Generated poisson or watertight surface mesh models of the spall.

merging planar points, the generated Poisson surface mesh exhibits large openings and redundant portions, as shown in Fig. 22(a). Even though the generated small triangular meshes fill the obvious holes at the location of corrosion instance of exposed rebar, it still fails to create a watertight model due to the excessively large hole at the opening. Performing Poisson reconstruction on merged damage points with incorrect normal orientations is ineffective in generating a desirable surface mesh model, as depicted in Fig. 22(b). The issue arises because the normal directions of spatial and planar points are not naturally oriented towards the center of their merged model, affecting the accurate identification of the interior and exterior surfaces of the spall. As a result, the surface appears to be composed of two separate parts. Moreover, due to the influence of the redundant portions, the subsequent hole filling process is ineffective in repairing holes in the surface mesh model. As shown in Fig. 22(c), in the case of no hole filling, the resulting surface mesh is smooth and continuous, with its concave and convex surface closely resembling the actual conditions. However, minor holes appear in areas with previously sparse or absent points, which lacks robustness in generating a watertight surface mesh. In contrast, the proposed method, which incorporates all procedures, adaptively repairs holes by generating smooth triangular meshes without significantly changing the initial geometry of the surface mesh model. Consequently, the proposed volume quantification method effectively utilizes these essential procedures, demonstrating robustness and accuracy in generating a watertight surface mesh model for surface or block damage.

In addition to the ablation study, the superiority of the proposed method is validated by a comparative study. Fig. 23 reveals that the Ball Pivoting algorithm generates spalling surface mesh with obvious holes and extensive noise, while the Alpha Shape algorithm faces challenges in

creating high-resolution and realistic surfaces. Although increasing the thresholds for both Ball Pivoting and Alpha Shape algorithms reduces the number and size of holes, reliable geometric details are sacrificed. Consequently, the Ball pivoting and Alpha Shape algorithms are not appropriate for generating the surface mesh model of complex damage due to their high sensitivity to parameter threshold variations. Two surface mesh models produced by Poisson reconstruction, regardless of the threshold used, are closely similar in resolution and shape, with only minor discrepancies in hole characteristics. In contrast to the previous two algorithms, the Poisson algorithm allows for a more generic configuration for damage surface mesh generation. Besides, the holes present in the Poisson surface mesh generally have smooth edges, which can be effectively detected and filled using a hole filling algorithm. Therefore, the Poisson algorithm is considered the optimal option for surface reconstruction in the proposed volume quantification method.

