Automation in Construction 168 (2024) 105878

Contents lists available at ScienceDirect

Automation in Construction

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

3D Pixelwise damage mapping using a deep attention based modified Nerfacto

Geontae Kim, Youngjin Cha *

Department of Civil Engineering, University of Manitoba, Winnipeg, MB, Canada

ARTICLE INFO

Keywords:
SHM
Damage detection
Deep learning
Crack segmentation
3D reconstruction
Neural radiance fields
NeRF
Nerfacto
Digital twin

ABSTRACT

Recent advancements in structural health monitoring have highlighted the necessity for accurate three-dimensional (3D) damage mapping on digital twins, moving beyond traditional methods such as photogrammetry, which frequently struggle to capture intricate planar surfaces. To address this limitation, this paper proposes a new advanced 3D reconstruction method and its integration with 3D damage mapping techniques. As the 3D reconstruction method, an Attention-based Modified Nerfacto (ABM-Nerfacto) model is developed, and is integrated with an advanced damage segmentation method. Using a three-span continuous bridge with concrete piers as an example structure, and concrete cracks as the example damage, the state-of-the-art STRNet is utilized for crack segmentation. Through extensive parametric studies and comparative evaluations, the proposed ABM-Nerfacto model was demonstrated to produce high-quality 3D reconstructions and corresponding damage mappings for this bridge system. This integrated approach provides a promising solution for comprehensive 3D digital twin-based structural health monitoring.

1. Introduction

For civil infrastructure damage detection, traditional image processing techniques with computer vision input have been extensively investigated. However, these traditional approaches have clear limitations: 1) manual formulation of damage-sensitive feature extractions is required, 2) they can only detect a single type of damage, and 3) they do not work well if the image quality is not ideal [9]. Due to the limitations of traditional image processing-based methods, recent developments in deep learning have led to significant innovations in structural health monitoring [6]. Deep learning methods have been extensively employed to overcome the shortcomings of the traditional approaches, based on pioneering research [7,8]. Following these initial studies, numerous follow-up research has been conducted, ranging from image-level classification [20], object detection [3,8,22], to pixel-wise segmentation [4] of various structural damages.

As the latest approach, pixel-wise segmentation of damage has gained attention because it can quantify the sizes of detected damage. Consequently, many existing encoder-decoder based deep convolutional neural networks (CNNs) have been employed for damage (crack) segmentation, such as U-Net-based methods [23,29], DenseNet-based approaches [2], ResNet-based methods [33], Mask R-CNN-based approaches [21,24], DeepLabV3+ [11] based applications [18], and hybrid approaches [21,35]. However, many conventional approaches employ neural networks originally developed for different applications, not specifically tailored for detecting structural damages such as cracks. These generic models often struggle with the unique characteristics of structural damage, such as their low aspect ratios and small pixel proportions, which are crucial for accurate detection. As a result, the accuracy of these methods may not meet the high standards required for structural health monitoring, necessitating a tremendous number of ground truth data labeled and corresponding a huge number of learnable parameters, which lead to high computational costs and make real-time processing of normal videos (e.g., 30 frames per second (FPS) with a large enough frame size like 1200 × 1000) infeasible [6].

To overcome the fundamental limitations of traditional image processing-based approaches, object-specific advanced deep learning models have been developed for structural damage segmentation. For example, SDDNet [12] was developed by modifying the atrous spatial pyramid pooling (ASPP) of the original DeepLabV3+ and incorporating depthwise separable convolutions, as well as creating a DenSep module to reduce computational cost and improve the accuracy of crack segmentation. Similarly, STRNet [19] was created by adopting squeeze and extension-based attention module and multi-headed self-attention

* Corresponding author at: SP427 Stanley Pauley Engineering, 97 Dafoe Rd W, Winnipeg, MB R3T 6B3, Canada.
E-mail address: young.cha@umanitoba.ca (Y. Cha).

https://doi.org/10.1016/j.autcon.2024.105878
Received 6 July 2024; Received in revised form 8 November 2024; Accepted 12 November 2024
Available online 19 November 2024

module in the encoder and decoder and generating an STR module with three different configurations and a learnable nonlinear swish activation function. Additionally, coarse upsampling and a focal-Tversky loss function were adopted to optimize the network's computational cost and performance. As a result, STRNet demonstrated state-of-the-art performance, achieving a mean intersection over union (mIoU) of 92.6 % for segmenting cracks on complex background scenes, with real-time processing off large input image frames (1200 × 800 × 3) at 49 FPS.

These deep learning-based damage detection methods using computer vision can segment damages within input images. However, to localize the segmented damages, it is necessary to determine the location of the field of view of the specific image. To achieve this, geotagging techniques have been used to approximately localize the 2D scene of the image within the structure of interest [3,20,34]. Nevertheless, localizing the segmented damages in a 2D map is not sufficient, as civil infrastructure is inherently a 3D structural system. For holistic monitoring and management purposes, mapping of 2D damage on reconstructed 3D models is required.

To overcome the spatial limitations of damage mapping in 2D images, various 3D reconstruction methods have been developed. For instance, radar has been employed to generate 3D point cloud models for seismically damaged structural members [14]. Additionally, the integration of YOLOv5sHSC-based damage detection and structure-from-motion (SfM)-based 3D reconstruction has been proposed to localize the detected damage within a 3D reconstructed model, leveraging the positional relationship between the camera and the reconstructed model coordinates for each damage [38]. To mitigate critical imaging errors, such as distortion caused by reflectivity and occluded light paths, an optical laser triangulation system fusing a linear laser ray generator and a sports camera has been developed for 3D reconstruction of concrete defects [17]. Furthermore, the combination of a laser liner and a stereo vision camera has been used to create a single-stripe-enhanced space-time stereo 3D reconstruction of concrete cracks [16]. However, these methods have been primarily applied to small-scale structural members or limited areas of concrete walls.

Neural Radiance Fields (NeRF) [26] have recently surfaced as an innovative method, drawing considerable interest as a promising substitute for traditional photogrammetry, which frequently fails to capture complex visual details accurately. In contrast to conventional photogrammetry that relies heavily on well-defined feature points, NeRF excels in processing objects with sparse features, offering a more adaptable approach for 3D reconstruction. This reliance on the quality of input data distinguishes it further. Moving away from traditional techniques that depend on manual texture mapping and geometric primitives such as triangles and voxels, NeRF utilizes a neural network to directly simulate the visual attributes of a scene. This advanced strategy allows NeRF to assimilate scenes from variably positioned camera captures and adeptly handle intricate lighting dynamics, including reflections and transparency, through distinctive point representations from multiple viewpoints. As a result, NeRF introduces a more dynamic and efficient alternative for 3D reconstruction, representing a notable shift from established methodologies.

Therefore, NeRF-based 3D reconstruction methods have also been applied to construction progress monitoring problems. For example, Pal et al. [28] proposed a novel framework, the Activity-level Progress Monitoring System (ALPMS), which utilizes a 4D Building Information Model (BIM) and the original NeRF algorithm to precisely measure the percent completion of construction activities associated with building elements. The NeRF model, trained for individual elements and with the virtual camera placed at the average distance of the training cameras from the element's face, synthesized the best quality orthographic view image for the ALPMS.

Additionally, the NeRF approach has been implemented in 3D damage mapping through the integration of deep learning-based damage segmentation methods. For instance, Yu et al. [37] proposed a Knn-Swin-T network to segment cracks in underwater bridge piers, and also generated 3D reconstruction using a series of NeRF models (NeRF, FastNeRF [15], MVSNeRF [10], IBRNet [32], and VNRF [36]). The performances of these original NeRF models showed nearly the same results, achieving a mIoU of 0.87 with 22 FPS using an input image size of 1024 × 1024. However, in this study, the original NeRF models were applied to a miniature structure model and a limited area of a real structure for segmented crack mapping.

Despite these promising results, there is still significant potential for improvement mapping of 2D damage on the 3D reconstructed model. The ability to efficiently and accurately reconstruct large-scale structures and capture detailed damage information across the entire structure remains a key challenge that requires further research and development. This is because the current NeRF-based approaches have several limitations: 1) significant computational cost: NeRF models only work effectively for compact-sized objects, including chairs or desks, due to their high computational requirements, 2) Restricted capability in accurately predicting pixel opacity and RGB values: NeRF models struggle to precisely predict actual pixel opacity and RGB values, often due to the limitations of the embedded deep neural networks and the method of selecting 3D sample points in each ray, and 3) unsuitable for large-scale civil infrastructures: NeRF models struggle with the vast scale and intricate details of large civil structures, which demand extensive computational resources and sophisticated data handling capabilities that exceed the capacities of standard NeRF implementations.

To overcome these limitations, the original Nerfacto model was introduced in [30] as an advanced variant of the original NeRF, specifically designed to enhance 3D scene reconstruction for large-scale environments that the original NeRF struggled to manage. Ref. [13] demonstrated the potential of Nerfacto within their SRecon-NeRF framework, which combines semantic and geometric information to generate a semantic point cloud for monitoring indoor construction progress. Evaluation results indicate that SRecon-NeRF outperforms existing semantic-based methods by 24 % in accuracy and 75 % in speed, while achieving a 36 % improvement in accuracy and an 83.3 % increase in speed compared to geometric-based methods.

