Automation in Construction 157 (2024) 105168

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Damage volumetric assessment and digital twin synchronization based on LiDAR point clouds

Yan Gao[a], Haijiang Li[a,*], Weiqi Fu[b], Chengzhang Chai[a], Tengxiang Su[a]

[a] BIM for Smart Engineering Centre, School of Engineering, Cardiff University, Cardiff CF24 3AA, UK
[b] Aston Business School, Aston University, Birmingham B4 7ET, UK

ARTICLE INFO

Keywords:
Point cloud
Digital twin synchronization
Voxelization
Binarization
Damage volumetric assessment
Finite-element modelling
Building information modelling

ABSTRACT

Point clouds are widely used for structure inspection and can provide damage spatial information. However, how to update a digital twin (DT) with local damage based on point clouds has not been sufficiently studied. This research presents an efficient framework for assessing and DT synchronizing local damage on a planar surface using point clouds. The pipeline starts from damage detection via DeepLabV3+ on the pseudo grayscale images from the point depth. It avoids the drawbacks of image and point cloud fusion. The target point cloud is separated according to the detected damage. Then, it can be converted into a 3D binary matrix through voxelization and binarization, which is highly lightweight and can be losslessly compressed for DT synchronization. The framework is validated via two case studies, demonstrating that the proposed voxel-based method can be easily applied to real-world damage with non-convex geometry instead of convex-hull fitting; finite-element (FE) models and BIM models can be updated automatically through the framework.

1. Introduction

Vision-based non-destructive inspection has been widely used for structural health monitoring (SHM), including 2D images and 3D point clouds. Many image-based approaches [1–3] were developed for structural surface damage detection, but 2D images cannot provide the damage spatial information for assessment and as-is model updating. Hence, the 3D point cloud data is utilized to solve this issue by offering depth information about the damage [1–3], which can be generated from photogrammetry, depth cameras, binocular cameras, terrestrial laser scanning (TLS), etc. However, the 3D survey is usually project-based with voluminous data. The post-processing of the obtained point cloud is also far from automated, which cannot achieve timely 3D damage assessment and digital twin (DT) synchronization, as well as provide feedback and decision-making to the physical entity in time. Little research has focused on efficient and automatic assessment for 3D local damage and as-is model updating during the survey. The only related one [4] is an approach for spalling detection by fusing point clouds and images derived from the iPhone LiDAR and camera. However, it requires extra photos for damage detection and complex coordinate transformation for data fusion, which can bring in potential errors. Meanwhile, its method for damage volumetric assessment is based on convex-hull fitting, requiring elaborate manual separation for each convex component, which is not always practical because real-world structural damage usually has complicated non-convex geometry. Furthermore, few previous efforts have been made in efficient data transmission for DT synchronization based on the point cloud. Therefore, it would be helpful to develop an efficient framework for 3D damage assessment, data transmission, and as-is model updating with local damage based on the point cloud during the survey.

This work aims at vertical 3D damage on a planar surface, such as spalling and cracking, and focuses on the post-processing of the point cloud, which can provide sufficient spatial damage information. The proposed framework includes surface damage detection, spatial segmentation, data transmission, and DT synchronization. Firstly, surface damage detection is achieved via a pre-trained DeepLabV3+ model on the pseudo grayscale image derived from the point cloud depth information, avoiding the drawbacks of image and point-cloud fusion. Here, the DeepLabV3+ model is trained on real grayscale images from a public image set, including damage and non-damage images, which can distinguish natural damage from legitimate concave patterns. An experiment is conducted on the specimens with different crack widths and depths, demonstrating that the trained DeepLabV3+ model can be applied to pseudo grayscale images when damage satisfies certain

[*] Corresponding author.
E-mail address: lih@cardiff.ac.uk (H. Li).

https://doi.org/10.1016/j.autcon.2023.105168
Received 3 August 2023; Received in revised form 15 October 2023; Accepted 31 October 2023
Available online 11 November 2023

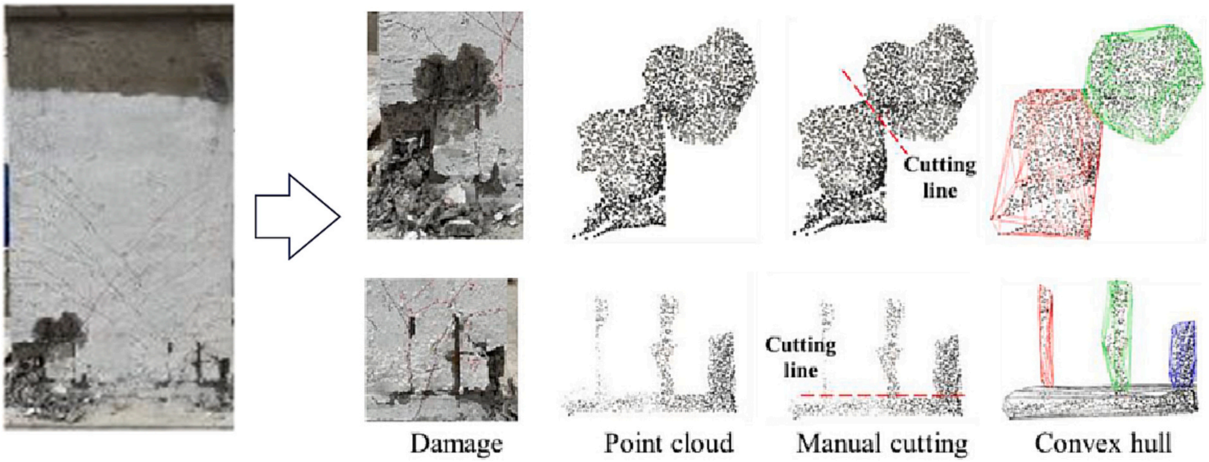


Fig. 1. Convex-hull fitting for damage quantification based on manual cutting [4].

conditions. Then the generated damage bounding box and mask are utilized for damage spatial segmentation through morphological operation from the surface until the maximum depth of the damage. After voxelization with appropriate resolution, the target point cloud can be converted into a highly lightweight 3D binary matrix representing the damage spatial geometry through voxelization and binarization, which can be further compressed through lossless running-length encoding (RLE) for efficient transmission.

The transmitted 3D binary matrix can be utilized for damage volumetric assessment and DT model updating with the local damage, including finite-element (FE) and BIM models. The proposed voxel-based fitting method for volumetric evaluation can be achieved by summing up the elements of the 3D binary matrix (i.e., empty voxel – 1, occupied voxel – 0). The method can be easily applied on real-world structural damage with complicated non-convex geometry instead of convex-hull fitting [4], and it does not require elaborate manual cutting for each component. Moreover, the FE geometric model can be updated automatically by deleting the corresponding void elements within the damage space, and the element ID list can be obtained by querying if the matrix element equals one. Moreover, as each matrix layer can be taken as a binary image, the damage contour in each layer can be fitted using an appropriate shape. Then, the damage geometry can be reconstructed by meshing through the contours across different layers and utilized for BIM model updating. Finally, the framework is validated via two case studies, including the synthetic groove on a specimen and the real-world building crack.

This study mainly has four contributions:

1) This study proposes an efficient framework for assessing and DT synchronizing 3D local damage using point clouds. After surface damage detection based on deep learning (DL), the bounding box and mask can be utilized for damage spatial segmentation to remove the redundant point cloud. The separated point cloud can be converted into a highly lightweight 3D binary matrix representing the damage spatial geometry, which can be further losslessly compressed for DT synchronization.
2) Surface damage detection is achieved through a DeepLabV3+ model on pseudo grayscale images from the point depth. It avoids the drawbacks of image and point cloud fusion. The approach is validated on specimens with variant crack widths and depths. The experiment demonstrates that the DL model trained from real grayscale images can be applied to pseudo grayscale images for damage detection when the damage satisfies appropriate conditions. Meanwhile, the model can distinguish natural damage from manual concave patterns by training on the annotated damage and non-damage images.
3) The proposed voxel-based fitting method for damage volumetric assessment, which does not require elaborate manual cutting for each component, can be easily applied to 3D damage with complicated non-convex geometry instead of the convex-hull fitting in previous research.
4) FE and BIM geometric models can be updated automatically with the local damage based on the transmitted 3D binary matrix.

The rest of this paper is structured as follows: Section 2 overviews the related work for crack detection, point-cloud processing, and model synchronization. Section 3 presents the proposed framework. Section 4 is the framework validation via case studies. Section 5 provides a dedicated discussion. Section 6 concludes the work.

2. Literature review

2.1. Damage detection and characterization

Research for vision-based damage detection and characterization has developed rapidly in recent years by leveraging deep learning. As photographs and videos are the most common visual data for inspection, many image-based object detection and semantic segmentation approaches based on deep neural networks (DNNs), such as YOLO [5], SSD [6], R-CNN [7], DeepLab [8], have been successfully applied to automatic visual inspection for buildings [9], bridges [6,8], roads [5,10], and many other infrastructures [6,11]. In addition to general visual inspection, these studies can use different technologies to extract the 2D damage characters for structural assessment and maintenance decision-making, such as damage types, locations, areas, lengths, widths, and directions. For example, with the pre-calibrated camera and distance sensor, the length and width of concrete cracking can be obtained at 1 mm accuracy within the estimation error of <10% [12]. Moreover, by integrating GNSS and IMU parameters, the detected crack can be localized in the same coordinate system of the infrastructure model during drone inspection at an accuracy of just a few centimetres [13].

Although image-based damage detection and characterization have significantly progressed, they cannot provide the necessary depth information for 3D damage assessment and geometric model updating. The high-resolution point cloud, derived from photogrammetry, depth cameras, binocular cameras, terrestrial laser scanning (TLS), etc., is expected to fill this gap. For example, spalling and crack can be detected using the luminance or depth variation of RGB point-cloud data from TLS [14,15] and then characterized approximately with the maximum depth. However, these approaches cannot distinguish between natural 3D damage and legitimate concave patterns, such as handcrafted circles, brick joints, and decorative textures. Hence, the latest research proposed

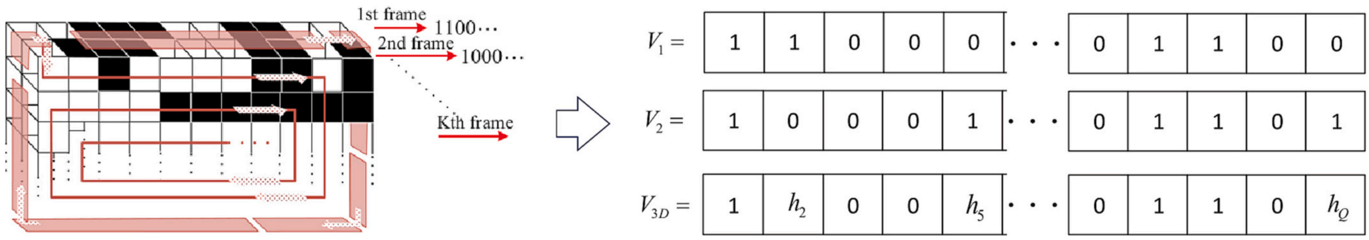

1st frame
1100···
2nd frame
1000···
Kth frame
$V_1 =$ 1 1 0 0 0 · · · 0 1 1 0 0
$V_2 =$ 1 0 0 0 1 · · · 0 1 1 0 1
$V_{3D} =$ 1 h_2 0 0 h_5 · · · 0 1 1 0 h_Q

Fig. 2. Compression with 3D run-length encoding [26].

a method based on mask R-CNN to achieve damage detection and segmentation. The mask geo-reference is achieved by the fusion of the image and point cloud [4]. This method can automatically distinguish target damage from the designed concaves. Still, it requires extra photo acquisition and complex coordinate transformation [16], which will bring in potential errors, especially when the image and the point cloud are obtained under different conditions (i.e., from different angles and distances). Therefore, developing an approach for surface damage detection and segmentation entirely based on the point cloud would be helpful.