Similarly, the proposed volume measurement method is used to generate watertight surface mesh models for two additional slab spalling instances. The results, along with their corresponding reconstructed semantic point cloud and the spalling instances, are shown in Fig. 24. The reconstructed point clouds capture the surface texture and geometry of the slabs and their spalling areas with sufficient density and detail. The spalling instance of Slab 1 (Slab spalling instance 1) contains 23812 points, while that of Slab 2 (Slab spalling instance 2) contains 7583 points. Both the two mesh models are sufficiently high-resolution, showing a high level of geometric consistency with the original point clouds. Despite some minor point sparsity near the bottom edge of the slab spalling instance 2, caused by occlusion and shadows within the narrow and deep cavity, the method is still able to reconstruct a continuous surface effectively and fill potential small holes, which
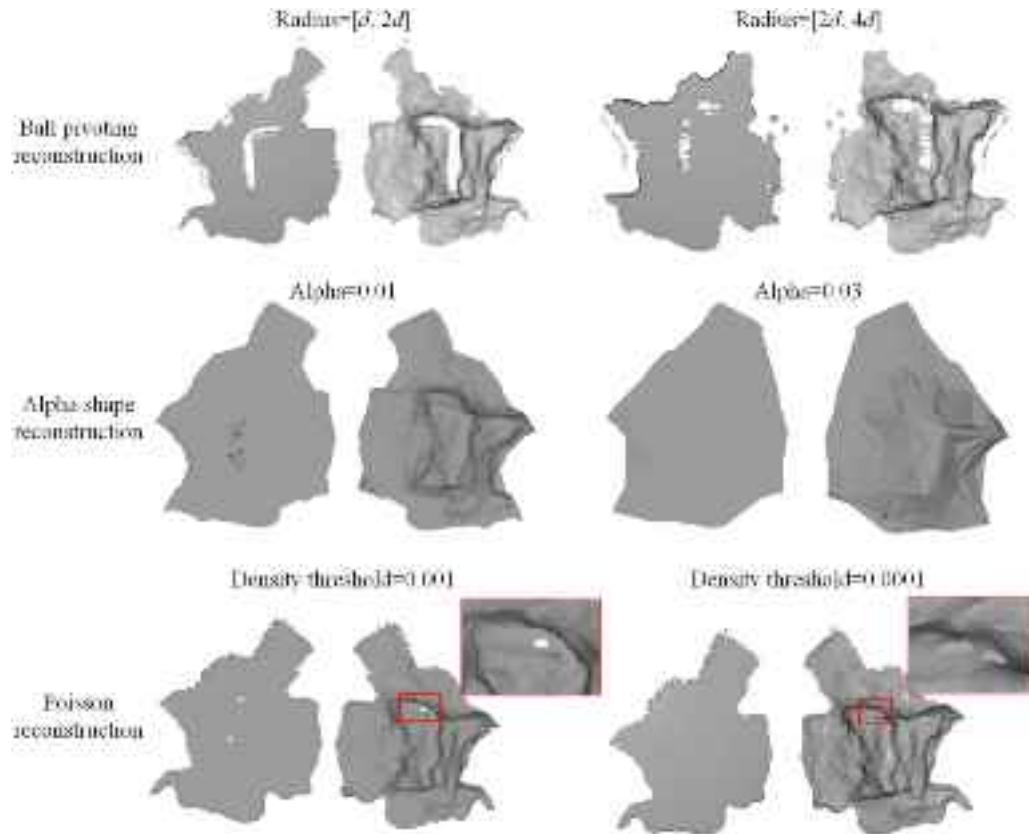
Fig. 23. Comparison of mesh models generated by different surface reconstruction algorithms.
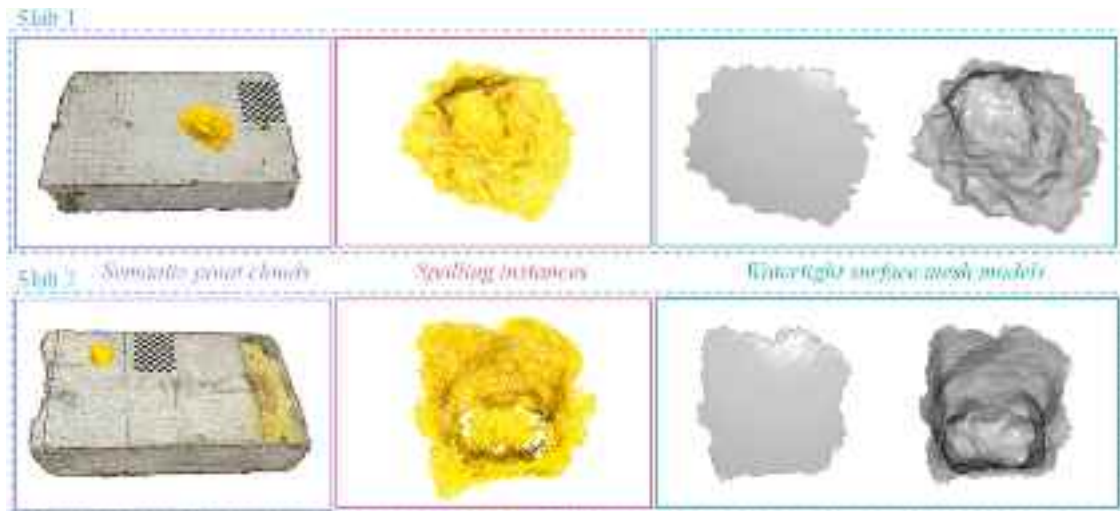

Fig. 24. Results of reconstructing concrete slab spalling damage for volume measurement.

further supports the robustness of the proposed approach.

Fig. 25 illustrates the area quantification results for the corrosion and spalling instances obtained using the proposed method and manual measurement. The planar meshes of both the corrosion and spall feature continuous and accurate contours that closely align with their corresponding spatial damage instances, indicating the mesh generation process is promising for accurate area calculation. Consequently, the proposed sematic point cloud based method effectively estimates areas of the corrosion and spall, with relative errors of only 0.32 % and 0.01 %, compared to the average values from manual measurement. The proposed area estimation method, based on the Poisson reconstruction,

demonstrates a remarkable capability for automatically quantifying areas for surface or block damage, highlighting its potential for engineering applications.