However, this advanced version of NeRF, including the Nerfacto model and SRecon-NeRF, still encounters challenges in maintaining high fidelity during large-scale 3D reconstructions. This limitation is particularly noticeable in capturing the fine details necessary for accurately assessing structural integrity. Addressing these challenges is critical for applying the original Nerfacto model and its variants to structural health monitoring (SHM), where precision and detail are essential for ensuring safety.

To address these limitations, this paper proposes an enhanced ABM-Nerfacto-based 3D damage mapping approach. The proposed method integrates the state-of-the-art (SOTA) STRNet model for advanced deep learning-based pixel-wise crack segmentation. The architecture of ABM-Nerfacto is significantly modified from original Nerfacto model, including the integration of a multi-head self-attention module, to improve the quality of the 3D reconstruction and the 3D damage mapping.

The paper is organized as follows: Section 2 presents the Methodology, Section 3 shows data collection, Section 4 discusses the Case Studies, Section 5 provides the Conclusions, and Section 6 lists the References.

2. Methodology

This paper presents a 3D damage mapping method that integrates a 3D reconstruction technique with an advanced deep learning-based pixel-wise damage segmentation approach. The 3D reconstruction is achieved using an ABM-Nerfacto model, while the damage segmentation is performed using our existing SOTA STRNet model. The focus of this work is on the development of the enhanced 3D reconstruction method, which enables clear and comprehensive 3D damage mapping for holistic understanding of the detected damage distribution across the entire structure of interest. This can contribute to more systematic management and automated SHM systems in the future.

The comprehensive workflow of the proposed approach is depicted Fig. 1. First, RGB images of the target structure are captured using a handheld camera. These images are then processed in Metashape software (Agisoft Metashape Professional Edition, version 2.1) [1] to extract the camera parameters using structure-from-motion (SfM) techniques, including ground control points (GCPs). In this step, SfM and GCPs are utilized solely for extracting the extrinsic camera parameters, as shown in Fig. 1.

The extracted camera parameters are subsequently transmitted to ABM-Nerfacto model for 3D reconstruction. Depending on the objectives, the raw RGB images or the damage-segmented images can be used for the 3D reconstruction. In this paper, the focus is on pixel-wise damage mapping on the 3D reconstructed model, so the damage-segmented images are used as input to the 3D reconstruction process. The STRNet model is employed for the initial damage segmentation. Finally, the pixel-wise damage mapping is rendered on the 3D reconstructed model using the Nerfstudio framework [30]. The specifics of each technique are outlined in the subsequent subsections.

2.1. SfM

The collected RGB images have their own locations with local coordinate systems (LCS) corresponding to the camera's field of view and orientation. These LCS will be used to generate a 3D reconstruction using an ABM-Nerfacto model. To convert all this image information from the LCS to a global coordinate system (GCS), the intrinsic and extrinsic parameters for each image must be extracted.

The intrinsic camera matrix $\boldsymbol{K}$ (Eq. (1)) and the extrinsic rotation and translation matrix $\boldsymbol{R}|\boldsymbol{t}$ (Eq. (2)) are required for this LCS to GCS conversion.

$$\boldsymbol{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

$$\boldsymbol{R}|\boldsymbol{t} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

In Eq. (1), $\left[f_x, f_y\right]$ represent the focal length, $\left[c_x, c_y\right]$ are the principal point coordinates corresponding to the camera's image sensor size, and s expresses the level of camera lens distortion. Eq. (2) describes the camera position transformation matrix $r_{i,j}$, which converts 3D points from the 3D GCS (X_G, Y_G, Z_G) to the LCS (x_c, y_c, z_c), and the translation vector $\boldsymbol{t} = \left[t_x, t_y, t_z\right]^T$, which represents the camera's position in the GCS. Eq. (3) relates the normalized image coordinates (x_n, y_n) to an arbitrary 3D point $[x_c, y_c, z_c]^T$ in the LCS, where z_c is the distance of the object from the camera's focus.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = z_c \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}. \tag{3}$$

Using the intrinsic camera matrix $\boldsymbol{K}$, the normalized image coordinates $[x_n, y_n, 1]^T$ in Eq. (3) are projected onto the camera's image sensor, resulting in the pixel coordinates $\left[x_p, y_p, 1\right]^T$ as shown in Eq. (4).

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \boldsymbol{K} \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}. \tag{4}$$

Integrating Eqs. (3)–(4), the 3D spatial coordinates $[x_c, y_c, z_c]^T$ can be transformed to the pixel coordinates as presented in Eq. (5).

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = z_c \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} = z_c \boldsymbol{K}^{-1} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}. \tag{5}$$

Finally, a 3D point in the LCS can be converted to a 3D point in the GCS using the extrinsic parameter matrix $\boldsymbol{R}|\boldsymbol{t}$, as described in Eq. (6). The rotation matrix $\boldsymbol{R}$ is composed of three rotation matrices around the X, Y, and Z axes, respectively.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = [\boldsymbol{R}|\boldsymbol{t}] \begin{bmatrix} X_G \\ Y_G \\ Z_G \\ 1 \end{bmatrix}, \tag{6}$$

where rotation matrix $\boldsymbol{R}$ is $\boldsymbol{R} = \begin{bmatrix} cos\theta_z & -sin\theta_z & 0 \\ sin\theta_z & cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} cos\theta_y & 0 & sin\theta_y \\ 0 & 1 & 0 \\ -sin\theta_y & 0 & cos\theta_y \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta_x & -sin\theta_x \\ 0 & -sin\theta_x & cos\theta_x \end{bmatrix}$.

In the original Nerfacto, the global directional vector $\boldsymbol{d}_k$ of each pixel (i.e., ray) is calculated using the rotation matrix ($\boldsymbol{R}$) extracted from SfM supported by ground control points (GCPs) as presented n Eq. (7). This directional vector, denoted as ($\boldsymbol{d}_k$), is utilized in Nerfacto to generate the input for the deep neural networks within the model, as illustrated in Fig. 2.

$$\boldsymbol{R}^T \times \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} = \boldsymbol{d}_k = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \tag{7}$$

2.2. ABM-Nerfacto for 3D reconstruction

The fundamental concept of the NeRF-based [26] Nerfacto model [30] is that the embedded deep neural networks (DNNs) learn the level of transparency and RGB values for each 3D sampled points along the ray of each pixel by comparing the predicted values from the DNNs with

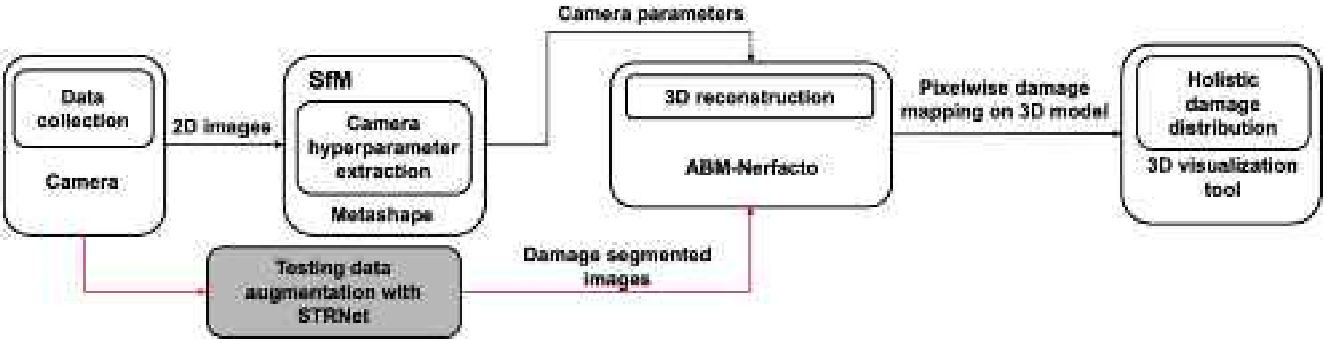

Camera parameters
Data collection
Camera
2D images
SfM
Camera hyperparameter extraction
Metashape
3D reconstruction
ABM-Nerfacto
Pixelwise damage mapping on 3D model
Holistic damage distribution
3D visualization tool
Testing data augmentation with STRNet
Damage segmented images

Fig. 1. Methodology framework.