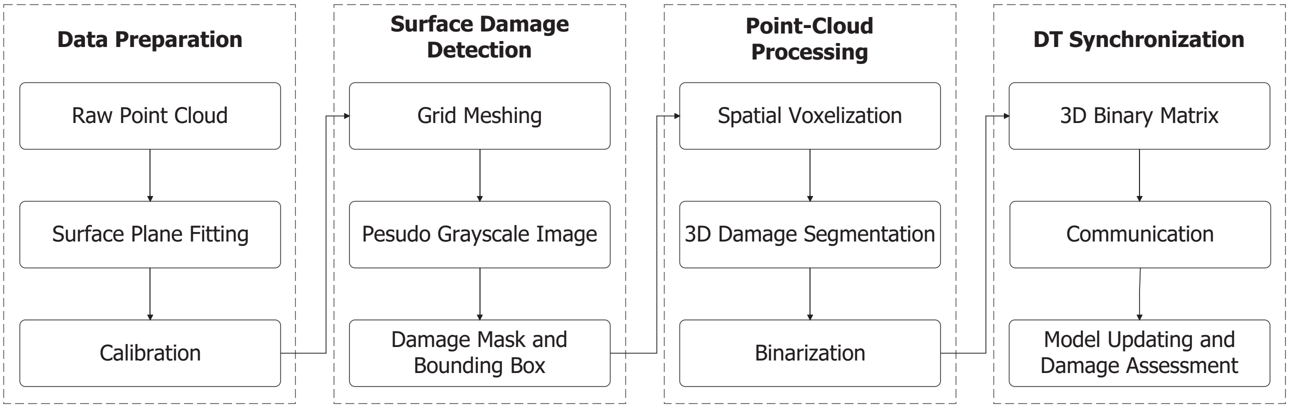
2.2. Data processing and damage quantification

Point-cloud data processing approaches can be categorized into point-based and voxel-based. Previous spatial damage detection and assessment studies mainly belong to the former, which compare the point cloud with the ideal 3D model or the previous captures. For example, the point cloud of a damaged reinforced concrete (RC) column after seismic testing can be sliced into different layers and compared with a rectangle to distinguish the spalling and residual areas [17]. The point cloud captures of each component at different times can be compared with each other to identify damage and monitor its evolution [18]. Moreover, the damage volumetric quantification can be achieved with convex-hull fitting [4] based on the point cloud. However, it is prone to exaggerate the ground truth of the target volume for non-convex geometry, so it requires elaborate manual separation for each component, as the dash lines shown in Fig. 1, which is not always practical for real-world damage with complicated geometry. The point-based approaches lose sight of the benefit of voxels for volume calculation and mitigation of point sparsity. Therefore, developing a voxel-based point-cloud processing approach for 3D damage detection and assessment would be helpful.

2.3. Model synchronization and data compression

The infrastructure DT for structure health monitoring (SHM) and maintenance is not only about 3D visualization but also involves communication and back-end services, such as FEA and BIM, as well as feedback to the physical entity. From the practitioners' view, one obstacle to DT application in practice is the difficulty of keeping model synchronization automatically in routine inspection [19]. The current 3D scanning survey for local damage is still far from automated, and the DT models cannot be updated in time, so developing an approach for the "as-is" model updating with the detected local damage during the survey would improve the work efficiency. One of the bottlenecks is the communication complexity, i.e., the massive volume of point cloud data. It is a heavy load for data transmission in both time and cost, especially for some infrastructures under circumstances with limited communication, such as bridges and tunnels. Therefore, it is necessary to only transmit the damage part in an efficient format instead of the raw point cloud.

Additionally, compression can be utilized to decrease communication complexity further. For example, previous studies proposed a few approaches to compress 3D point clouds using RNN with residual blocks [20] or a hierarchical auto encoder [21]. However, the transmitted data for DT synchronization is not necessarily a point cloud. It can be any format that distinguishes the 3D damaged space from the residual entity. Therefore, binarised voxels have become a promising way to reduce complexity significantly, and lossless run-length encoding (RLE) can be utilized for further compression. In the medical field, the 3D binary matrix through the volumetric RLE has already been proven successfully as an efficient approach for the transmission of 3D medical data, as shown in Fig. 2. Moreover, the transmitted binarised voxels can be employed to update the FE and BIM models. The latest research [4] has demonstrated the effectiveness of updating the FE model by deleting the corresponding elements less than the damage depth. The method is based on iteration and can be integrated into mainstream FE software programs. Moreover, the previous studies [22–25] have revealed the availability and workflow for damage modelling in geometry and semantics using IFC files.


Data Preparation
Raw Point Cloud
Surface Plane Fitting
Calibration

Surface Damage Detection
Grid Meshing
Pesudo Grayscale Image
Damage Mask and Bounding Box

Point-Cloud Processing
Spatial Voxelization
3D Damage Segmentation
Binarization

DT Synchronization
3D Binary Matrix
Communication
Model Updating and Damage Assessment

Fig. 3. Proposed framework for 3D damage assessment and DT synchronization.

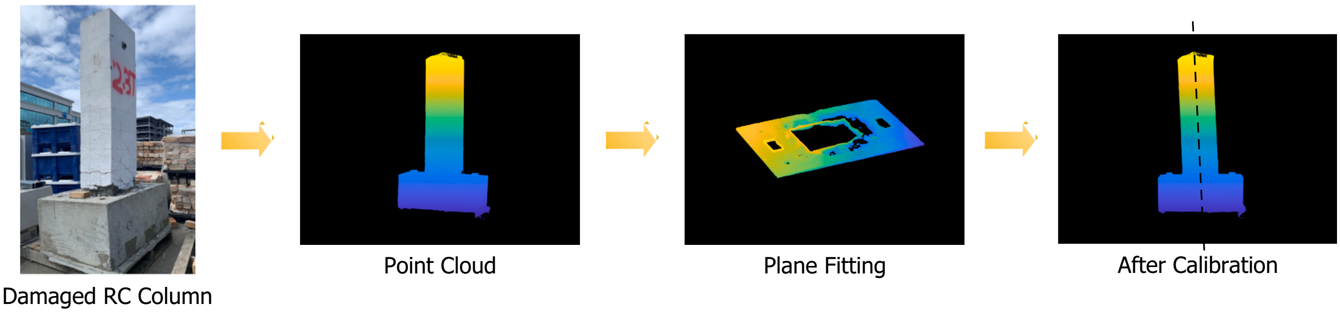


Fig. 4. Plane fitting and calibration of point cloud to check damaged column verticality.

3. Proposed framework

3.1. Problem statement

This research aims to propose an efficient framework to achieve spatial damage assessment and DT synchronization based on the 3D point cloud. The work focuses on point-cloud post-processing of vertical damages on a planar surface, such as spalling or relatively wide cracks. Hence, this work does not consider insufficient measurement situations due to technology limits, such as TLS for diagonal cracking or narrow cracks, which cannot provide sufficient damage face and depth information due to occlusion. In detail, the research problems are shown below.

1. The existing surface damage detection relies on image and point cloud fusion. It requires extra photo acquisition, and if the photo cannot be taken under the same condition as the survey, i.e., from the same angle and distance, it will lead to complex coordinate transformation and potential errors. Therefore, developing an approach for surface damage detection entirely based on the point cloud is necessary.
2. After detecting the surface damage area, an efficient point-cloud processing method is required for damage spatial segmentation. Meanwhile, the generated result, representing the 3D damage geometry, should be lightweight enough for data transmission.
3. 3D damage assessment and DT model synchronization, such as FE and BIM models, must be achieved automatically from the transmitted data.

3.2. Overall design

The overall design of the proposed framework is shown in Fig. 3, including data preparation, surface damage detection, point cloud processing, and DT synchronization. In data preparation, the target planar surface in the raw point cloud is calibrated through normal lines and rotation matrices. The surface damage detection is achieved through the state-of-the-art semantic segmentation model (i.e., DeepLabV3+) on a pseudo grayscale image from the point depth. The point cloud processing for damage spatial segmentation is achieved through voxelization and binarization. The result representing the spatial damage geometry is a highly lightweight 3D binary matrix and can be losslessly compressed for data transmission. Finally, the DT model synchronization (i.e., FE and BIM models) and damage volumetric assessment can be achieved automatically from the transmitted 3D binary matrix.

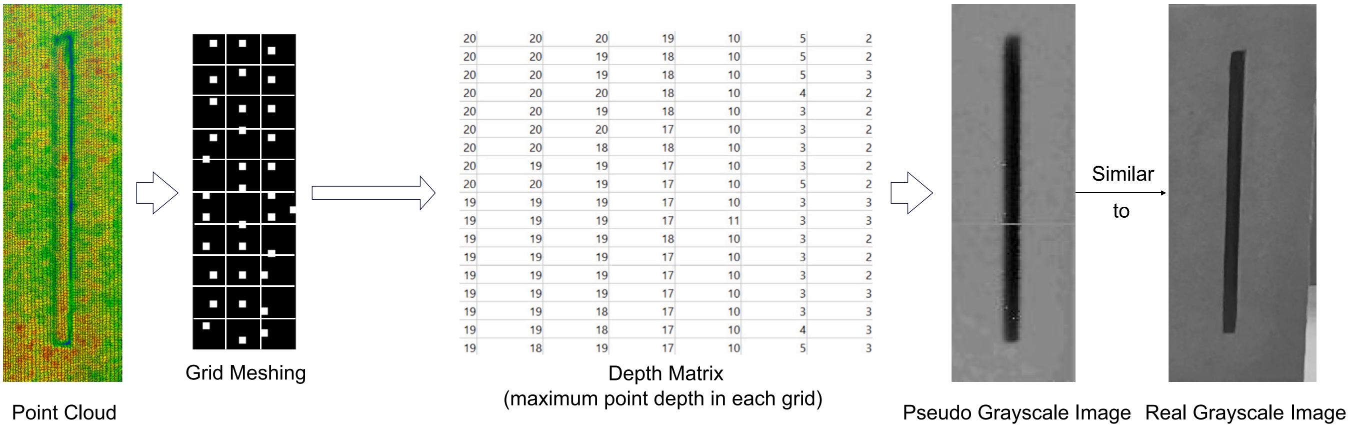
3.3. Data preparation

In the beginning, the target surface plane in the point cloud can be fitted using the M-estimator Sample Consensus (MSAC) algorithm by finding a plane that has a maximum allowable distance from an inlier to it [27]. The MSAC algorithm is a variant of the Random Sample Consensus (RANSAC) algorithm, which can partially compensate for the undesirable effect in noise threshold selection [28]. Then, the point cloud can be calibrated by adjusting the surface normal line perpendicularly to the horizontal plane, i.e., parallel to the z-axis drawn vertically in the coordinate system. The calibration can be achieved by multiplying the rotation matrix M, indicated in Eq. (1).

$$M = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix} \tag{1}$$

Where β is the pitch angle between the surface normal line and the z-axis.

This process can be illustrated by checking the verticality of a damaged RC column based on point clouds, as shown in Fig. 4. The



20	20	20	19	10	5	2
20	20	19	18	10	5	2
20	20	19	18	10	5	3
20	20	20	18	10	4	2
20	20	19	18	10	3	2
20	20	20	17	10	3	2
20	20	18	18	10	3	2
20	19	19	17	10	3	2
20	20	19	17	10	5	2
19	19	19	17	10	3	3
19	19	19	17	11	3	3
19	19	19	18	10	3	2
19	19	19	17	10	3	2
19	19	19	17	10	3	2
19	19	19	17	10	3	3
19	19	18	17	10	3	3
19	19	18	17	10	4	3
19	18	19	17	10	5	3

Fig. 5. Pseudo grayscale images derived from point-cloud depth information.

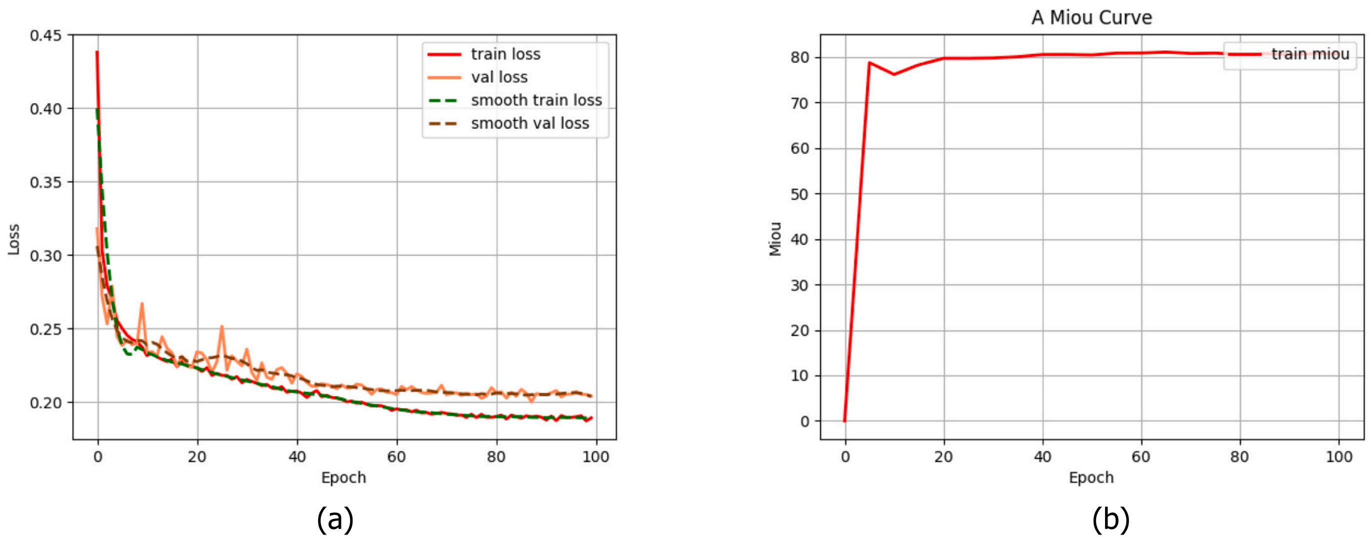


Fig. 6. (a) training loss and validation loss; (b) training MIoU.

column base upper surface is fitted using the MSAC algorithm and calibrated through the above rotation matrix. It is worth noting that the normal points on the target surface should be sufficiently greater than the damage face points, which is required for surface plane fitting.