Figs. 25–27 display the resulting clay models and displaced water volumes from three repeated manual measurements (Tests A, B, and C) conducted for the pier and slab spalling instances. From the bottom surfaces, it can be seen that the clay models exhibit similar surface characteristics, indicating a good fit to the spall with its complex surface. However, the top surfaces of clay model reveal minor differences in the geometric contours of the planar surfaces, primarily due to the difficulty in accurately determining the contours of the spall, which are obscured
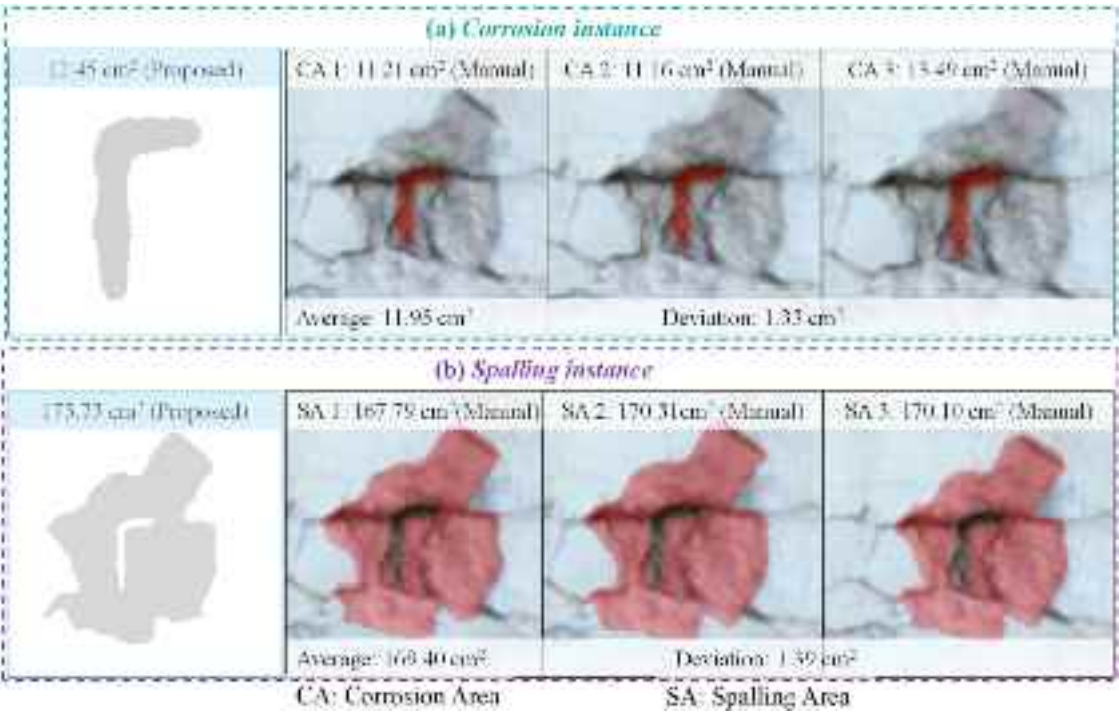
Fig. 25. Comparison of the proposed method vs. manual measurement for both (a) corrosion instance and (b) spalling instance.
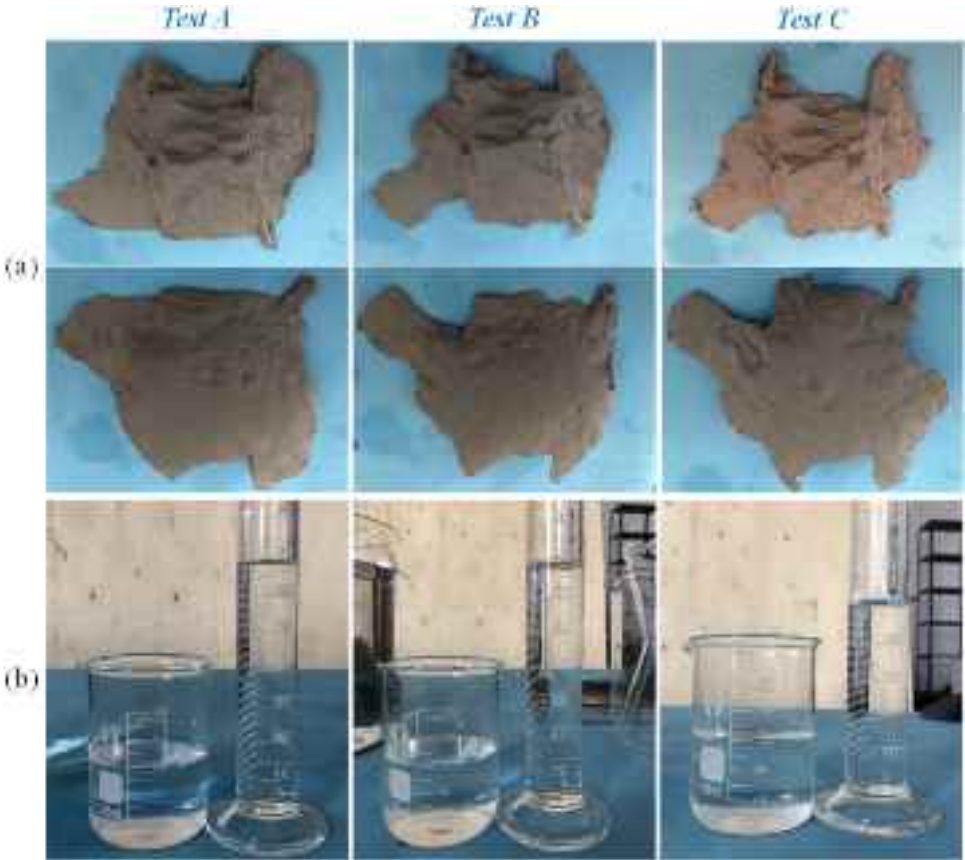