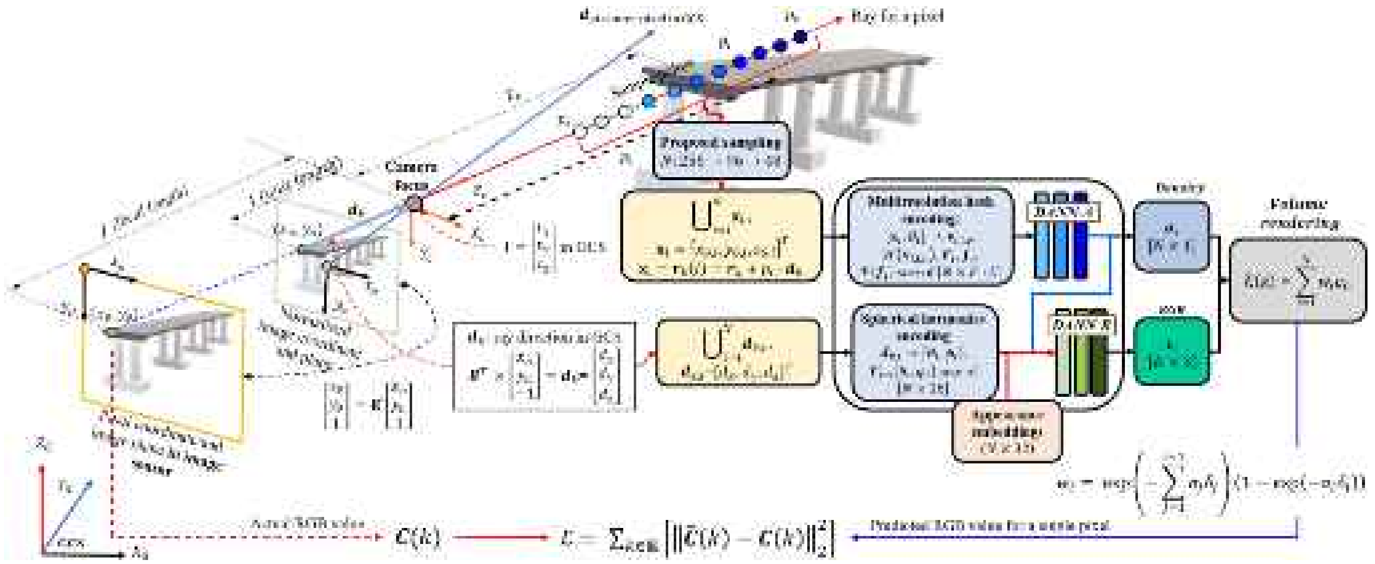


Fig. 2. Schematic view of proposed ABM-Nerfacto model.

the collected RGB images during the training process, enabling 3D reconstruction. As discussed in Eq. (4), the intrinsic parameter $\boldsymbol{K}$ is necessary for converting the 'pixel coordinates and image plane on the image sensor' to 'normalized image coordinates and plane,' as depicted in Fig. 2. Using this new local coordinate system and camera focus shown in Fig. 2, the directional vector ($\boldsymbol{d}_{k,i} = \left[d_x, d_y, d_z\right]^T$) for pixel i is determined using the rotation matrix $\boldsymbol{R}$ presented in Eq. (7). In each 2D image, it is assumed that there is a series of 3D sample points along each ray (pixel) between the camera's focus and the object, as shown in Fig. 2. To sample these 3D points, the original Nerfacto model employs a uniform random sampling and then it utilizes proposal sampling technique [5], which selectively samples the 3D scene in an intelligent and targeted manner. These selected 3D sample points $\mathbf{x}_i$ in the GCS have three coordinates and a directional vector $\boldsymbol{d}_{k,i} = \left[d_x, d_y, d_z\right]^T$. The collection of these directional vectors for all the 3D sample points must be efficiently encoded to be used as input to the embedded DNNs. The original Nerfacto model employs three advanced encoding methods that were not used in the original NeRF model: multiresolution hash encoding [27], spherical harmonics encoding [31], and appearance embedding [25]. These encoding techniques allow for more efficient and effective representation of the 3D scene information compared to the original NeRF model.

The proposal sampling technique employed by the original Nerfacto model enables it to sample the 3D points in a more intelligent and targeted manner. This allows the model to focus on the most important and informative regions of the 3D scene. As a result, the original Nerfacto model can represent complex scenes more accurately than the NeRF model. The original Nerfacto model utilizes a multiresolution hash encoding to compactly represent the 3D sample points' coordinate inputs. This approach allows for significant improvements in inference speed compared to previous NeRF methods, enabling faster and more efficient neural radiance fields rendering.

The key role of the spherical harmonics encoding that is used in original Nerfacto model is to efficiently capture the view direction information. This allows the model to effectively represent the angular dependence (i.e., directional information) of the radiance fields, which is essential for producing high-quality novel view synthesis results.

The appearance embedding, on the other hand, allows the model to capture and represent factors like material properties, texture, color, and other attributes that contribute to the overall appearance. This enables the original Nerfacto model to flexibly render objects with varying appearance properties, making it more versatile and able to handle a wider range of scene compositions and material variations.

These efficiently and compactly encoded coordinate values and directional vectors of the pseudo 3D sample points in each ray of the 2D input image are fed into the deep neural networks (DNNs) separately, as shown in Fig. 2. Specifically, the ABM-Nerfacto model proposed in this paper employs deep attention neural networks (DANN-A and DANN-B) to improve the prediction quality of the opacity (density) and RGB values for each pixel, as illustrated in Fig. 2. The original Nerfacto consists of two separate deep neural networks, DNN-1 and DNN-2, for predicting density and color for each pixel, respectively. While these networks were designed to improve upon the limitations of the original NeRF for large-scale object 3D reconstruction, they remain inadequate for the specific problem addressed in this paper. Therefore, we developed advanced DANN-A and DANN-B, which incorporate an attention module to enhance the quality of 3D reconstruction for the structure of interest in this study.

The process involves feeding the encoded coordinate values and directional vectors of the pseudo 3D sample points in each ray of the 2D input image into the DANN models separately. The DANN models then use their deep attention mechanisms to enhance the prediction of the opacity and RGB values for each pixel. The predicted RGB value for a single pixel is then compared to the actual pixel values to enable backpropagation and training of the DANN models. This iterative process allows the ABM-Nerfacto model to refine its predictions and improve the overall quality of the rendered images.

The subsequent sections will delve into the detailed structure and inner workings of the proposed ABM-Nerfacto model, providing a comprehensive understanding of this novel approach.

2.2.1. *Proposal sampling*

The traditional NeRF model employs a uniform sampling approach, where it samples points along each ray and processes them through a deep neural network (DNN). This is followed by Probability Density Function (PDF) sampling and another pass through the same DNN. However, this approach can be computationally demanding and inefficient, particularly when dealing with large and complex scenes.

To address these limitations, the Nerfacto model introduces a proposal sampling technique, as depicted in Fig. 2. This proposal sampling approach identifies the regions of the scene that are most important for rendering, such as object boundaries, high-contrast areas, or regions with complex geometry. It then allocates more samples to these critical regions, while reducing the number of samples in less important areas. As shown in Fig. 3, the initial set of 512 3D sample points is selectively

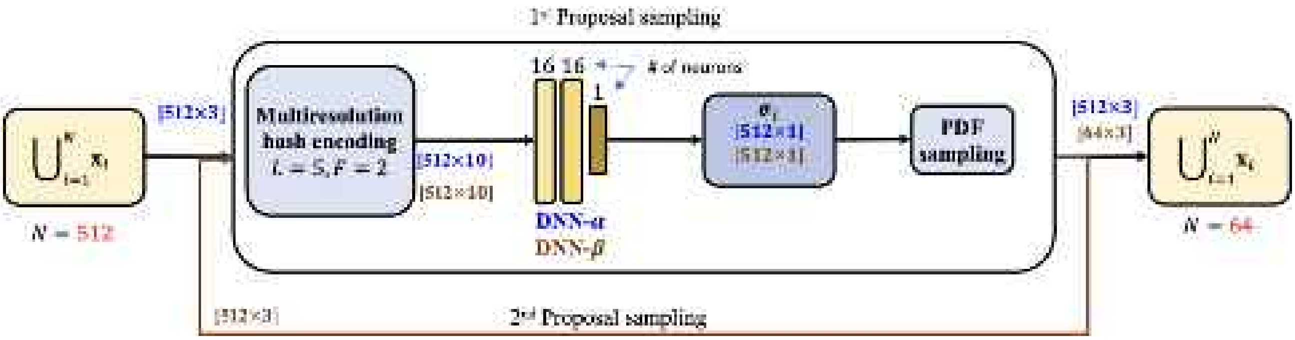


Fig. 3. Proposal sampling network structure.

reduced to 64 points before they are processed by the DANN-A network, as illustrated in Fig. 2. This selective and adaptive sampling strategy allows the Nerfacto model to focus its computational resources on the most relevant parts of the scene, leading to significant efficiency gains.

By prioritizing the sampling of important regions, the ABM-Nerfacto model can represent complex scenes more effectively than the traditional NeRF model. This improved efficiency and scene representation make the model a more practical and effective solution, particularly for large-scale and complex 3D rendering tasks. The proposal sampling module in the Nerfacto model is schematically depicted in Fig. 3. This module comprises two distinct deep neural networks, referred to as DNN-α and DNN-β, each having three layers. The process begins by batching together the initial $N = 512$ 3D sampling points ($\mathbf{x}_i$). These points first undergo a multiresolution hash encoding, which is detailed in the subsequent section. The results of the hash encoding are then input into the DNN-α network, which predict the corresponding density values ($\boldsymbol{\sigma}_i$) for each sampling point.

A Probability Density Function (PDF) sampling process is then performed on these density predictions, yielding 512 new 3D sampling points. This proposal sampling stage using DNN-α is followed by a second stage utilizing the DNN-β network. The DNN-β network takes the 512 sampling points generated by the previous step and further refines them, ultimately producing a set of 64 new sample points. This two-stage proposal sampling approach leverages the complementary strengths of the DNN-α and DNN-β networks to efficiently identify and focus on the most salient regions of the 3D scene. By selectively sampling the 3D space in this manner, the Nerfacto model can allocate computational resources more effectively, leading to improved rendering quality and efficiency, especially for complex scenes.