3.4. Surface damage detection and segmentation

3.4.1. Depth-to-grayscale transformation

In previous research [4], surface damage detection on a point cloud is achieved through image processing or deep learning on 2D images (usually RGB) and fusion with the point cloud. It requires extra photo acquisition with a built-in camera. If the photo cannot be taken under the same condition as the survey, i.e., from the same angle and distance, it will lead to complex coordinate transformation and potential errors. This work aims to solve this issue by achieving surface damage detection with pseudo grayscale images entirely based on the point-cloud depth information.

Assuming the deeper spot in the damaged space has lower luminance (i.e., lower grayscale level), the pseudo grayscale images can be generated through the pipeline shown in Fig. 5. Here, the point cloud for the manual groove on a specimen is taken as an example. The resolution in grid meshing is considered consistent for the following voxelization and downstream tasks so that it can be determined by damage conditions and assessment requirements. For example, the assessment for the building crack in the maintenance manuals [29–31] is at the millimetre level, so the survey for cracking is usually achieved with high-resolution TLS scanning, and the grid resolution is set as 1 mm. In contrast, the assessment for spalling, which is much broader than crack with more relaxed inspection standards, can be achieved using the iPhone LiDAR under 1 cm resolution for both volumetric quantification and FE model updating in the previous research [4].

Initially, grid meshing is applied on the point cloud to tighten the point sparsity. As seen in Fig. 6, multiple points can be in a single grid, and the depth matrix is generated with the maximum point depth in the grid. Notably, this process will not lead to the loss of information related to the crack. In contrast, it will enhance the impact of the maximum depth in the grid because the deepest point represents the depth of a single grid.

Then, the depth-to-grayscale transformation for pseudo grayscale images is achieved through Eqs. (2)–(4). d is the depth value for each element in the depth matrix; d_{th} is the depth threshold for normalization (i.e., $d = d_{th}, \forall d > d_{th}$); I_{mean} is the average grayscale level of the training image set I; D_{gray} is the generated grayscale level for each pixel in a pseudo grayscale image. Moreover, the ratio k of the minimum damage depth d'_{min} to the depth threshold d_{th} is an essential indicator for successful transformation, indicated in Eq. (5).

$$D = 1 - \frac{d}{d_{th}} \tag{2}$$

$$K = \frac{I_{mean}}{D_{mean}} \tag{3}$$

$$D_{gray} = K \cdot D \tag{4}$$

$$k = \frac{d'_{min}}{d_{th}} \tag{5}$$

Here, a realistic damage image set I [32] is utilized to train a DNN model for surface damage detection on the real grayscale images and then apply the model on the pseudo grayscale images. The pseudo images are scaled to the same average grayscale level of the damage image set I (i.e., I_{mean}), which is beneficial for the model's performance. As the pseudo grayscale images are entirely generated from the point-cloud depth information, this method overcomes the angle and distance difference between the point cloud and the photo (see Fig. 6), thereby avoiding complex coordinate transformation and potential errors within data fusion.

3.4.2. DeepLabV3+ model

Surface damage detection on the generated pseudo grayscale images can be achieved via image processing such as OTSU's method [33] through exhaustively searching the optimal threshold to maximize interclass variance (Eq. (6)) based on grayscale. The result can reflect the depth difference between the surface and the damaged areas. However, it cannot distinguish between natural damage and legitimate concave patterns, such as handcrafted holes, brick joints, and decorative textures. Hence, deep learning is utilized to solve this issue for surface damage detection.

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \tag{6}$$

Where, ω_0 and ω_1 are the probabilities of the two classes (i.e., background and damage areas) separated by a threshold t; σ_0^2 and σ_1^2 are variances of these two classes.

This work uses the state-of-the-art semantic segmentation model DeepLabV3+, which combines the Atrous Spatial Pyramid Pooling benefits and the Encoder-Decoder architecture for surface damage detection and segmentation on grayscale images. Here, the supervised learning approach for crack detection is taken as an example and can be

Table 1
Model training configuration.

Architecture	Input	Split	Total Epochs	Batch Size	Learning Rate
DeepLabV3+	448 × 448	80%:10%:10%	100	8	0.00005

easily extended to detect other surface damages (such as spalling) by using the corresponding annotated images to train the model.

The DeepLabV3+ model is trained on a public crack image set [32], which includes 9584 crack and 1411 non-crack images. The images are all transformed into grayscale images and split into the training, validation, and test sets (i.e., 80%:10%10%). The pre-trained MobileNet-v2 based on the PASCAL VOC dataset [34] is employed as the backbone for feature extraction. The training condition is shown in Table 1. The training loss and MIoU are presented in Fig. 7. The performance on the test set is evaluated using mean Intersection over Union (MIoU) and mean Pixel Accuracy (MPA), which are 80% and 88%, respectively. It demonstrates that the trained DeepLabV3+ has excellent performance for crack detection and segmentation on real grayscale images. Moreover, the model can distinguish between natural cracks and legitimate concave patterns such as handcrafted holes and brick joints.

To test the trained model performance on the pseudo grayscale images, four specimens created in the lab with variant crack widths and depths are utilized in this work, as shown in Fig. 7. The RealSense LiDAR Camera L515 is used for crack scanning to generate point clouds. The real RGB images are derived from the RGB information for each point using the built-in camera, and its grayscale images are calculated through Eq. (7). The pseudo grayscale images are generated through the pipeline in Fig. 6 with a resolution of 1 mm and $d_{th} = d_{max}$.

$$Gray = 0.2989{*}R + 0.5870{*}G + 0.1140{*}B \tag{7}$$

The proposed approach is tested by comparing the model performance on the pseudo grayscale images with its performance on the real grayscale images, and the latter is taken as the ground truth for segmentation. The model performance is evaluated using the MIoU (Eq. (8)) and the MPA (Eq. (9)) for binary segmentation, i.e., crack (positive) and background (negative) pixels. IoU_{pos} and IoU_{neg} denote positive and negative Intersection of Union; P_{pos} and P_{neg} represent positive and negative precision; TP, FP, TN, and FN denote true positive, false positive, true negative, and false negative pixels, respectively.

$$MIoU = \frac{IoU_{pos} + IoU_{neg}}{2} = \frac{\frac{TP}{TP+FP+FN} + \frac{FN}{TN+FN+FP}}{2} \tag{8}$$

$$MPA = \frac{P_{pos} + P_{neg}}{2} = \frac{\frac{TP}{TP+FP} + \frac{TN}{TN+\mathrm{F}N}}{2} \tag{9}$$

The segmentation results through the trained DeepLabV3+ model on both real and pseudo grayscale images are demonstrated in Fig. 7. Here, the crack depth is the vertical depth detected by LiDAR. As can be seen, in the experiment for the cracking with width $>$ 5 mm and depth $>$ 6 mm, the trained DeepLabV3+ model can achieve excellent crack detection and segmentation performance (i.e., $MIoU$ = 84.60%, MPA = 97.20%) on the pseudo grayscale images, which are entirely derived from the point depth. The failure on the fourth beam is because the crack is too narrow, and the LiDAR cannot obtain sufficient points within the cracking space due to occlusion. Notably, although this approach is illustrated with crack detection and segmentation, it is also available to detect other volumetric damages on a planar surface, such as spalling.

The experiment demonstrates that when structural damages satisfy certain conditions (e.g., cracking width $>$ 5 mm and $k > 0.17$ in the experiment), a pre-trained DNN model from the real grayscale damage images with annotation can achieve excellent performance for surface damage detection and segmentation on the pseudo grayscale images, which are entirely derived from the point depth with appropriate resolution. This approach avoids the drawbacks of image and point cloud fusion, such as extra photo acquisition, complex coordinate transformation, and potential data fusion errors. Notably, the resolution selection is affected by the survey equipment, such as the 1 cm resolution with iPhone LiDAR in the previous research [4]. The model performance may degrade when using a relatively low resolution to generate pseudo

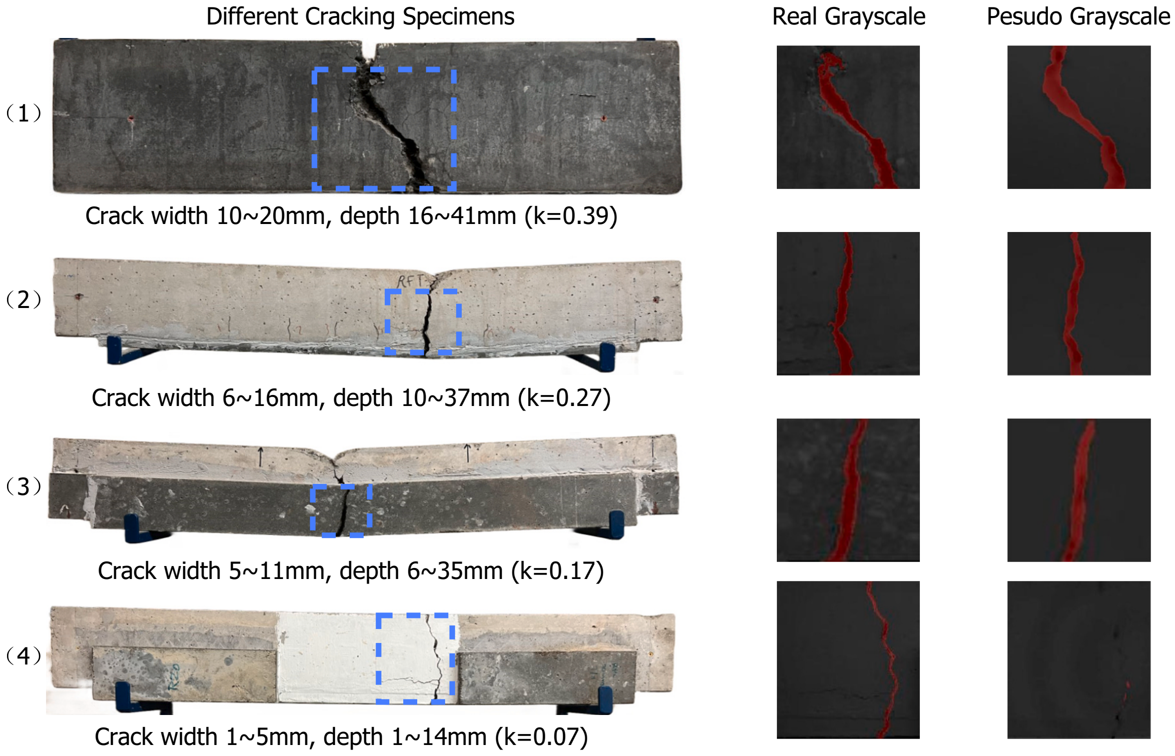

Different Cracking Specimens
Real Grayscale
Pesudo Grayscale
(1)
Crack width 10~20mm, depth 16~41mm (k=0.39)
(2)
Crack width 6~16mm, depth 10~37mm (k=0.27)
(3)
Crack width 5~11mm, depth 6~35mm (k=0.17)
(4)
Crack width 1~5mm, depth 1~14mm (k=0.07)

Fig. 7. Damage segmentation using DeepLabV3+ model on the pseudo grayscale images.