Fig. 26. Results of three repeated measurements of a pier spalling instance: (a) oil-based clay models and (b) drainage volume measurements.
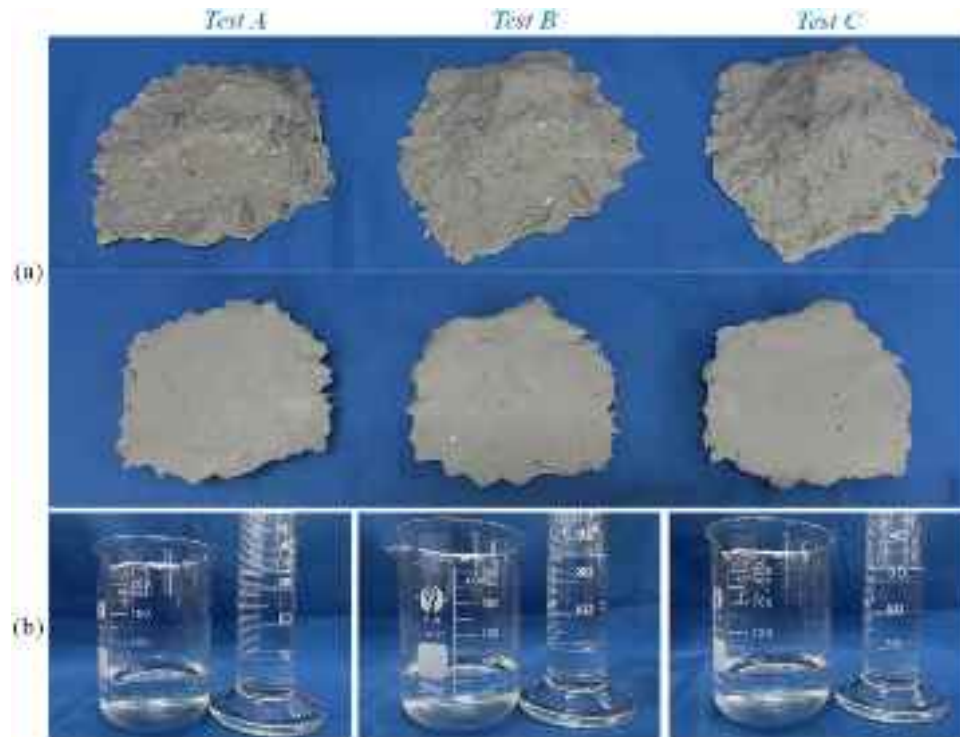
Fig. 27. Results of three repeated measurements of slab spalling instance 1: (a) oil-based clay models and (b) drainage volume measurements.
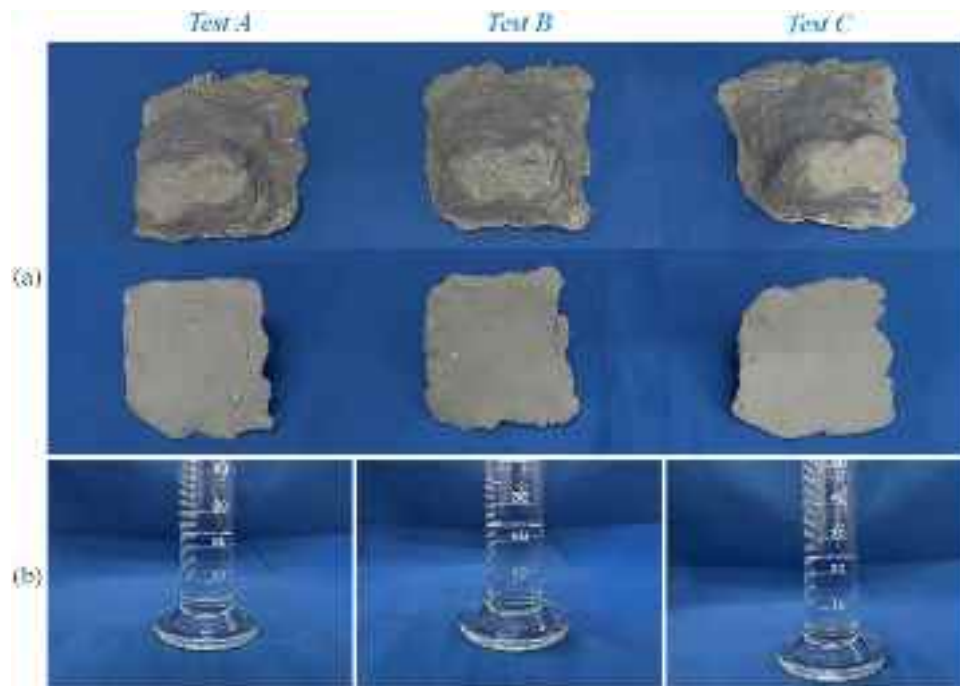