The PDF sampling used in this proposal sampling technique incorporates a PDF for normalization, as outlined in Eq. (8):

$$PDF_i = \widehat{\chi}_i = \frac{\chi_i}{\sum \chi} \tag{8}$$

where $\chi_i = exp\left(- \sum_{j=1}^{i-1} \boldsymbol{\sigma}_j \delta_j \right) (1 - exp(- \boldsymbol{\sigma}_i \delta_i))$. This formula calculates the normalized PDF value for each sampling point. Following this, the Cumulative Distribution Function (CDF) is employed to compute the cumulative values, as illustrated in Eq. (9):

$$CDF_i = \sum_{j=1}^{i} \widehat{\chi}_j \tag{9}$$

Following this, 512 and 64 random values are generated for the first and second sampling stages, respectively, where the random values (u_τ) are drawn from a uniform distribution $\mho(0,1) = \{\mathrm{u}_1, \mathrm{u}_2 \cdots \mathrm{u}_\tau\}$, and τ ranges from 1 to 512 and 1 to 64. For improvement, 512 and 64 new sampling points are chosen, each associated with a distinct interval defined by the CDF. For instance, if a random value (u_τ) reside within a specific interval ($CDF_{N=1} \leq u_\tau \leq CDF_{N=N}$), the relevant points ($p_i$) and ($p_{i+1}$) are pinpointed. Within this interval, a new sampling distance (p'_τ) is derived through interpolation, as delineated Equation (10):

$$p'_\tau = p_i + \left(\frac{u_\tau - CDF_i}{CDF_{i+1} - CDF_i} \right) \bullet (p_{i+1} - p_i) \tag{10}$$

This PDF sampling produces new distances $\{p'_1, p'_2 \cdots p'_\tau\}$. Utilizing these newly calculated distances, the corresponding 3D sample points are determined using the same formula as previously. These new sample points are then subjected to a process same to the initial coarse estimation, resulting in revised color and density values for the τ sample points.

By employing this technique, the proposal sampling method assists in pinpointing high-density samples via DNNs and the PDF, effectively reducing the total number of 3D sample points to $N = 64$. This decrease in sample points is instrumental in lowering the computational demands for the entire Nerfacto model by effectively allocating more samples to the important regions while reducing the sample numbers in less critical areas. This is especially apparent in comparison to the NeRF model, which utilizes $N = 128$ sample points.

2.2.2. Multiresolution hash encoding

Original Nerfacto adopts a multiresolution hash encoding method that uses a multi-scale hash encoding to compactly represent the coordinate inputs of the newly selected 3D sample points obtained through proposal sampling. This approach allows for significant improvements in inference speed and enables faster and more efficient neural radiance field rendering.

The encoded values are then fed into the DANN-A as depicted in Fig. 4. Initially, within a 3D space, for a sample point $\mathbf{x}_i$ on a specific k^{th} pixel's ray, various voxel sizes are created at each level l using a grid size ratio B_l, with the maximum-level L dictating the resolution. The grid size ratio B_l is calculated using Eq. (11):

$$B_l := \lfloor B_{min} \bullet b^l \rfloor, \text{where } b := exp\left(\frac{lnB_{max} - lnB_{min}}{L - 1} \right). \tag{11}$$

Here, $\lfloor \bullet \rfloor$ represents the floor function, "$:=$" denotes "is defined as", B_{min} is the base resolution, and b is a growth factor. B_{min} and B_{max} are defined by the user, establish the largest and smallest voxel sizes respectively. These voxels are generated in a 3D pseudo-space surrounding $\mathbf{x}_i$, as depicted in Fig. 4. The maximum level L can also be arbitrarily set.

With a 3D sample point's coordinates $\mathbf{x}_i = \left[x_{G,i}, y_{G,i}, z_{G,i} \right]^T$, the 3D coordinates for eight neighboring (voxel) vertices are derived based on the voxel size at each grid level (resolution), as illustrated in Fig. 4. These coordinates are subsequently transformed into hash key values $\mathscr{H}(\mathbf{v}_{i,l,v})$ for the hash table $\boldsymbol{F}_l$ using the hash function $\mathscr{H}$, as shown in Eq. (12):

$$\mathscr{H}(\mathbf{v}_{i,l,v}) = \left(\underset{h=1}{\overset{H}{\boxtimes}} \ generated \ \mathbf{v}_{i,l,v} \cdot \pi_h \right) mod \ T \tag{12}$$

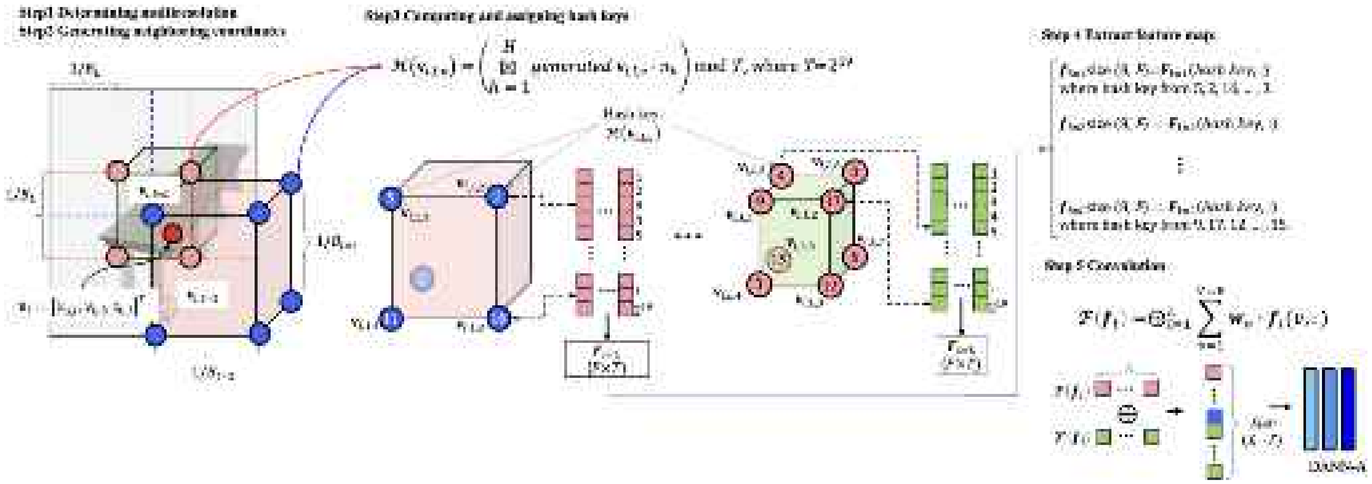

Step1 Determining multiresolution
Step2 Generating neighboring coordinates
Step3 Computing and assigning hash keys
Hash key
Step 4 Extract feature maps
Step 5 Convolution
DANN-A

Fig. 4. Overview of multiresolution hash encoding process.

Here, l represents the resolution level, $H(= 3)$ denotes the number of 3D coordinate dimensions, π are large prime numbers, and $\boxtimes$ signifies the bitwise XOR operation. The entries of the hash table $\boldsymbol{F}_l$, with a size of $F \times T$, are initially randomized as depicted in Fig. 4. In this case, $F = 2$, and $T = 2^{19}$, but these values can be set to any values as long as T is huge number to accommodate the features of all 3d sample points cases voxels' eight hash keys. The dimensionality of each hash key's feature map is also arbitrarily defined by the user.

To derive the final feature vectors, $\mathscr{F}(\boldsymbol{f}_l)$, a convolution operation is conducted on the feature vectors positioned at each vertex of the voxel delineated by $\mathbf{x}_i$'s location within its respective space. This process utilizes the computed feature vectors, $\boldsymbol{f}_l$, and their corresponding weights, $\boldsymbol{w}_v$, to interpolate the final feature vector, which will be used as input for the DANN-A. This convolution formula, incorporating concatenation ($\oplus$), is outlined in Eq. (13):

$$\mathscr{F}(\boldsymbol{f}_l) = \oplus_{l=1}^{L} \sum_{v=1}^{V=8} \boldsymbol{w}_v \bullet \boldsymbol{f}_l(v, :) \tag{13}$$

where $\boldsymbol{w}_v$ is each dimension's convolution weight, derived using Eq. (14):

$$\mathbf{w}_l : = \mathbf{x}_{i,l} - \lfloor \mathbf{x}_{i,l} \rfloor = \mathbf{x}_i \bullet B_l - \lfloor \mathbf{x}_i \bullet B_l \rfloor$$

$$\mathbf{w}_l = (w_x, w_y, w_z) \tag{14}$$

In this context, $\mathbf{w}_l$ encompasses the coordinates of the sample point $\mathbf{x}_{i,l}$. Consequently, the generated input, which has a dimension of $L \bullet F$, is contingent upon the 3D coordinates of the particular sample point along a ray.

This method enables a compact representation of the 3D coordinates of the sample points selected from the proportional sampling process, allowing for faster inference and rendering. This makes it possible to process large-scale environments and intricate geometries efficiently. However, while the multiresolution hash encoding effectively handles the spatial data, it has the potential to lose precision in the representation of the 3D coordinates. As a result, it does not consider the intricate interactions of light and color within a scene as influenced by varying viewing angles, which are essential for attaining photorealistic renderings. To address this limitation, the original Nerfacto model incorporates spherical harmonics encoding. This additional encoding technique helps capture the complex lighting and color information that is essential for producing high-quality, photorealistic results, even in the presence of diverse viewing directions within the scene.