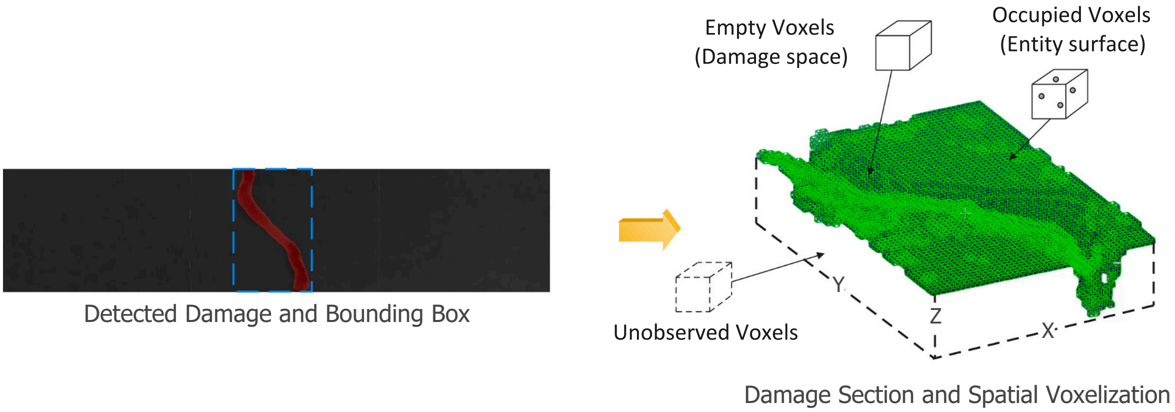


Fig. 8. Separated cuboid damage section and spatial voxelization.

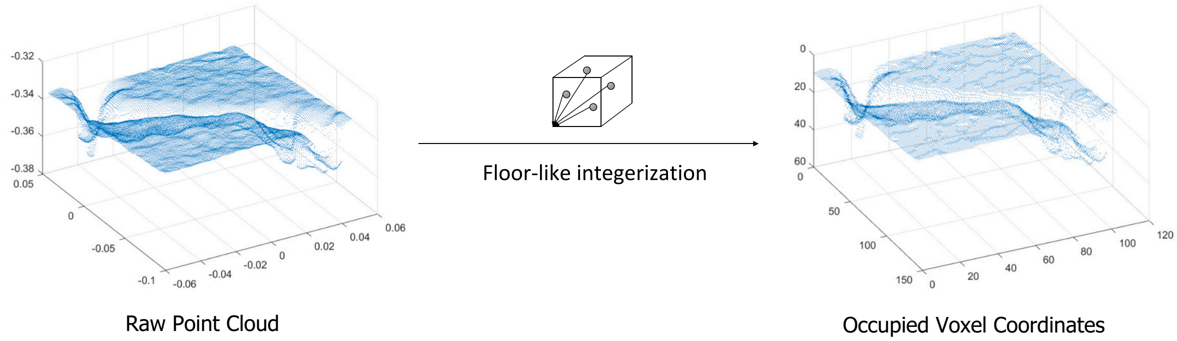


Fig. 9. Voxelization through floor-like integerization.

grayscale images. At that moment, a new model would be required based on the reduced training images with the corresponding resolution. Additionally, the model can distinguish natural structural damages from legitimate concave patterns by training through the annotated damage and non-damage images.

3.5. *Damage spatial segmentation*

3.5.1. *Spatial voxelization*

After surface damage detection and segmentation, the damage can be masked with a bounding box. Then, the point cloud section involving target damage can be separated as a cuboid according to the bounding box from the surface until the maximum depth along the depth direction, i.e., the z-axis, as shown in Fig. 8. The separated cuboid section can be fully voxelized with an appropriate resolution, which should be consistent with the grid resolution in the previous stage and suitable in downstream tasks for damage assessment. As seen in Fig. 8, the cuboid after voxelization includes three different types of voxels, i.e., (1) the empty voxels in the damage space, (2) the occupied voxels by the entity surface and damage face, and (3) the unobserved voxels due to occlusion. The benefit of doing this is that the enclosed damage space and the entity can be distinguished using the status of each voxel, i.e., the empty and the occupied voxels. Here, the voxel status is defined as occupied even if only a single point is included, so the voxelization will not lead to losing the point information. In contrast, it will enlarge the single-point impact, which is beneficial for assessment reliability.

In principle, voxelization can be implemented through the Octree [35], but in practice, the observed occupied voxels on the surface can be easily obtained by using the floor-like integerization for point coordinates according to the resolution, as shown in Fig. 9. Here, the integer coordinate (x, y, z) of each node at the corner corresponds to the position of each occupied voxel in the cuboid.

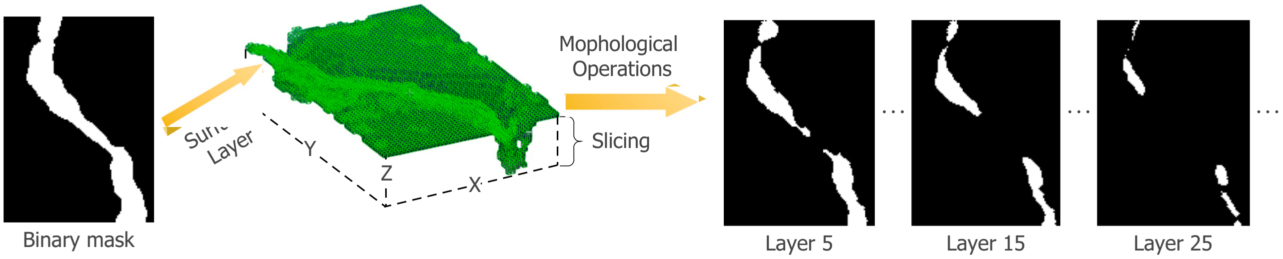


Fig. 10. Surface binary mask and following morphological operations.

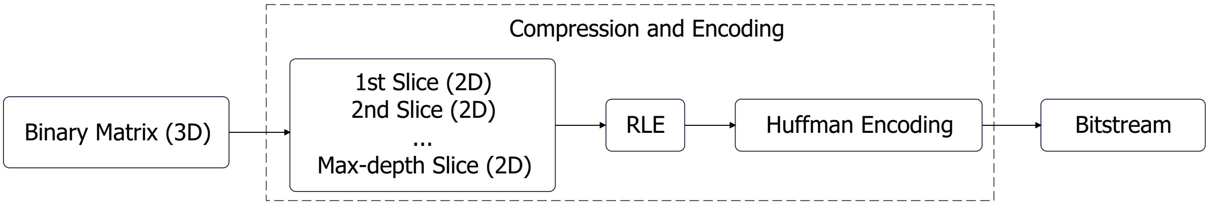


Fig. 11. 3D Binary matrix compression and transmission.

3.5.2. *Binarization and morphological operation*

The binary mask for the detected damage on the surface is shown in Fig. 10, including the damage area (white pixels – 1) and the background area (black pixels – 0). It can be utilized as the damage mask directly on the surface layer of the separated cuboid damage section when grid meshing and voxelization have the same resolution. Otherwise, the mask needs to be resized.

After assigning occupied voxels with 0 and empty voxels (including unobserved voxels) with 1, each layer (or slice) of the voxelized cuboid along the depth direction can be taken as a binary image. Then, the empty voxels, occupied voxels, and unobserved voxels in the following layers along the depth direction can be updated through morphological operations, as shown in Algorithm 1. The Hadamard product enables the damage area to shrink when new occupied voxels arise in the current layer, and image closing can remove the outlier pixels enclosed in the damage area. Finally, the result is a highly lightweight 3D binary matrix representing the damage spatial geometry.

Notably, this processing method relies on the detected damage face points and will not lead to the loss of the damage space. For example, if the equipment cannot perceive the damage face in a few layers, these layers will inherit the damage mask of the previous layer. Hence, the damage area in each layer will only change as the new occupied voxels arise, i.e., new damage face points are detected. This is beneficial for assessment reliability based on the perceived point cloud.

3.6. *Digital-twin synchronization*

3.6.1. *Data compression and transmission*

The transmission data includes the generated 3D binary matrix representing the damage spatial geometry and the corner point coordinates of the segmented cuboid on the surface for geo-referencing. Here, the 3D binary matrix is much lighter than the original point

Algorithm 1 Binarization and morphological operation

```
Input: Voxelized cuboid damage section and structural element $s$
Output: 3D binary matrix to represent damage spatial geometry
Binary mask $M_0 \leftarrow$ $surface\ damage\ segmentation\ (damage\ 1,\ background\ 0)$
while $i \le d_{max}$ $(maximal\ depth)$ do
    $M_i = M_{i-1} \odot M_i (Hadamard\ product)$
    $M_i = M_i \oplus s\ (Dilation)$
    $M_i = M_i \ominus s\ (Erosion)$
end
return $M \leftarrow$ concatenate $(M_1, M_2, \ldots M_{d_{max}})$
```

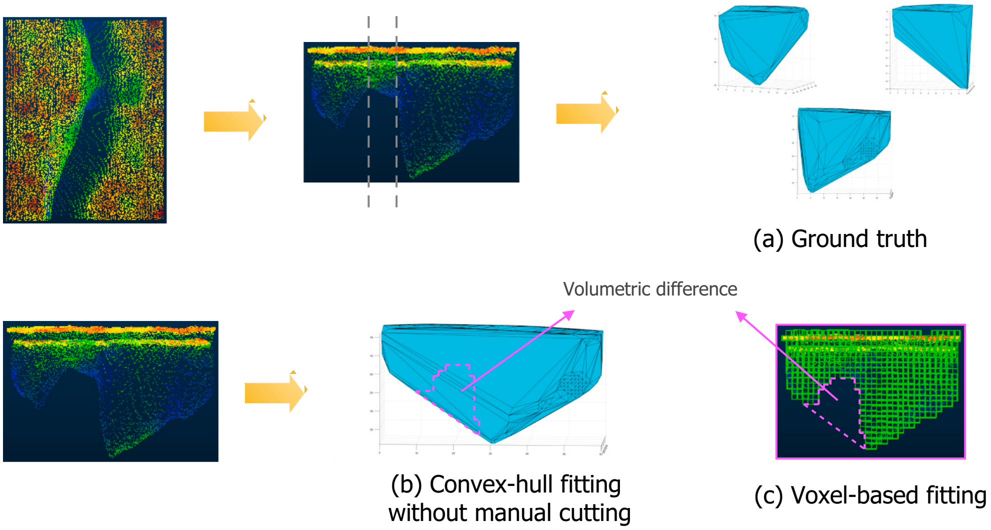


Fig. 12. Comparison of convex-hull fitting and voxel-based fitting: (a) ground truth; (b) convex-hull fitting without manual separation; (c) voxel-based fitting.

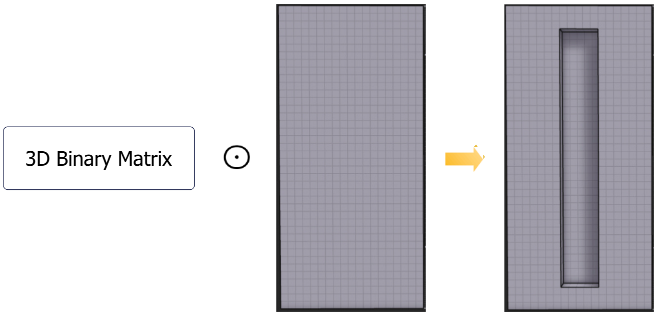


Fig. 13. Proposed FE model geometric updating via Hadamard product.

cloud and can be further compressed through the lossless RLE before transmission, as shown in Fig. 11. The compression efficiency can be evaluated with compression ratio (CR), indicated in Eq. (10).

$$Compression\ Ratio = \frac{Uncompressed\ Size}{Compressed\ Size} \tag{10}$$

3.6.2. *Damage volumetric assessment*

The damage volume assessment aims to evaluate the residual bearing capacity of the damaged structure by quantifying the damaged void space. The latest research [4] uses convex-hull fitting to achieve damage volumetric quantification, but this method requires handcrafted cutting for each convex component. Otherwise, it will exaggerate the damage volume. However, elaborate manual cutting is time-consuming and is not always available for real-world structural damage with complicated non-convex geometry. Hence, voxel-based fitting is proposed to solve this issue in this work.

A point cloud for cracking is shown in Fig. 12. Here, it assumes the crack is wide enough, and the survey provides sufficient points for damage spatial assessment. As can be seen, the damage space is a non-convex geometry, which can be manually separated into three convex polyhedrons for convex-hull fitting, as shown in Fig. 12(a). According to previous research [4], the sum of these three volumes can be taken as the ground truth. Fig. 12(b) shows the convex-fitting result for the intact point cloud without manual separation, and Fig. 12(c) demonstrates the voxel-based fitting result. As can be seen, the convex-hull fitting without manual separation tends to exaggerate the damage volume with the purple-marked volume, and the voxel-based fitting is much closer to the ground truth. In practice, the voxel-based fitting volume can be easily obtained by summing up the value of the transmitted 3D binary matrix (i.e., empty voxel – 1 and occupied voxel – 0) to figure out the voxel amount and then multiplying it with the volume of the unit voxel.