Fig. 28. Results of three repeated measurements of slab spalling instance 2: (a) oil-based clay models and (b) drainage volume measurements.

by the oil-based clay during the filling process. To reduce these experimental deviations, using a transparent modeling material could be beneficial. Throughout the process, no significant changes in density or reductions in measured volume are observed, and the clay maintains its structural integrity.

The estimated and measured volumes of these spalling cases are included in Table 9. Due to the controlled laboratory environment, the three repeated manual measurements for both slab spalling instances 1 and 2 yield consistent results with minimal experimental variation. The comparative results on slab spalling demonstrate that the proposed method maintains high accuracy even when measuring very small spalling volumes. The actual measured volume averages for both concrete pier and slab spalling instances closely match their estimated values, with a minor average absolute error of only 1.43 cm^3 . The results strongly demonstrate the accuracy and reliability of the proposed method under varying conditions, validating its effectiveness in

Table 9
Comparative results of drainage measurements and estimated volume.

		Estimated (cm^3)	Actual measurement (cm^3)			Average (cm^3)	Absolute Error (cm^3)	Relative Error (%)
	–		Test A	Test B	Test C			
Pier spalling instance		182.90	189	190	171	183.33	0.40	0.22
Slab spalling instances	1	96.59	95	96	93	94.66	1.93	2.03
	2	22.04	23	24	25	24.00	1.96	8.17
Average							1.43	3.47

accurately estimating the volume of surface or block damage with complex geometries.

5. Conclusions

To improve the automation and effectiveness of surface inspection for concrete structures, this paper proposes an advanced automatic damage detection and quantification approach by integrating image segmentation with point cloud reconstruction and processing techniques. Experimental validations demonstrate that fine and complex concrete cracks are effectively detected and quantified. The results show minor absolute errors compared to manual measurements, with mean and standard deviation values of 0.09 mm and 0.046 mm, respectively. Compared to existing methods, this approach generates high-resolution, realistic watertight surface meshes for surface or block damage detection, yielding outstanding quantification results for damage area and volume, exhibiting average relative errors remaining within 1 % and 3.5 %, respectively, and absolute errors within 2.5 cm^2 and 1.5 cm^3, respectively. Notably, the generated semantic point cloud in the intermediate step, brings clear 3D visualization and perception of surface statues, and enhances subsequent structural condition assessment. A key strength of the proposed method, compared to methods such as LiDAR or specialized scanning techniques, lies in its reliance on human-interpretable, low-cost RGB imagery. Detection results can be directly reviewed and interpreted without extensive additional processing, offering a transparent and practical solution for field inspections.

Despite the proposed approach demonstrating effectiveness and applicability for structural inspection, it has limitations that deserve further attention. The proposed damage quantification method is mainly developed for flat concrete surfaces and has not been thoroughly tested on other geometries, such as cylindrical structures. The robustness of damage extraction based on empirically determined RGB thresholds requires further validation across more test cases. Future work will focus on enhancing the versatility of methods to accommodate a wider range of structural forms and practical engineering scenarios. Additionally, intersections of crack branches often exhibit sparse or absent skeleton points due to the uneven and sparse distribution of crack points. It is advisable to develop skeletonization and point interpolation algorithms that can generate accurate and consistent crack and skeleton points, therefore enhancing crack width quantification. Owing to space constraints, comprehensive comparisons with other photogrammetric and non-photogrammetric volume measurement techniques will be presented in future studies.