2.2.3. *Spherical harmonics encoding*

The original Nerfacto model leverages spherical harmonic functions [31] to efficiently encode the view direction information. This encoding approach enables the model to effectively capture and represent the angular dependence (i.e., directional information) of the radiance field, which is crucial for producing high-quality novel view synthesis results. The encoded values obtained through the spherical harmonics encoding are then fed as input to the DANN-B. From the $\boldsymbol{d}_{k,i}$ values, the model computes new viewing directional angles, $\theta_i = tan^{-1}\left(\frac{d_y}{d_x}\right)$ and $\phi_i = cos^{-1}\left(\frac{d_z}{d_x^2+d_y^2+d_z^2}\right)$, in the X-Y plane and Z-plane, respectively. These angles are represented in the spherical coordinate system (SCS). To encode the various complex angular information, the model computes the spherical harmonics function $\boldsymbol{Y}_{sm}^{i}(\theta_i, \phi_i)$ using Eq. (15). This spherical harmonics encoding allows the model to effectively represent the intricate directional dependencies within the radiance field.

The spherical harmonics function $\boldsymbol{Y}_{sm}^{i}(\theta_i, \phi_i)$ is defined as:

$$\boldsymbol{Y}_{sm}^{i}(\theta_i, \phi_i) = \begin{cases} (-1)^m \sqrt{\dfrac{2s+1}{4\pi}\dfrac{(s-m)!}{(s+m)!}}\, P_s^m(cos\phi_i)e^{im\theta_i}, if\, m \geq 0 \\ (-1)^m \sqrt{\dfrac{2s+1}{4\pi}\dfrac{(s-m)!}{(s+m)!}}\, P_s^{-m}(cos\phi_i)e^{-im\theta_i}, \qquad if\, m < 0 \end{cases}, \tag{15}$$

Here, s represents the degree of the spherical harmonic function and m is an integer that ranges from $-s\ to\ s$. P_s^m refers to the associated Legendre polynomial, as follows:

$$P_s^m(x) = (1-x^2)^{\frac{m}{2}} \frac{d^m}{dx^m}\left(\frac{1}{2^s s!}\frac{d^s}{dx^s}(x^2-1)^s\right) \tag{16}$$

For a given degree s of the spherical harmonic function, there are $2s+1$ possible values for the integer m. This results in $2s+1$ distinct spherical harmonic functions being generated. Then the number of functions for each degree from 0 to s is summed up to determine the total number of spherical harmonic functions until a maximum degree s. This summation follows the sequence: $1+3+5+\ldots+(2s+1)$, as illustrated in Fig. 5.

Spherical harmonics offer a concise and effective way to represent angular information, rendering it especially proficient in capturing view-dependent effects and complex lighting variations. By converting 3D direction vectors, $\boldsymbol{d}_{k,i}$into spherical coordinates, $[\theta_i, \phi_i]$, and utilizing spherical harmonic functions, $\boldsymbol{Y}_{sm}^{i}(\theta_i, \phi_i)$, Nerfacto encodes directional data in a 16D space when applying spherical harmonics (in ABM-Nerfacto, $s = 3$). This results in a much smaller representation compared to the original NeRF approach.

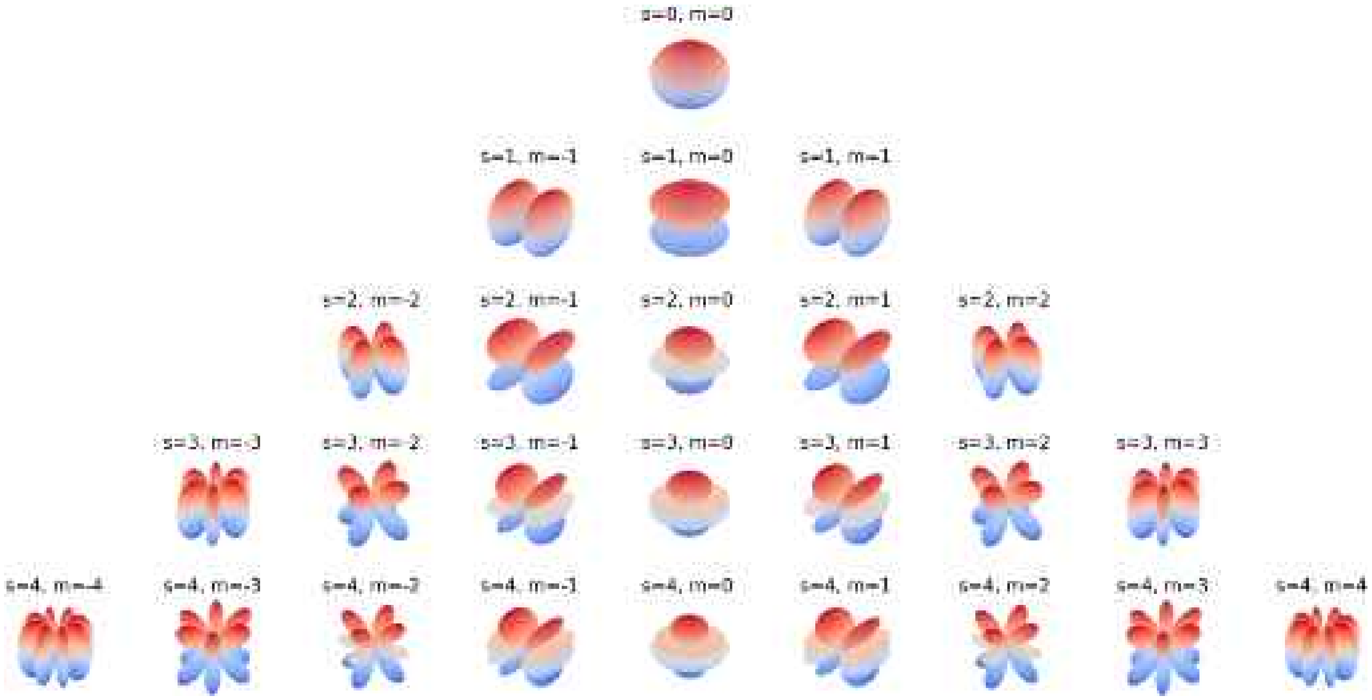


Fig. 5. Visualization of spherical harmonic functions.

2.2.4. *Appearance embedding*

Original Nerfacto model incorporates the appearance embedding [25] that allows the model to capture and represent factors like material properties, texture, color, etc. that contribute to the overall appearance of the generated 3D model. Eventually, this embedding enables the model to flexibly render objects with varying appearance properties, and it makes Nerfacto more versatile. This appearance embedding method empowers model to accommodate variations in the training dataset, such as diverse lighting conditions depicted in Fig. 6. The technique produces a feature vector $\boldsymbol{A}_q$ for each image q, initialized with a $[N \times 128]$ vector of trainable parameters, which is fixed-size (in ABM-Nerfacto).

The ABM-Nerfacto model integrates outputs derived from the spherical harmonics of the ray direction $(N \times 16)$ with the feature vector $(N \times 15)$ obtained from DANN-A through multiresolution hash encoding process. Consequently, the final concatenated input size presented to DANN-B is $N(=64) \times 159$. DANN-B subsequently processes this integrated vector, employing the loss function specified in Eq. (17) to update the model based on the color value of each pixel.

$$\mathscr{L} = \sum_{k \in \mathbb{K}} \left[\| \widehat{\boldsymbol{C}}(k) - \boldsymbol{C}(k) \|_2^2 \right] \tag{17}$$

Here, $\mathbb{K}$ represents the set of all pixels, $\widehat{\boldsymbol{C}}(k)$ is the predicted value, and $\boldsymbol{C}(k)$ is the actual pixel color from the dataset. The loss is calculated as the mean squared error (MSE). The $\boldsymbol{A}_q$ are updated on a pixel-wise basis, but a unique A is generated and updated for each individual image to effectively capture intricate appearance features, including variations in lighting. During the inference phase, the model computes the average of all feature vectors A. This image-dependent methodology allows the Nerfacto model to discern differences between images, thereby enhancing its adaptability to varying lighting conditions. Consequently, the model is capable of producing a more consistent and coherent 3D reconstruction, with improved alignment of lighting and colors across different regions of the object.