3.6.3. *Model synchronization*

3.6.3.1. FE model geometric updating. Here, the element for FEA has the equivalent resolution of grid meshing and voxelization in Sections 3.4 and 3.5. The latest research [4] demonstrates that the FE model can be updated geometrically by removing the elements less than the damage depth, but the algorithm is based on loop iteration. In principle, with the generated 3D binary matrix, the more efficient method to update the FE geometric model is to apply the Hadamard product on the corresponding cuboid (after geo-referencing) in the FE model, as shown in Fig. 13. Its time complexity is $O(n)$, where $n = l \times w \times d$ (l, w, and d are numbers of elements in the matrix along length, width, and depth, respectively), and space complexity is $O(1)$.

However, most commercial FE software cannot support the above manipulation, and the elements are usually denoted with indices rather than a matrix, such as in Abaqus and Ansys. Hence, the empty voxel coordinates are retrieved by querying with the matrix element equal to 1 and then used to generate the damaged elements' indices. Finally, the FE model can be updated geometrically by deleting the corresponding elements or setting them invalid for the calculation. The pseudocode of this procedure is shown below as Algorithm 2. Its time complexity is $O(n) + O(m)$ as querying across the matrix – $O(n)$ and adapting identified void element IDs – $O(m)$, where n is the number of elements and m is the number of damaged elements in the matrix, which is superior to the previous method [4] with time complexity of $O(3 \times n \times log_2 n)$. Additionally, its space complexity is $O(m)$.

```
Algorithm 2 FE model geometric updating
Input: 3D binary matrix representing damage spatial geometry
Output: Updated FE geometry model
Empty voxel coordinate set $\{X, Y, Z\}$ by querying $M = 1$ (empty =1, occupied = 0)
while $\{x_i, y_i, z_i\}$ in $\{X, Y, Z\}$ do
    Void element $id \leftarrow \{\mathrm{x}_i, y_i, z_i\}$ and $\{\mathrm{x}_0, y_0, z_0\}$
    Append $id$ into void element list $L$
end
Delete the elements or set them invalid according to $L$
```

3.6.3.2. BIM model geometric and semantic updating. The damage contour in each layer (or slice) of the transmitted 3D binary matrix can be fitted with an appropriate shape. For example, the circles or ellipses fitting for spalling can indicate the damage centroid and radius (including major and minor radius) in each layer, which is useful for damage recording and assessment. The 3D damage geometry can be constructed automatically from the transmitted matrix through commercial software like Dynamo. Then, the BIM model can be further updated with the damage geometry following the pipeline shown in Fig. 14 according to previous research [23,36,37]. The fitted damage polyhedron can be taken as an entity and assigned with *IfcVoidingFeature*, which is a modification of an element to reduce its volume. After setting its parameter *PredefinedType* to *CUTOUT*, the damage geometry can be subtracted from the intact component via the *DamagedGeometryCutout* relationship.

4. Proof of concept

4.1. *Experiment preparation*

A public TLS point cloud dataset [15,38] for synthetic grooves and real-world cracking is utilized for framework validation. The synthetic grooves are created on a specimen with different widths (from 1 mm to

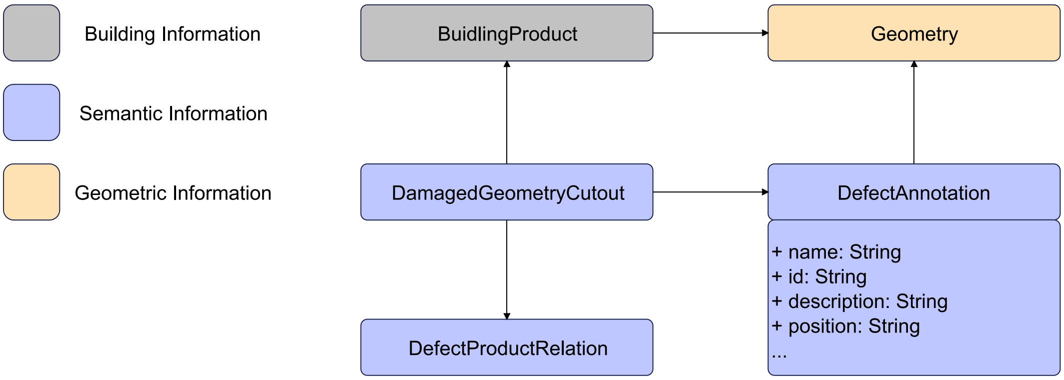


Fig. 14. BIM model updating with the local damage geometry [37].

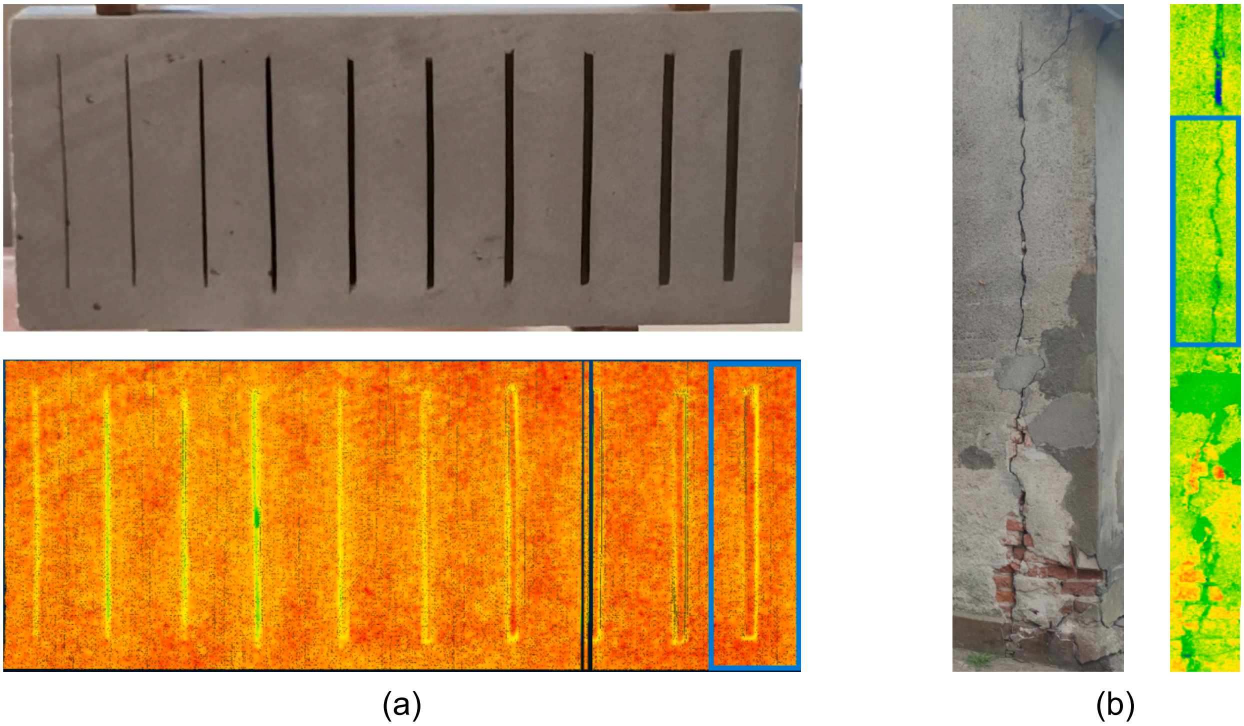


Fig. 15. (a) Synthetic grooves; (b) Real-world building crack.

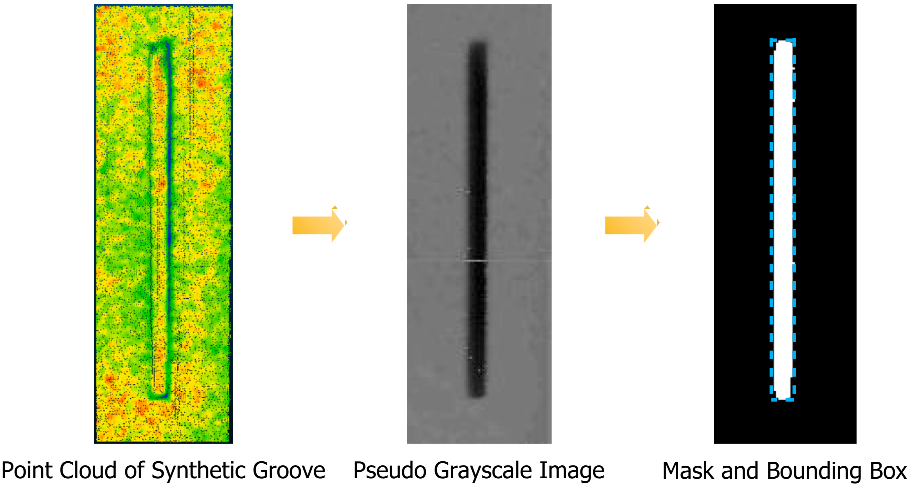


Fig. 16. Groove detection on a pseudo grayscale image.

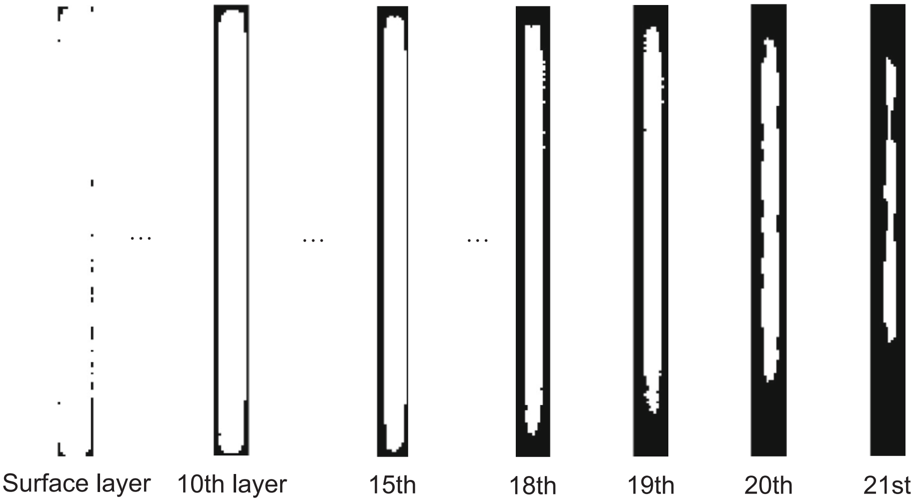


Fig. 17. Each layer (or slice) for the groove in the voxelized section.

10 mm), and the cracking happens on a building wall, as shown in Fig. 15. The survey is taken using a high-quality laser scanner (Z + F IMAGER® 5016) at 5 m with an incidence angle of 0°. The scanner's range resolution is 0.1 mm, and the linearity error is <1 mm + 10 ppm/m. The blue rectangles indicate the selected point clouds for the experiment. CloudCompare and MATLAB are utilized for point-cloud processing. Dynamo is used to reconstruct the 3D damage geometry automatically, and Abaqus is used for updating the FE geometric model.

4.2. Case 1 – Synthetic groove

4.2.1. Groove detection and spatial segmentation

As the wider groove has a better scanning performance for its internal space, the biggest groove with a 10 mm width is adopted for the experiment, shown as blue-marked in Fig. 15(a). As indicated in previous research [15]. its spatial geometry can be described effectively using the point cloud for damage detection. The target groove is initially cropped from the point cloud, and the surface plane is fitted using the MSAC. The pitch angle β of the fitted surface is $-0.0218\ rad$, so the point cloud can be calibrated through Eq. (1), i.e., rotation around the y-axis by β, to make sure the fitted surface plane is horizontal.

The pseudo grayscale image for the cropped section is generated based on the point depth information through the pipeline in Fig. 5, with an average grayscale of 124. The grid resolution is 1 mm. Because the trained DeepLabV3+ model cannot recognize synthetic damage (like grooves), the OTSU thresholding method is adopted here for groove segmentation on the surface. The groove region, i.e., the mask and the bounding box, can be generated, as shown in Fig. 16.