CRediT authorship contribution statement

Linjie Huang: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation. **Jun Li:** Writing – review & editing, Supervision, Funding acquisition. **Gao Fan:** Writing – review & editing, Supervision, Resources, Methodology, Funding acquisition, Conceptualization. **Zhen Peng:** Writing – review & editing. **Shihong Chen:** Software. **Hong Hao:** Writing – review & editing.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

References

[1] Cha Y-J, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. ComputAided Civ Infrastruct Eng 2017;32 (5):361–78. https://doi.org/10.1111/mice.12263.

[2] Deng J, Lu Y, Lee VC-S. Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. ComputAided Civ Infrastruct Eng 2020;35(4):373–88. https://doi.org/10.1111/mice.12497.

[3] Zhang C, Chang C, Jamshidi M. Concrete bridge surface damage detection using a single-stage detector. ComputAided Civ Infrastruct Eng 2020;35(4):389–409. https://doi.org/10.1111/mice.12500.

[4] Yang X, Li H, Yu Y, Luo X, Huang T, Yang X. Automatic pixel-level crack detection and measurement using fully convolutional network. ComputAided Civ Infrastruct Eng 2018;33(12):1090–109. https://doi.org/10.1111/mice.12412.

[5] Rubio JJ, Kashiwa T, Laiteerapong T, Deng W, Nagai K, Escalera S, et al. Multi-class structural damage segmentation using fully convolutional networks. Comput Ind 2019;112:103121. https://doi.org/10.1016/j.compind.2019.08.002.

[6] Liu Z, Cao Y, Wang Y, Wang W. Computer vision-based concrete crack detection using U-net fully convolutional networks. Autom Constr 2019;104:129–39. https://doi.org/10.1016/j.autcon.2019.04.005.

[7] Bhowmick S, Nagarajaiah S, Veeraraghavan A. Vision and deep learning-based algorithms to detect and quantify cracks on concrete surfaces from UAV videos. Sensors 2020;20(21):6299. https://doi.org/10.3390/s20216299.

[8] Huang B, Zhao S, Kang F. Image-based automatic multiple-damage detection of concrete dams using region-based convolutional neural networks. J Civ Struct Health Monit 2023;13:413–29. https://doi.org/10.1007/s13349-022-00650-9.

[9] Kang D, Benipal SS, Gopal DL, Cha Y-J. Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning. Autom Constr 2020;118:103291. https://doi.org/10.1016/j.autcon.2020.103291.

[10] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, ArXiv Preprint (2021), https://doi.org/10.48550/arXiv.2103.14030.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, ArXiv Preprint (2017), https://doi.org/10.48550/arXiv.1706.03762.

[12] Ding W, Yang H, Yu K, Shu J. Crack detection and quantification for concrete structures using UAV and transformer. Autom Constr 2023;152:104929. https://doi.org/10.1016/j.autcon.2023.104929.

[13] Huang L, Fan G, Li J, Hao H. Deep learning for automated multiclass surface damage detection in bridge inspections. Autom Constr 2024;166:105601. https://doi.org/10.1016/j.autcon.2024.105601.

[14] Miao P, Srimahachota T. Cost-effective system for detection and quantification of concrete surface cracks by combination of convolutional neural network and image processing techniques. Constr Build Mater 2021;293:123549. https://doi.org/10.1016/j.conbuildmat.2021.123549.

[15] Wang W, Hu W, Wang W, Xu X, Wang M, Shi Y, et al. Automated crack severity level detection and classification for ballastless track slab using deep convolutional neural network. Autom Constr 2021;124:103484. https://doi.org/10.1016/j.autcon.2020.103484.

[16] Liu Y, Nie X, Fan J, Liu X. Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction. ComputAided Civ Infrastruct Eng 2020;35(5):511–29. https://doi.org/10.1111/mice.12501.

[17] Zhang C, Shu J, Shao Y, Zhao W. Automated generation of FE models of cracked RC beams based on 3D point clouds and 2D images. J Civ Struct Health Monit 2021: 1–18. https://doi.org/10.1007/s13349-021-00525-5.

[18] Xue Y, Cai X, Shadabfar M, Shao H, Zhang S. Deep learning-based automatic recognition of water leakage area in shield tunnel lining. Tunn Undergr Space Technol 2020;104:103524. https://doi.org/10.1016/j.tust.2020.103524.