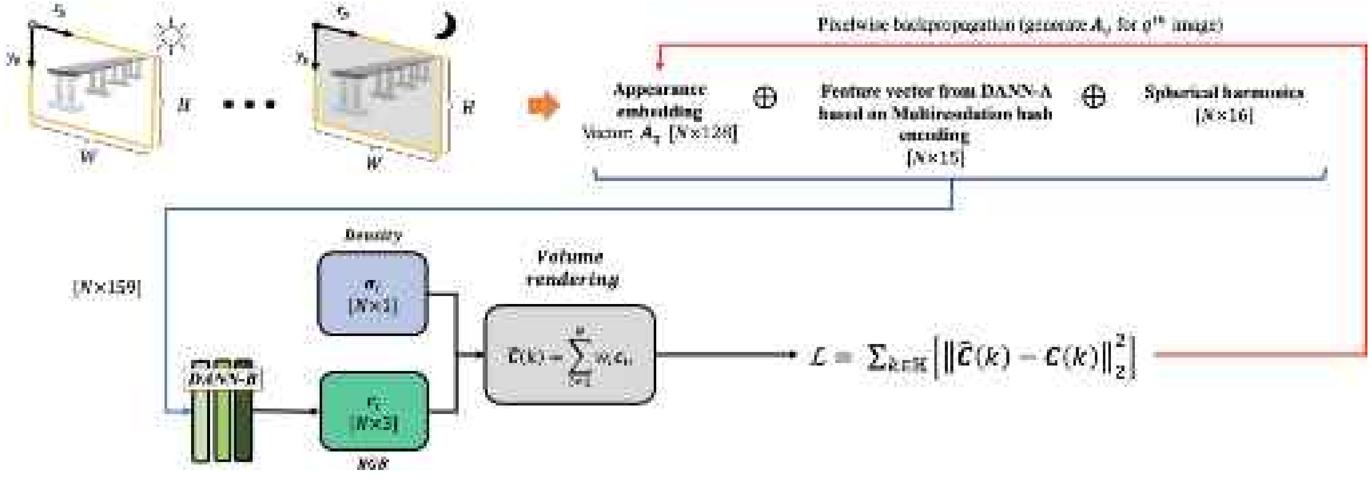

x_b
y_b
H
W
Pixelwise backpropagation (generate A_q for q^{th} image)
Appearance embedding
Vector: A_q [N×128]
⊕
Feature vector from DANN-A based on Multiresolution hash encoding
[N×15]
⊕
Spherical harmonics
[N×16]
[N×159]
DANN-B
Density
σ_i
[N×1]
RGB
c_i
[N×3]
Volume rendering
$\hat{C}(k) = \sum_{i=1}^{N} w_i c_i$
$\mathcal{L} = \sum_{k \in \mathbb{K}} \left[\| \hat{C}(k) - C(k) \|_2^2 \right]$

Fig. 6. Appearance embedding process visualization.

2.2.5. Deep attention neural networks

As depicted in Fig. 2, the expanded inputs derived from the different encoding of the 3D points' position and direction vectors are fed into DANN-A and DANN-B, which predict the color and density of each 3D sampled point along the ray.

Fig. 7 illustrates the structures of the deep attention-based neural networks, DANN-A and DANN-B. Both networks consist of modified hyperparameters from the original Nerfacto model, including 256 neurons and 4 hidden layers, each followed by a rectified linear unit (ReLU). As proposed earlier, a multi-head self-attention module is incorporated into the network.

Initially, $(N \times 32)$ inputs, encoded through multiresolution encoding of the 3D positional vectors, are fed into DANN-A. Each layer within DANN-A produces a $(N \times 256)$ feature map; particularly notable is the feature map produced at the 2^{nd} layer, which is subjected to a multi-head self-attention module. This module generates three distinct $(N \times 256)$ feature maps corresponding to the attention mechanism's query Q, key K, and value V components. Since this multi-head self-attention comprises eight heads, each component is split into eight separate $(N \times 32)$ feature maps. The attention computation for each head follows Eq. (18).

$$Attention = softmax\left(\frac{(QK^T)}{\sqrt{D}_h}\right)V \tag{18}$$

where D_h denotes the dimension of each head.

For each head, the Q $(N \times 32)$ is multiplied by the transposed key K^T through matrix multiplication and then divided by $\sqrt{D_h}$ for scaling, in this case $D_h = 32$. The symbol $\otimes$ in Fig. 7 indicates the multiplications. A softmax function is then applied. The resulting scalars $(N \times N)$ from each head are multiplied by each head's V $(N \times 32)$. The outputs $(N \times 32)$ from all eight heads are concatenated to form a single vector $(N \times 256)$ to reintroduce the input to the 3^{rd} layer. This addition allows the model to focus on different parts of the input data simultaneously, enhancing its ability to learn detailed representations.

In the final stages of processing by DANN-A, both the color density $\boldsymbol{\sigma}$ $(N \times 1)$ and a feature vector with size $(N \times 15)$ are generated simultaneously. To maintain non-negative values, a ReLU activation function is applied to the density values.

Following this, the direction vector's spherical harmonics encoding $(N \times 16)$ and the appearance embeddings $(N \times 128)$ are merged with the $(N \times 15)$ feature vector and processed through DANN-B in a manner similar to DANN-A. The resultant data from DANN-B, activated by a sigmoid function, outputs the RGB color values $\boldsymbol{c}_i$. The composed 4D vector $(\boldsymbol{c}_i, \boldsymbol{\sigma}_i)$ delivers the predicted RGB color and the density values for each 3D sample point coordinate ($\mathbf{x}_i = \left[x_{G,i}, y_{G,i}, z_{G,i}\right]^T$) along the ray.

2.3. STRNet for crack segmentation

To map damages on the 3D reconstruction model, an ABM-Nerfacto model was introduced in Section 2.2. This section outlines the pixel-wise damage segmentation method employed in the paper. As an example, the existing state-of-the-art STRNet model was utilized for damage segmentation, with concrete cracks selected as the target damage type. STRNet, developed by Kang and Cha [19], has demonstrated impressive performance, achieving a mIoU of 92.6 % with 49 FPS processing speed for input image frames of 1280 × 720 resolution. The key benefit of this network is its ability to segment cracks in complex background scenes in real-time. The detailed architecture of the STRNet model is illustrated in Fig. 8.

The STRNet model is specifically designed for the crack segmentation problem, with an optimized encoder and decoder architecture. The encoder is composed of various operators and modules, such as standard convolution layers and the STR Module. The STR Module itself has three different configurations, each with varying hidden layer structures as shown in Fig. 8. These configurations are irregularly repeated 11 times, with different hyperparameters, including various nonlinear activation functions like ReLU, Swish, and H-sigmoid. The first configuration (Config 1) consists of pointwise convolution (PW Conv) and depthwise (DW) Conv. Config 2 includes PW Conv, as well as a squeeze-and-excitation-based attention (SEA) module. Config 3 is composed of PW Conv, DW Conv, the SEA module, and upsampling.

The decoder architecture includes coarse upsampling, a multi-head attention module, and concatenation. All convolutional operators incorporate batch normalization. The decoder performs two types of upsampling: standard and coarse. The conventional upsampling process is executed twice, reinstating the bypassed connections. In the final stage, the outputs from the upsampling, initial convolution, and coarse upsampling blocks are amalgamated. This aggregated output is then passed to a final pointwise convolution, yielding the precise pixel-level segmentation of cracks. Despite the dense and highly complex array of advanced modules, operators, and functions, the STRNet model manages to reduce the total number of learnable parameters to 2 million. This enables the network to process large input frame videos in real-time while achieving state-of-the-art performance in segmenting cracks on complex background scenes.

3. Data

To generate a 3D digital twin and accurately segment cracks, computer vision data is essential. This section outlines the methods used to collect the RGB image data for both the 3D reconstruction and crack

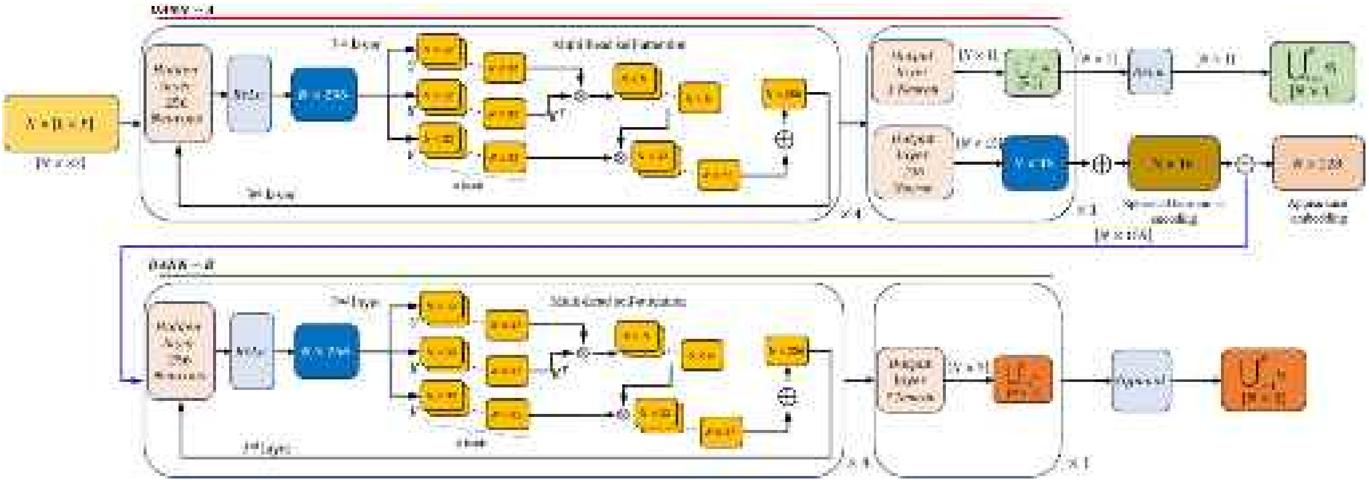


Fig. 7. Deep attention-based neural network structures.