Then, a 3D cuboid involving the groove is separated according to the bounding box from the surface plane until the maximum depth. The cuboid can be voxelized spatially with the same resolution as grid meshing in the above stage, i.e., 1 mm. The occupied voxels by the entity surface and the damage face are set to 0; the empty voxels (including unobserved voxels) are set to 1. Subsequently, the spatial segmentation for the groove is achieved through Algorithm 1 in Section 3.5.1. The voxelized cuboid's layers (or slices) can be shown in Fig. 17, where empty voxels are white and occupied voxels are black. Finally, a 3D binary matrix $M\,(200 \times 35 \times 21)$ representing the groove spatial geometry is generated as a ".mat" file.

4.2.2. DT synchronization

4.2.2.1. Data compression and transmission. The original point cloud of the target section in Fig. 15(a) has 9.5428×10^4 points and can be saved as a ".pcd" file of 2.18 MB. In contrast, the generated 3D binary matrix is saved as a ".mat" file of only 2252 bytes. Compared to transmitting the full-scale point cloud, the proposed method can decrease communication complexity by over 99%. The matrix can be further compressed through RLE to 977 bytes. The compression rate (CR) reaches 56.62%. It is also smaller than the produced matrix (i.e., ".mat" file, 1638 bytes) through the method in previous research [4], which represents the maximum depth in each grid on the target surface. Meanwhile, the x-y coordinates (m) of the bounding box diagonal corner points, i.e., (0.029, 0.197) and (0.042, 0.018), are utilized for geo-referencing of the target groove section on the surface plane. It demonstrates that the proposed framework can enable highly efficient data transmission for DT

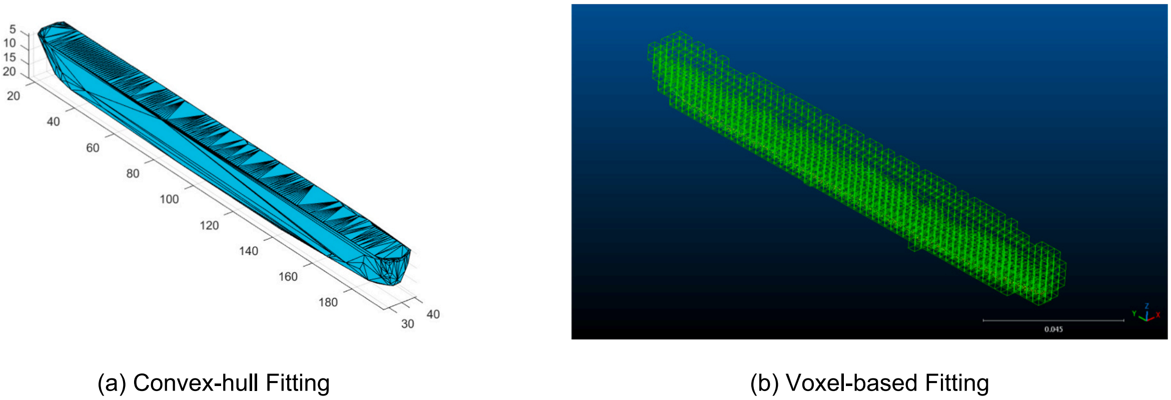
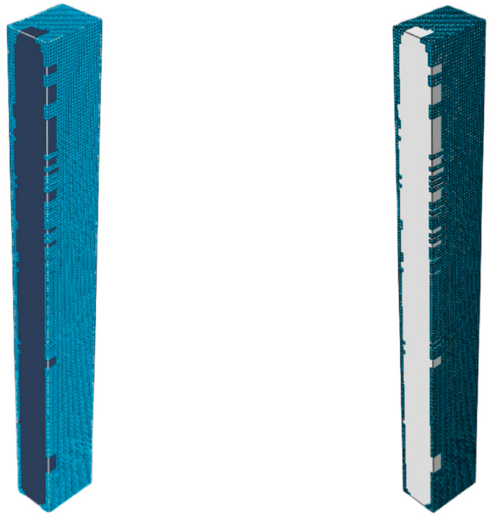


Fig. 18. Groove volumetric assessment with convex-hull and voxel-based fitting.



(a) Groove in hidden mode

(b) Groove in shaded mode

Fig. 19. FE model geometric updating for synthetic groove.

synchronization of the target groove. The generated data through voxelization and binarization is significantly lighter than the original point cloud, which can support the timely as-is model updating during the survey.

4.2.2.2. Groove volumetric assessment. As the target vertical groove is a simply convex geometry, the convex-hull fitting method in the previous research [4] can be implemented directly on the groove point cloud without handcrafted separation, and the volume result is 25,687 mm^3. The voxel-based fitting assessment can be achieved by summing up the Boolean values of the transmitted 3D binary matrix (i.e., the unit voxel is 1 mm^3), and the result is 25,231 mm^3. As can be seen, the result difference between the two fitting methods is <2% for the target groove. It demonstrates the effectiveness of the proposed voxel-based method for volumetric assessment of the 3D damage having a convex geometry based on appropriate resolution (Fig. 18).

4.2.2.3. FE model geometric updating. A model for FEA is established in Abaqus with the equivalent resolution of voxelization. Initially, the target groove section (corresponding to the transmitted 3D binary matrix) can be localized in the FE model using the diagonal corner point coordinates. Then, the ID list of the empty elements can be obtained by retrieving the matrix element that equals 1. Finally, the FE model can be updated automatically by deleting the empty elements in the groove using the Python-based script, as shown in Fig. 19. It demonstrates that the proposed framework can automatically enable FE model geometric updating with the 3D local damage.

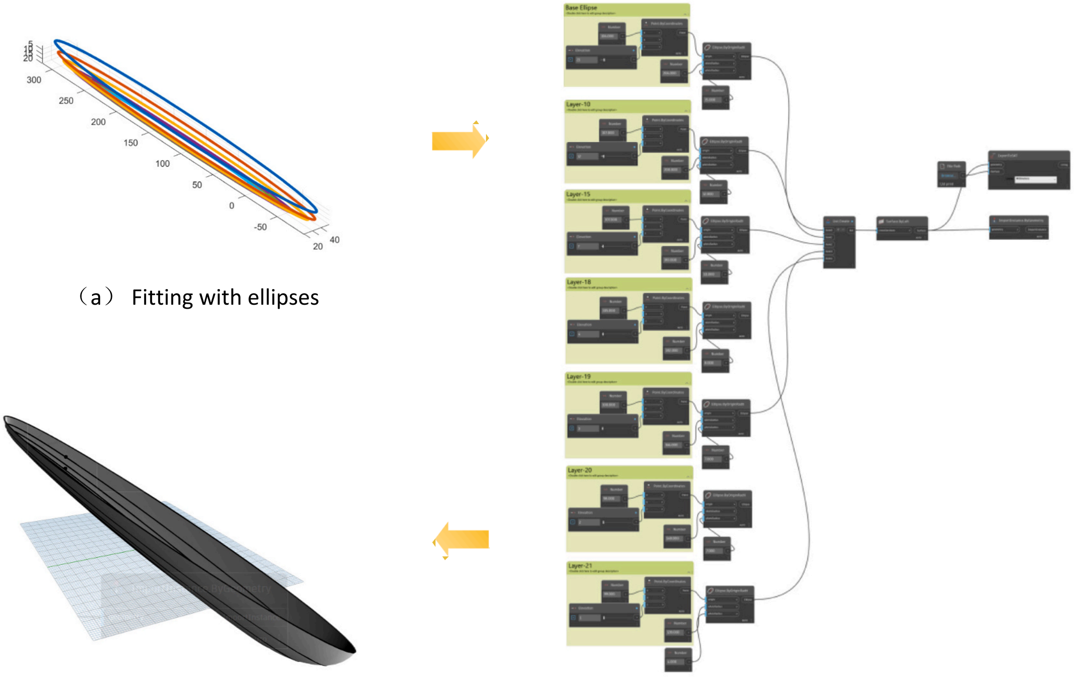
4.2.2.4. BIM model updating. The groove contour in each layer can be fitted with ellipses based on centroids, major and minor axes, as shown in Fig. 20(a), indicating the critical damage features, such as location, width, and length. The fitted ellipses can be generated in Dynamo using *Ellipse.ByOriginRadii* through visual programming, as shown in Fig. 20 (b). Finally, the groove geometry can be reconstructed, as shown in Fig. 20(c).

Furthermore, the groove component can be taken as an entity with *IfcVoidingFeature,* and the parameter *PredefinedType is* set to *CUTOUT*. Then, the groove geometry can be subtracted from the intact *BuildingProduct* based on the *DamagedGeometryCutout* relationship. Consequently, *BuildingProduct* will point to the specimen with *componentGeometry*, and *DamagedGeometryCutout* will refer to the groove with *damagedGeometry*. Finally, *DefectAnnotation* (including name, id, description, position, etc.) can be associated with the groove component to update the semantic information in the BIM model. The complete pipeline is shown in Fig. 14.

4.3. Case 2 – Real-world building crack

4.3.1. Crack detection and spatial segmentation

In the second case, the framework is tested on the point cloud of a real-world building crack, as shown in Fig. 15 (2). The experiment



（a） Fitting with ellipses

（c） Reconstructed groove geometry

（b） Visual programming

Fig. 20. BIM model geometric updating for synthetic groove.

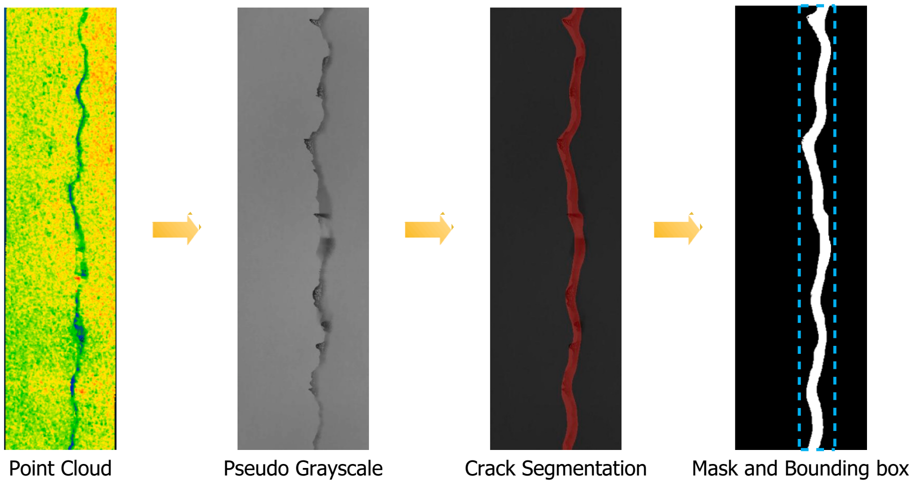


Fig. 21. Real-world crack detection with pseudo grayscale image via DeepLabV3+.