[19] Li S, Zhao X, Zhou G. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. ComputAided Civ Infrastruct Eng 2019;34(7):616–34. https://doi.org/10.1111/mice.12433.

[20] Zhao S, Kang F, Li J. Concrete dam damage detection and localisation based on YOLOv5s-HSC and photogrammetric 3D reconstruction. Autom Constr 2022;143: 104555. https://doi.org/10.1016/j.autcon.2022.104555.

[21] Deng L, Sun T, Yang L, Cao R. Binocular video-based 3D reconstruction and length quantification of cracks in concrete structures. Autom Constr 2023;148:104743. https://doi.org/10.1016/j.autcon.2023.104743.

[22] Son H, Bosché F, Kim C. As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: a survey. Adv Eng Inform 2015;29(2):172–83. https://doi.org/10.1016/j.aei.2015.01.009.

[23] Hou T-C, Liu J-W, Liu Y-W. Algorithmic clustering of LiDAR point cloud data for textural damage identifications of structural elements. Measurement 2017;108: 77–90. https://doi.org/10.1016/j.measurement.2017.05.032.

[24] Kim M-K, Sohn H, Chang C-C. Localization and quantification of concrete spalling defects using terrestrial laser scanning. J Comput Civ Eng 2015;29(6):04014086. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000415.

[25] Guldur B, Hajjar JF. Damage detection on structures using texture mapped laser point clouds. Structures Congress 2014. American Society of Civil Engineers; 2014. p. 324–34. https://doi.org/10.1061/9780784413357.030.

[26] Olsen MJ, Kuester F, Chang BJ, Hutchinson TC. Terrestrial laser scanning-based structural damage assessment. J Comput Civ Eng 2010;24(3):264–72. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000028.

[27] Kim H, Yoon J, Hong J, Sim S-H. Automated damage localization and quantification in concrete bridges using point cloud-based surface-fitting strategy. J Comput Civ Eng 2021;35(6):04021028. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000993.

[28] Yang M-D, Chao C-F, Huang K-S, Lu L-Y, Chen Y-P. Image-based 3D scene reconstruction and exploration in augmented reality. Autom Constr 2013;33: 48–60. https://doi.org/10.1016/j.autcon.2012.09.017.

[29] Chen S, Fan G, Li J. Improving completeness and accuracy of 3D point clouds by using deep learning for applications of digital twins to civil structures. Adv Eng Inform 2023;58:102196. https://doi.org/10.1016/j.aei.2023.102196.

[30] Colomina I, Molina P. Unmanned aerial systems for photogrammetry and remote sensing: a review. ISPRS J Photogramm Remote Sens 2014;92:79–97. https://doi.org/10.1016/j.isprsjprs.2014.02.013.

[31] Bayram B, Nemli G, Özkan T, Oflaz OE, Kankotan B, Çetin İ. Comparison of laser scanning and photogrammetry and their use for digital recording of cultural monument case study: byzantine land walls-istanbul. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 2015;II-5/W3:17–24. https://doi.org/10.5194/isprsannals-II-5-W3-17-2015.

[32] Torok MM, Golparvar-Fard M, Kochersberger KB. Image-based automated 3D crack detection for post-disaster building assessment. J Comput Civ Eng 2014;28(5): A4014004. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000334.

[33] Khaloo A, Lattanzi D, Cunningham K, Dell'Andrea R, Riley M. Unmanned aerial vehicle inspection of the placer river trail bridge through image-based 3D modelling. Struct Infrastruct Eng 2018;14(1):124–36. https://doi.org/10.1080/15732479.2017.1330891.

[34] Gillins MN, Gillins DT, Parrish C. Cost-effective bridge safety inspections using unmanned aircraft systems (UAS). Geotech Struct Eng Congr 2016 2016:1931–40. https://doi.org/10.1061/9780784479742.165.

[35] Kim I-H, Jeon H, Baek S-C, Hong W-H, Jung H-J. Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle. Sensors 2018;18(6):1881. https://doi.org/10.3390/s18061881.

[36] Schonberger JL, Frahm J-M. Structure-from-motion revisited. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2016. p. 4104–13. https://doi.org/10.1109/CVPR.2016.445.

[37] McLaughlin E, Charron N, Narasimhan S. Automated defect quantification in concrete bridges using robotics and deep learning. J Comput Civ Eng 2020;34(5): 04020029. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000915.

[38] Fernandez Galarreta J, Kerle N, Gerke M. UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning. Nat Hazards Earth Syst Sci 2015;15(6):1087–101. https://doi.org/10.5194/nhess-15-1087-2015.