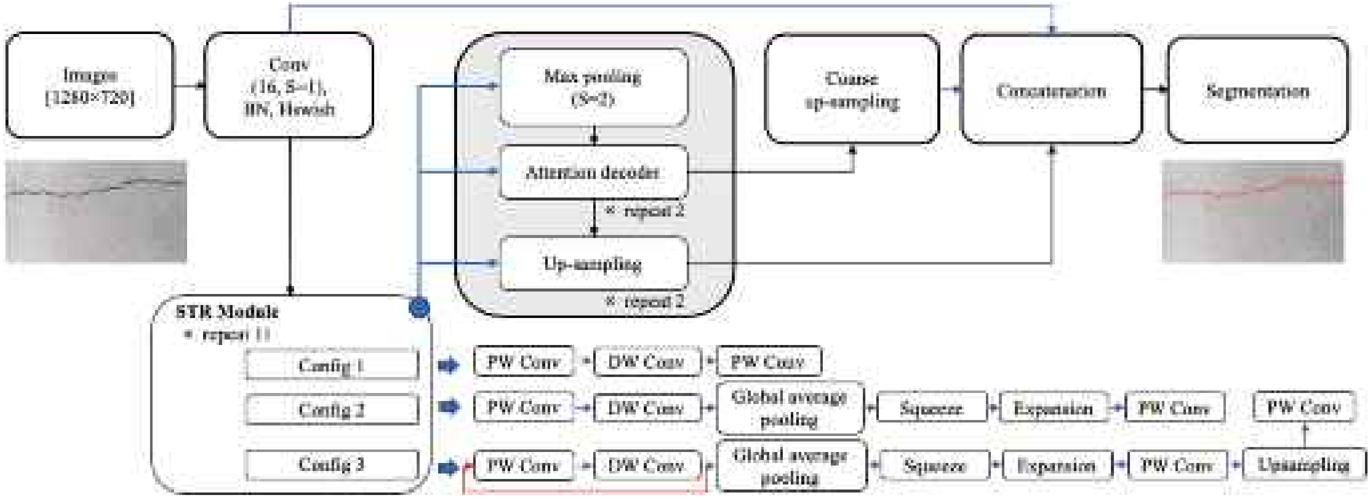


Fig. 8. Architecture of STRNet [19].

segmentation approaches. Section 3.1 describes the data collection method for the 3D reconstruction input to the ABM-Nerfacto model. Section 3.2 explains the data used for crack segmentation using the STRNet model.

3.1. Data collection and SfM for 3D reconstruction

The key components for high-quality 3D reconstruction using the proposed ABM-Nerfacto model are the quality of the RGB images and their accurate intrinsic and extrinsic parameters to convert from each image's LCS to the GCS. To ensure the quality of these components, a Nikon DSLR camera was used to capture a total of 1931 RGB images with a frame size of 2992 × 2000, as presented in Table 1. These images were captured by the camera at a distance ranging from 1 m to 2 m, with at least 70 % overlap between consecutive images to help ABM-Nerfacto learn the features more clearly.

Accurate extraction of the camera parameters is essential for high-quality 3D reconstruction. In this paper, a ground control point (GCP) marker-based approach was adopted to facilitate this process. As shown in Fig. 9, a specific type of marker was placed on the bridge model, as illustrated in Fig. 10. The Metashape software (Agisoft Metashape Professional Edition, version 2.1) was used to implement this process. The use of markers enabled more precise retrieval of the necessary camera parameters, as depicted in Fig. 9.

The GCP markers fulfill a critical role in the 3D reconstruction workflow: they establish accurate reference coordinates and scale, while also aiding in the precise alignment of cameras by offering distinct features for each marker across images. Before photography commenced, coded target markers were strategically positioned within the scene. Within the photogrammetry software, these markers were leveraged not only for reference but also as key elements in the image matching process. To adhere to established best practices, the presence of at least one marker in every photographed image significantly enhances the accuracy of feature matching and spatial orientation.

The bridge system utilized in this study comprises a deck width of 118 cm, a total length of 921 cm with three simply supported sections, and piers that rise to a height of 124 cm, including the deck thickness.

Table 1
Collected RGB images for 3D reconstruction.

Model name	Image size	# of images	Focal length	File format
Nikon D7200	2992 × 2000	1931	18 mm	JPEG

3.2. Data for STRNet

For concrete crack segmentation, the STRNet model was employed and trained using previously established datasets with varying resolutions (1024 × 512 and 1280 × 720), initially acquired by Kang and Cha [19]. The training and test dataset was prepared by manually labeling all pixels associated with concrete cracks using the Procreate software on an Apple iPad Pro with an Apple pencil. The data for the 3D reconstruction were used for image synthesis augmentation technique and was additionally included for training of the STRNet to enable the model to learn the new environment. In total, 2406 images were used for training the STRNet, and 31 images were used for testing. Table 2 provides details on the number of images and their sizes for each case. Fig. 11 illustrates examples of concrete crack images used to train and test the STRNet model. As indicated at the bottom of Fig. 11, we created synthesized images by merging images without cracks with images that feature cracks on pure concrete surfaces [19]. This approach allows the network to learn about new background environments.

4. Case studies

The case studies are extensively conducted for 3D reconstruction using the ABM-Nerfacto model, crack segmentation using the STRNet model, and using the crack-segmented images, another 3D reconstruction is performed for damage mapping on the 3D reconstruction model. To carry out all these simulations, a high-performance computer was used, equipped with an AMD Ryzen 3990× CPU and an Nvidia RTX A6000 GPU, running on the Ubuntu 20.04.6 LTS operating system. Section 4.1 presents various case studies for training the ABM-Nerfacto model and their results. Section 4.2 presents the crack segmentation using the STRNet model. Section 4.3 presents the crack mapping on the 3D reconstruction model as the final result.

4.1. 3D reconstruction using ABM-Nerfacto

The ABM-Nerfacto model is implemented through the Nerfstudio framework [30]. Nerfstudio provides a highly flexible platform to conduct various visualizations and simulations through the different NeRF models and its advanced variants. The collected RGB images from Section 3 are plotted using the intrinsic and extrinsic camera parameters obtained through Metashape, based on their global coordinates, as shown in Fig. 12.

4.1.1. Evaluation metrics for 3D reconstruction

To evaluate the quality of the 3D reconstruction through the ABM-

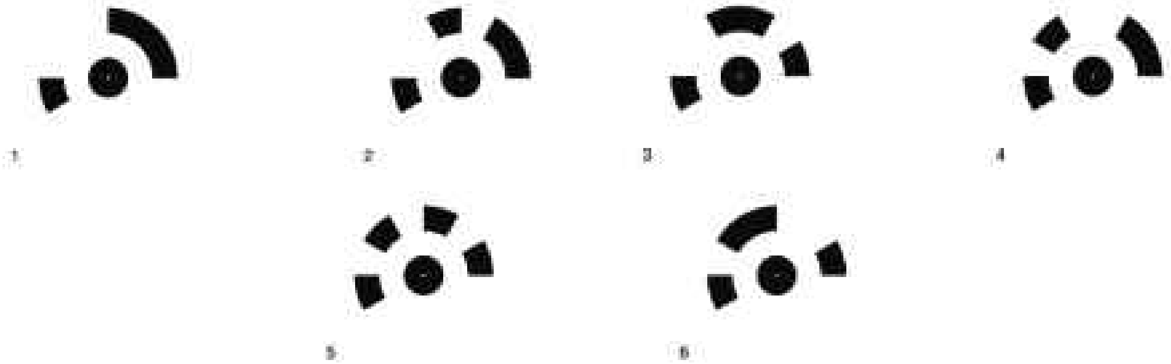
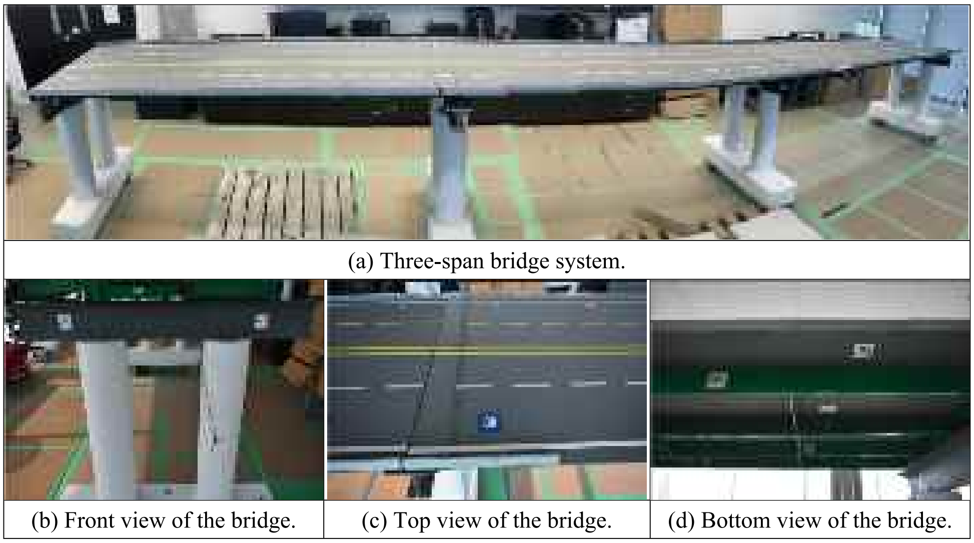


Fig. 9. GCP Markers.