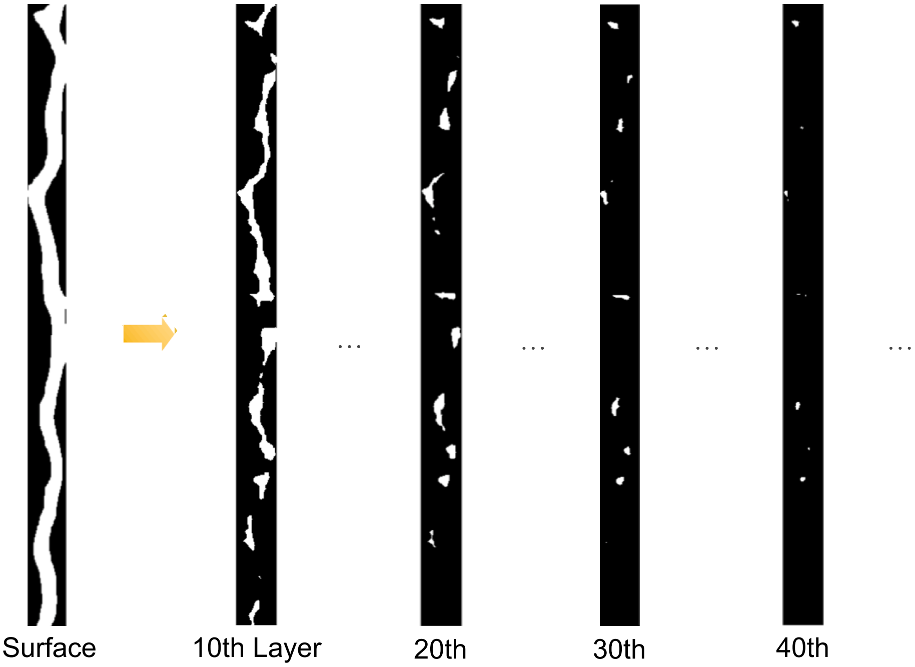


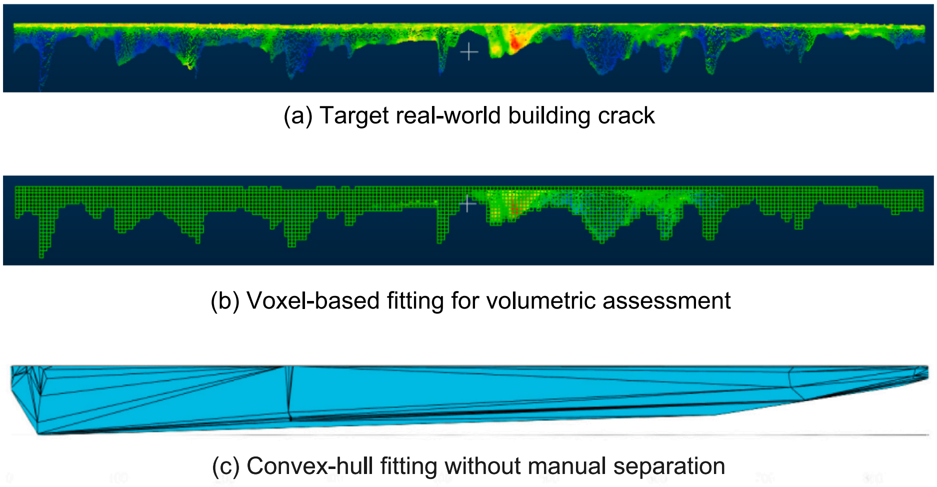
Fig. 22. Each layer (or slice) for real-world cracking in the voxelized section.

adopts the blue-marked region, with the cracking width from 11 mm to 30 mm. As the high-resolution TLS scanning can provide sufficient spatial information for the synthetic groove with a width of 10 mm (see Section 4.2.1), the point-cloud data under the same survey condition (i.e., the same equipment, angle, and distance) for the crack >11 mm is available for damage spatial assessment. The surface plane is fitted with the MSAC, and the point-cloud calibration is achieved through Eq. (1) with $\beta = 0.0023\,rad$. The pseudo grayscale image is entirely derived from the point depth with a grid resolution of 1 mm and utilized for the crack segmentation on the surface through the pre-trained DeepLabV3+ model. The crack mask and bounding box are shown in Fig. 21.

The 3D crack section (i.e., cuboid) is separated according to the bounding box from the surface plane until the maximum depth. The cuboid is voxelized spatially at a resolution of 1 mm. The occupied voxels by the entity surface and the damage face are set to 0; the empty voxels (including unobserved voxels) are set to 1. The spatial segmentation for the crack can be achieved through Algorithm 1 in Section 3.5.1. The voxelized cuboid's layers (or slices) can be shown in Fig. 22, where empty voxels are white and occupied voxels are black. Finally, a 3D binary matrix M (850 × 55 × 56) representing the crack spatial geometry can be generated as a ".mat" file.

4.3.2. *DT synchronization*

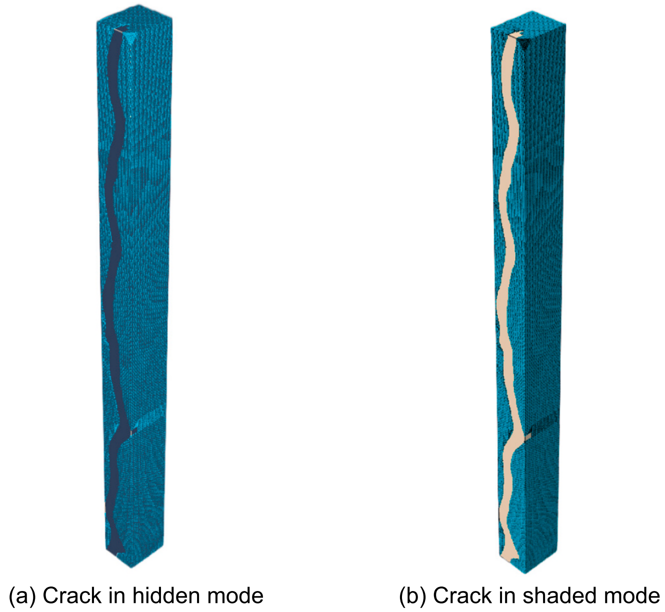
4.3.2.1. Data compression and transmission. The original point cloud of the target crack section in Fig. 15(b) has 2.001331×10^6 points and is saved as a ".pcd" file of 53.4 MB. In contrast, the generated 3D binary matrix is a ".mat" file of only 58.6 KB. Compared with the full-scale point cloud, the transmission payload size decreases by over 99%. The matrix can be further compressed through lossless RLE until 16.9 KB. The compression rate (CR) reaches 71.16%. It is also smaller than the result (i.e., ".mat" file, 40.2 KB) through the method in previous research [4]. Meanwhile, the x-y coordinates (m) of the bounding-box diagonal corner points, i.e., (0.129, 0) and (0.184, 0.183), are utilized for geo-referencing of the target crack section on the surface. It demonstrates that the proposed framework can enable highly efficient data


(a) Target real-world building crack

(b) Voxel-based fitting for volumetric assessment

(c) Convex-hull fitting without manual separation

Fig. 23. Volumetric assessment for the real-world building crack.


(a) Crack in hidden mode

(b) Crack in shaded mode

Fig. 24. FE model geometrically updating for real-world cracking.

transmission for DT synchronization of the real-world building crack. The transmitted data is significantly lighter than the original point cloud, which can support the as-is model updating during the survey.

4.3.2.2. Damage volumetric assessment. The target real-world building crack is a complicated non-convex geometry, shown in Fig. 23(a), so it is impractical to apply elaborate handcrafted separation for each convex component to achieve volumetric assessment through the convex-hull fitting. In contrast, the voxel-based fitting can be easily applied for the damage volumetric assessment in this situation. As Section 4.2.2.2 has demonstrated the effectiveness of the voxel-based method for a convex geometry, the voxel-based fitting with an appropriate resolution can be closer to the ground truth than the convex-hull fitting for the intact point cloud without manual cutting, as shown in Fig. 23(b) and (c). In practice, the voxel-based volumetric assessment can be achieved by summing up the Boolean values of the transmitted 3D binary matrix (i.e., the unit voxel is 1mm^3), and its result is 4.0652×10^5 mm^3. This method can also be extended for volumetric assessment of other 3D damages having a non-convex geometry, such as the spalling in Fig. 1. It demonstrates that the proposed voxel-based method can be easily applied instead of the convex-hull fitting for volumetric assessment of the real-world damage having a non-convex geometry.

4.3.2.3. Model updating. Like Section 4.2.2.3, an FE model can be established in Abaqus with the equivalent resolution of voxelization. The separated crack section (i.e., a cuboid) can be localized using the diagonal corner point coordinates in the FE model. Then, the ID list of the empty element can be obtained by retrieving the matrix element that equals 1. Finally, the FE model can be updated automatically by deleting the empty elements in the cracking space through a Python-based script, as shown in Fig. 24. It demonstrates that the proposed framework can update the FE geometric model efficiently and automatically for real-world building crack based on the point cloud.

Furthermore, the crack contour in each layer can be fitted using a bounding box, as shown in Fig. 25, to indicate the cracking location and skeleton. Then, the crack spatial geometry can be generated by meshing the bounding boxes across different layers. Finally, the BIM model can be updated geometrically and semantically like in Section 4.2.2.4. The proposed framework for FE and BIM model updating with local damage is also available for other 3D structural damage, such as concrete spalling shown in Fig. 1.

5. Discussion

Although the proposed framework performs excellently on synthetic and real-world spatial damage in the case studies, it still has some limitations. For example, the accessibility of the target from the LiDAR plays an important role in the proposed methodology while capturing the point cloud with the required interpoint spacing, which is crucial to the performance of damage assessment and reality modelling. This study keeps the scanner 5 m apart from the target and perpendicular to the planner surface without any angles in the experiment. However, this configuration may not always be practical for a survey of damaged structures if they cannot be accessible like this. Hence, it is necessary to discuss the ambient and intrinsic factors affecting the proposed methodology's performance, summarising as distance, angle, and edge effect (or laser beam size).

The distance from the scanner to the target influences measurement accuracy and point cloud density. As the distance grows, the measurement accuracy declines, so it is necessary to adapt the grid and voxel resolution accordingly. Although the accuracy remains acceptable within a specific distance range determined by the equipment, increased distance leads to a decrease in point density. This results in more empty

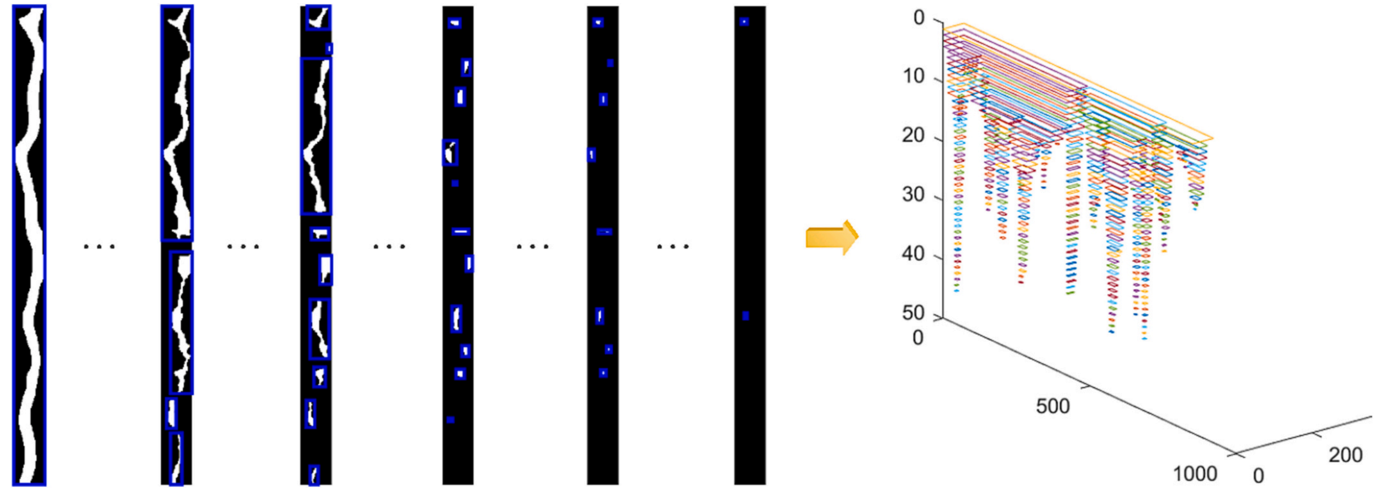


Fig. 25. Crack contour in each layer fitted with bounding boxes.

Table 2
Comparative analysis between proposed methodology and existing practices.

Methodology (data form)	Payload	Volumetric Assessment	FE complexity (time/space)	BIM updating
Project-based (full-scale PCD)	Heavy	Manually comparing	$O(P \times N)$ $O(M)$	Time-consuming
Research [4] (segmented PCD)	Medium	Manual cutting (convex)	$O(3 \times n \times log_2 n)$ $O(m)$	n/a
Proposed framework (binary matrix)	Light	Automatically	$O(n) + O(m)$ $O(m)$	Fast

Note: PCD – point cloud data; the damage volumetric estimation in the current project-based survey is achieved by manually comparing the full-scale point cloud with an ideal 3D model; P is the number of all the points in PCD; N is the number of elements, and M is the number of damaged elements in the full-scale model; n is the number of elements and m is the number of damaged elements in the target section.

grids or voxels without enclosed points, consequently raising ambiguity in determining whether a voxel is void within a damaged volume. In such instances, appropriate computer vision methods based on image processing or deep learning are required to perform effective spatial denoising.

The angle of incidence is directly related to the scanning field of the structural damage. The previous study [15] has shown that when the incidence angle is oblique, it causes a restricted visual field of the damage area due to occlusion, resulting in loss of information during inspection. Therefore, when it is impossible to conduct perpendicular scanning, it becomes necessary to perform scanning from different viewpoints to acquire sufficient damage spatial information. Then, the damage space can be reconstructed or potentially predicted through various technologies like neural radiance fields (NeRF) or 3D shape completion. This procedure can be carried out before transmission with point clouds or after transmission using the generated binary matrices through the proposed method.