[39] Chaiyasarn K, Buatik A, Likitlersuang S. Concrete crack detection and 3D mapping by integrated convolutional neural networks architecture. Adv Struct Eng 2021;24 (7):1480–94. https://doi.org/10.1177/1369433220975574.

[40] Kalfarisi R, Wu ZY, Soh K. Crack detection and segmentation using deep learning with 3D reality mesh model for quantitative assessment and integrated visualization. J Comput Civ Eng 2020;34(3):04020010. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000890.

[41] Zeng Q, Fan G, Wang D, Tao W, Liu A. A systematic approach to pixel-level crack detection and localization with a feature fusion attention network and 3D reconstruction. Eng Struct 2024;300:117219. https://doi.org/10.1016/j.engstruct.2023.117219.

[42] Chen S, Fan G, Li J, Hao H. Automatic complex concrete crack detection and quantification based on point clouds and deep learning. Eng Struct 2025;327: 119635. https://doi.org/10.1016/j.engstruct.2025.119635.

[43] Yuan C, Xiong B, Li X, Sang X, Kong Q. A novel intelligent inspection robot with deep stereo vision for three-dimensional concrete damage detection and quantification. Struct Health Monit 2022;21(3):788–802. https://doi.org/10.1177/14759217211010238.

[44] Schönberger JL, Zheng E, Frahm J-M, Pollefeys M. Pixelwise view selection for unstructured multi-view stereo. In: Computer Vision – ECCV 2016. Springer; 2016. p. 501–18. https://doi.org/10.1007/978-3-319-46487-9_31.

[45] Omar T, Nehdi ML. Remote sensing of concrete bridge decks using unmanned aerial vehicle infrared thermography. Autom Constr 2017;83:360–71. https://doi.org/10.1016/j.autcon.2017.06.024.

[46] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J.M. Alvarez, P. Luo, SegFormer: Simple and efficient design for semantic segmentation with transformers, ArXiv Preprint (2021), https://doi.org/10.48550/arXiv.2105.15203.

[47] Umeyama S. Least-squares estimation of transformation parameters between two point patterns. IEEE Trans Pattern Anal Mach Intell 1991;13(4):376–80. https://doi.org/10.1109/34.88573.

[48] Q.-Y. Zhou, J. Park, V. Koltun, Open3D: A modern library for 3D data processing, ArXiv Preprint (2018), https://doi.org/10.48550/arXiv.1801.09847.

[49] Ankerst M, Breunig MM, Kriegel H-P, Sander J. OPTICS: ordering points to identify the clustering structure. ACM Sigmod Rec 1999;28(2):49–60. https://doi.org/10.1145/304181.304187.

[50] Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 1981;24(6):381–95. https://doi.org/10.1145/358669.358692.

[51] Cao J, Tagliasacchi A, Olson M, Zhang H, Su Z. Point cloud skeletons via laplacian based contraction. In: 2010 Shape Modeling International Conference. IEEE; 2010. p. 187–97. https://doi.org/10.1109/SMI.2010.25.

[52] Kruskal JB. On the shortest spanning subtree of a graph and the traveling salesman problem. Proc Am Math Soc 1956;7(1):48–50. https://doi.org/10.2307/2033241.

[53] Kazhdan M, Hoppe H. Screened poisson surface reconstruction. ACM Trans Graph 2013;32(3):1–13. https://doi.org/10.1145/2487228.2487237.

[54] Wang J, Oliveira MM. Filling holes on locally smooth surfaces reconstructed from point clouds. Image Vis Comput 2007;25(1):103–13. https://doi.org/10.1016/j.imavis.2005.12.006.

[55] A. Muntoni, P. Cignoni, PyMeshLab, (2021), https://doi.org/10.5281/zenodo.4438751.

[56] Bernardini F, Mittleman J, Rushmeier H, Silva C, Taubin G. The ball-pivoting algorithm for surface reconstruction. IEEE Trans Vis Comput Graph 1999;5(4): 349–59. https://doi.org/10.1109/2945.817351.

[57] Edelsbrunner H, Kirkpatrick D, Seidel R. On the shape of a set of points in the plane. IEEE Trans Inf Theory 1983;29(4):551–9. https://doi.org/10.1109/TIT.1983.1056714.

[58] Beckman GH, Polyzois D, Cha Y-J. Deep learning-based automatic volumetric damage quantification using depth camera. Autom Constr 2019;99:114–24. https://doi.org/10.1016/j.autcon.2018.12.006.