Although the damage size and depth may meet specific criteria for a high-quality point cloud description, the edge effect still exists. As defined in [15], this effect refers to using an average object distance when the laser beam covers multiple surfaces with varying depths along the edge. It will blur the damage boundary in the point cloud and can only be mitigated by reducing the size of the laser beam. Therefore, a modest enhancement of the points' depth along the edge would benefit damage segmentation in the pseudo grayscale images (i.e., equivalent to image sharpening) while also expanding the margin for reliable damage assessment.

6. Conclusion

This study presents a highly efficient framework for damage volumetric assessment and DT synchronization based on the point cloud. It includes surface damage detection, damage spatial segmentation, and DT model synchronization. The surface damage detection is achieved through a pre-trained DeepLabV3+ model on the pseudo grayscale images derived from the point cloud depth information. It avoids the drawbacks of using image and point cloud fusion, such as extra photo acquisition, complex coordinate transformation and potential errors. The approach is validated on the specimens created in the lab with different crack widths and depths. It demonstrates that the trained model from the real grayscale images for damage detection can be applied to the pseudo grayscale images when the damage satisfies certain conditions, such as appropriate width and normalization threshold. Meanwhile, the natural damage and manual concave patterns can be distinguished via the DeepLabV3+ model by training on the annotated damage and non-damage images.

After surface damage detection, the bounding box and mask are utilized for damage spatial segmentation to remove the redundant point cloud. Then, the separated point cloud can be converted into a highly lightweight 3D binary matrix representing the spatial damage geometry through voxelization and binarization. Compared with the full-scale object's point cloud transmission, communication complexity can be significantly decreased (over 99%) by only transmitting the segmented target point cloud in a binary matrix. The binary matrix can be further compressed through lossless RLE (with a CR of over 50%) for efficient data transmission and practical downstream tasks. This approach addresses the communication complexity challenge when DT synchronizes with voluminous point cloud data in 3D scanning surveys.

Finally, the transmitted 3D binary matrix and the geo-referencing coordinates can be utilized for different downstream tasks, including SHM and reality modelling. For example, the damage volumetric assessment can be easily achieved by summing up the elements of the binary matrix (as void voxel – 1 and occupied voxel – 0). This voxel-based method doesn't require elaborate manual cutting for each component like the previous method based on the convex-hull fitting [4]. It can perform well for real-world damage with a complicated non-convex geometry under an appropriate resolution. Moreover, the FE model can be updated geometrically through the binary matrix with lower time and space complexity (see Table 2). The BIM model can also be updated automatically as an IFC file using appropriate patterns (such as ellipses and bounding boxes) to fit the damage contour in each layer.

The intact framework is validated based on two case studies, i.e., a synthetic groove and a real-world building crack. A comparative analysis, as shown in Table 2, demonstrates that the proposed framework is superior to the existing practices for damage assessment and reality modelling via 3D scanning.

It is important to highlight that the proposed framework and approaches hold promise for further study in extending their application to spatial damage on the object with a curved surface, such as cracks or spalling on a pillar. The proposed methodology can revolutionize the existing workflow for infrastructure maintenance surveys based on 3D scanning, which is unidirectional, time-consuming, and burdensome due to the large volume of point cloud data. Instead, it can facilitate near real-time damage assessment and reality modelling during the scanning process and provide timely feedback to the physical entity, effectively streaming the challenging 3D scanning survey into the infrastructure's DT.

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.

Acknowledgements

This work was supported by the Cardiff School of Engineering and the Cardiff University – China Scholarship Council (CSC) joint program.

References

[1] C. Axel, J. van Aardt, Building damage assessment using airborne lidar, J. Appl. Remote. Sens. 11 (04) (2017) 1, https://doi.org/10.1117/1.jrs.11.046024.

[2] M. Rashidi, M. Mohammadi, S.S. Kivi, M.M. Abdolvand, L. Truong-Hong, B. Samali, A decade of modern bridge monitoring using terrestrial laser scanning: review and future directions, Remote Sens. 12 (22) (2020) 1–34, https://doi.org/10.3390/rs12223796.

[3] F. Zhang, et al., Instance segmentation of LiDAR point clouds, Proc. IEEE Int. Conf. Robot. Autom. (2020) 9448–9455, https://doi.org/10.1109/ICRA40945.2020.9196622.

[4] Q. Kong, J. Gu, B. Xiong, C. Yuan, Vision-aided three-dimensional damage quantification and finite element model geometric updating for reinforced concrete structures, Comput. Aided Civ. Inf. Eng. (2023), https://doi.org/10.1111/mice.12967.

[5] M. Nie, C. Wang, Pavement Crack Detection based on yolo v3, in: Proceedings - 2019 2nd International Conference on Safety Produce Informatization 2019, IICSPI, 2019, pp. 327–330, https://doi.org/10.1109/IICSPI48186.2019.9095956.

[6] C. Zhang, C.C. Chang, M. Jamshidi, Bridge Damage Detection using a Single-Stage Detector and Field Inspection Images [Online]. Available: http://arxiv.org/abs/1812.10590, 2018.

[7] Y. Xue, Y. Li, A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects, Comput. Aided Civ. Inf. Eng. 33 (8) (2018) 638–654, https://doi.org/10.1111/mice.12367.

[8] H. Fu, D. Meng, W. Li, Y. Wang, Bridge crack semantic segmentation based on improved deeplabv3+, J. Mar. Sci. Eng. 9 (6) (2021) https://doi.org/10.3390/jmse9060671.

[9] S. Li, X. Zhao, Image-based concrete crack detection using convolutional neural network and exhaustive search technique, Adv. Civil Eng. 2019 (2019), https://doi.org/10.1155/2019/6520620 no. Ml.

[10] N.H.T. Nguyen, S. Perry, D. Bone, H.T. Le, T.T. Nguyen, Two-stage convolutional neural network for road crack detection and segmentation, Expert Syst. Appl. 186 (December 2020) (2021), https://doi.org/10.1016/j.eswa.2021.115718, pp. 115718.

[11] E. Zhang, L. Shao, Y. Wang, Unifying transformer and convolution for dam crack detection, Autom. Constr. 147 (December) (2022), 104712, https://doi.org/10.1016/j.autcon.2022.104712.

[12] J.S. Kang, K. Chung, E.J. Hong, Multimedia knowledge-based bridge health monitoring using digital twin, Multimed. Tools Appl. 80 (26–27) (2021) 34609–34624, https://doi.org/10.1007/s11042-021-10649-x.

[13] Y. Gao, H. Li, G. Xiong, H. Song, AIoT-informed digital twin communication for bridge maintenance, Autom. Constr. 150 (March) (2023), https://doi.org/10.1016/j.autcon.2023.104835, pp. 104835.

[14] S. Cho, S. Park, G. Cha, T. Oh, Development of image processing for crack detection on concrete structures through terrestrial laser scanning associated with the octree structure, Appl. Sci. 8 (12) (2018) 2373, https://doi.org/10.3390/app8122373.

[15] P. Stałowska, C. Suchocki, M. Rutkowska, Crack detection in building walls based on geometric and radiometric point cloud information, Autom. Constr. 134 (November) (2021), https://doi.org/10.1016/j.autcon.2021.104065, pp. 104065.

[16] Z. Zhang, A flexible new technique for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1330–1334, https://doi.org/10.1109/34.888718.

[17] H. Zhang, Y. Zou, E. del Rey Castillo, X. Yang, Detection of RC spalling damage and quantification of its key properties from 3D point cloud, KSCE J. Civ. Eng. 26 (5) (2022) 2023–2035, https://doi.org/10.1007/s12205-022-0890-y.

[18] C. Liu, L. Zhou, W. Wang, X. Zhao, Concrete surface damage volume measurement based on three-dimensional reconstruction by smartphones, IEEE Sensors J. 21 (10) (2021) 11349–11360, https://doi.org/10.1109/JSEN.2021.3067739.

[19] C. Ye, S.C. Kuok, L.J. Butler, C.R. Middleton, Implementing bridge model updating for operation and maintenance purposes: examination based on UK practitioners' views, Struct. Infrastruct. Eng. 18 (12) (2022) 1638–1657, https://doi.org/10.1080/15732479.2021.1914115.

[20] C. Tu, E. Takeuchi, A. Carballo, K. Takeda, Point cloud compression for 3d lidar sensor using recurrent neural network with residual blocks, Proc. Int. Conf. Robot. Auto. 2019-May (2019) 3274–3280, https://doi.org/10.1109/ICRA.2019.8794264.

[21] T. Huang, Y. Liu, 3D point cloud geometry compression on deep learning, in: *MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 890–898, https://doi.org/10.1145/3343031.3351061.

[22] M. Artus, M.S.H. Alabassy, C. Koch, A BIM based framework for damage segmentation, modeling, and visualization using IFC, Appl. Sci. (Switzerland) 12 (6) (2022) 1–26, https://doi.org/10.3390/app12062772.

[23] M. Artus, C. Koch, Object-oriented damage information modeling concepts and implementation for bridge inspection, J. Comput. Civ. Eng. 36 (6) (2022) 1–21, https://doi.org/10.1061/(asce)cp.1943-5487.0001030.

[24] M. Artus, C. Koch, Modeling geometry and semantics of physical damages using IFC, in: EG-ICE 2020 Workshop on Intelligent Computing in Engineering, Proceedings, 2020, pp. 144–153 no. July, https://depositonce.tu-berlin.de/items/848a354a-ed23-4174-85f9-e937fb3133b7.

[25] D. Isailović, V. Stojanovic, M. Trapp, R. Richter, R. Hajdin, J. Döllner, Bridge damage: detection, IFC-based semantic enrichment and visualization, Autom. Constr. 112 (January) (2021), 103088, https://doi.org/10.1016/j.autcon.2020.103088.

[26] E. Aldemir, G. Tohumoglu, M.A. Selver, Binary medical image compression using the volumetric run-length approach, Imaging Sci. J. 67 (3) (2019) 123–135, https://doi.org/10.1080/13682199.2019.1565695.

[27] Fit plane to 3-D point cloud - MATLAB pcfitplane - MathWorks United Kingdom. https://uk.mathworks.com/help/vision/ref/pcfitplane.html (accessed Aug. 02, 2023).

[28] Random Sample Consensus – Wikipedia. https://en.wikipedia.org/wiki/Random_sample_consensus (accessed Aug. 02, 2023).

[29] 'Your garden walls: better to be safe', Garden, pp. 4–6, Accessed: May 29, 2023. [Online]. Available: https://www.gov.uk/guidance/your-garden-walls-better-to-be-safe.

[30] D. buildings Ltd, Cracking and building movement, Struct. Surv. 23 (1) (Feb. 2005), https://doi.org/10.1108/ss.2005.11023aae.007.

[31] NR-L3-CIV-006 - Handbook for The Examination of Structures | PDF | Highway | Road. https://www.scribd.com/document/632406023/NR-L3-CIV-006-Handbook-for-the-Examination-of-Structures (accessed Jun. 15, 2023).

[32] E. Bianchi, M. Hebdon, Concrete Crack Conglomerate Dataset, University Libraries, 2021. Accessed: May 29, 2023. [Online]. Available: https://data.lib.vt.edu/articles/dataset/Concrete_Crack_Conglomerate_Dataset/16625056/1.

[33] A. Akagic, E. Buza, S. Omanovic, A. Karabegovic, Pavement crack detection using Otsu thresholding for image segmentation, in: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings, 2018, pp. 1092–1097, https://doi.org/10.23919/MIPRO.2018.8400199.

[34] The PASCAL visual object classes homepage, Pascal-Voc. (2018). Accessed: May 29, 2023. [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/.

[35] Y. Xu, X. Tong, U. Stilla, Voxel-based representation of 3D point clouds: methods, applications, and its potential use in the construction industry, Autom. Constr. 126 (March) (2021), https://doi.org/10.1016/j.autcon.2021.103675, pp. 103675.

[36] M. Artus, M. Alabassy, C. Koch, A BIM based framework for damage segmentation, modeling, and visualization using IFC, Appl. Sci. (Switzerland) 12 (6) (2022) 10–12, https://doi.org/10.3390/app12062772.

[37] M. Artus, M. Alabassy, C. Koch, IFC based framework for generating, modeling and visualizing spalling defect geometries, in: Proceedings of EG-ICE 2021 Workshop on Intelligent Computing in Engineering, 2021, pp. 176–186. https://depositonce.tu-berlin.de/items/2b47a17c-284b-4795-a2c9-8bda8dc50a30.

[38] P. Stałowska, C. Suchocki, TLS data for cracks detection in building walls, Data Brief 42 (May) (2022), https://doi.org/10.1016/j.dib.2022.108247, pp. 108247.