(a) Three-span bridge system.

(b) Front view of the bridge.

(c) Top view of the bridge.

(d) Bottom view of the bridge.

Fig. 10. Aligned GCP Markers on the bridge.

Table 2
Datasets for crack segmentation case studies.

Cases	Image size (pixel)	Number of images
Training	1024 × 512	1203
Training	2992 × 2000	1203
Training total		2406
Test for this new bridge	2992 × 2000	31

Nerfacto model, three different evaluation metrics are employed in this paper: the Mean Squared Error (MSE) Loss function, as presented in Eq. (17), Peak Signal to Noise Ratio (PSNR), as shown in Eq. (19), and Structural Similarity Index (SSIM), as shown in Eq. (20). These evaluation metrics were implemented in the original Nerfacto model [30] to assess the performance of the proposed ABM-Nerfacto.

The Peak Signal to Noise Ratio (PSNR) is calculated as:

$$PSNR = 10log_{10}\left(R^2/MSE\right) \tag{19}$$

where $R = 255$ dB, representing the maximum possible signal value. The Structural Similarity Index (SSIM) is calculated as:

$$SSIM(x,y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)}, \tag{20}$$

where μ_x and μ_y are the pixel sample means of x and y, respectively. σ_x^2 and σ_y^2 are the variances of x and y, and $\sigma_{xy}$is the covariance of x and y. The constants $c_1 = (k_1L)^2$, and, $c_2 = (k_2L)^2$, where L is the dynamic range of the pixel values ((2$^{\text{bits per pixel}}$ − 1), $k_1 = 0.01$, and $k_2 = 0.03$.

PSNR quantifies the ratio between the maximum possible value of a signal and the distortion noise that affects its representation, offering a measure of reconstruction fidelity. SSIM, in contrast, evaluates the perceived quality of images by considering elements such as luminance, contrast, and structural similarity. While both PSNR and SSIM are widely utilized in the field, each metric comes with inherent limitations..

Tancik et al. [30] noted that these quantitative metrics may not accurately reflect the model's performance in real-world scenarios with unstructured data and diverse viewpoints. A more comprehensive assessment should incorporate qualitative evaluation, such as interactive viewers, to gain a deeper understanding of the model's capabilities across various conditions, not just the capture trajectory used for training. While quantitative metrics have value, a holistic approach combining both quantitative and qualitative evaluation is essential for a thorough analysis of neural rendering models.

4.1.2. *Parametric studies of ABM-Nerfacto*

In this section, extensive case studies are conducted to evaluate the performance of the ABM-Nerfacto model for high-quality 3D reconstruction of the structure of interest. As presented in Table 3, a traditional photogrammetry-based 3D reconstruction was also performed, and the visualized results are shown in Fig. 13. This approach is fundamentally different from the original Nerfacto series used in this

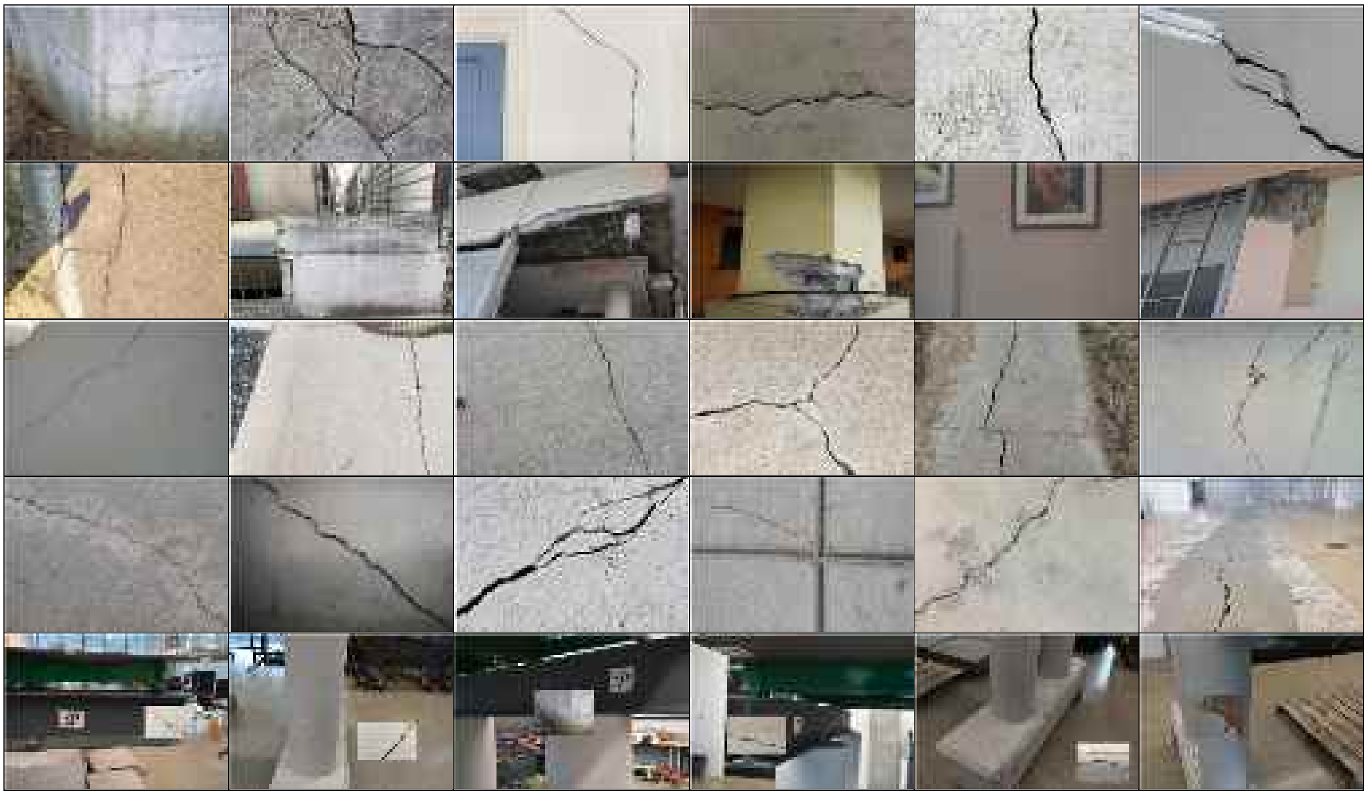


Fig. 11. Concrete cracks images.

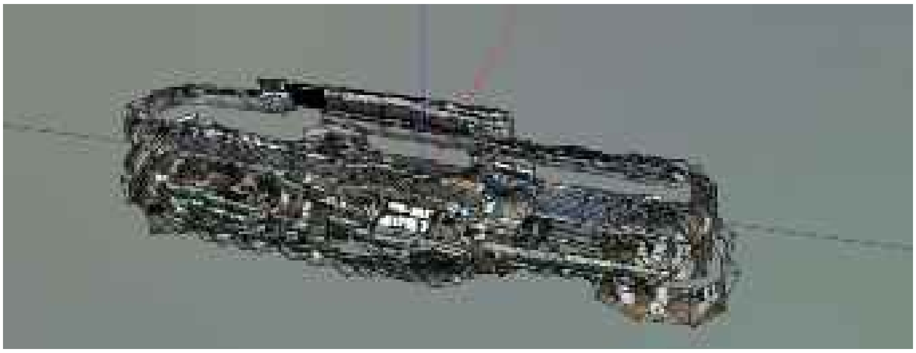


Fig. 12. Global distributions of all RGB input images.

Table 3
Comparative and parametric studies.

Case	Image size (pixel)	# of images	PSNR (db)	SSIM	Loss function (MSE)	Training time (hour)
Case 1: Photogrammetry	2992 × 2000	1931	–	–	–	–
Case 2: Original Nerfacto	2992 × 2000	1931	22.424	0.8069	0.00672	0.81
Case 3: Reduced number of used input images	2992 × 2000	900	23.258	0.7749	0.0058	0.58
Case 4: Original Nerfacto without GCPs	2992 × 2000	1931	20.296	0.7163	0.01045	0.83
Case 5: Modified Nerfacto without attention #1	2992 × 2000	1931	26.570	0.8025	0.00278	3.87
Case 6: Modified Nerfacto without attention #2	2992 × 2000	1931	28.313	0.7913	0.00216	8.14
Case 7: ABM-Nerfacto	2992 × 2000	1931	29.12	08083	0.00195	10.62

work. As a result, the suggested evaluation metrics, such as MSE, PSNR, and SSIM, cannot be directly applied to compare the two methods. However, based on a qualitative visual inspection, the quality of the 3D reconstruction obtained through the traditional photogrammetry approach appears to be quite poor compared to the actual bridge system and the results produced by the series of Nerfacto models presented in this paper. The Nerfacto models, especially the ABM-Nerfacto model, generated more accurate and visually appealing 3D reconstructions. While quantitative metrics provide valuable insights, a comprehensive evaluation should also consider qualitative assessments, such as the visual fidelity and perceived realism of the reconstructed 3D models. The visual comparison between the traditional photogrammetry and the Nerfacto-based approaches highlights the advantages of the latter in capturing the intricate details and overall structure of the bridge system

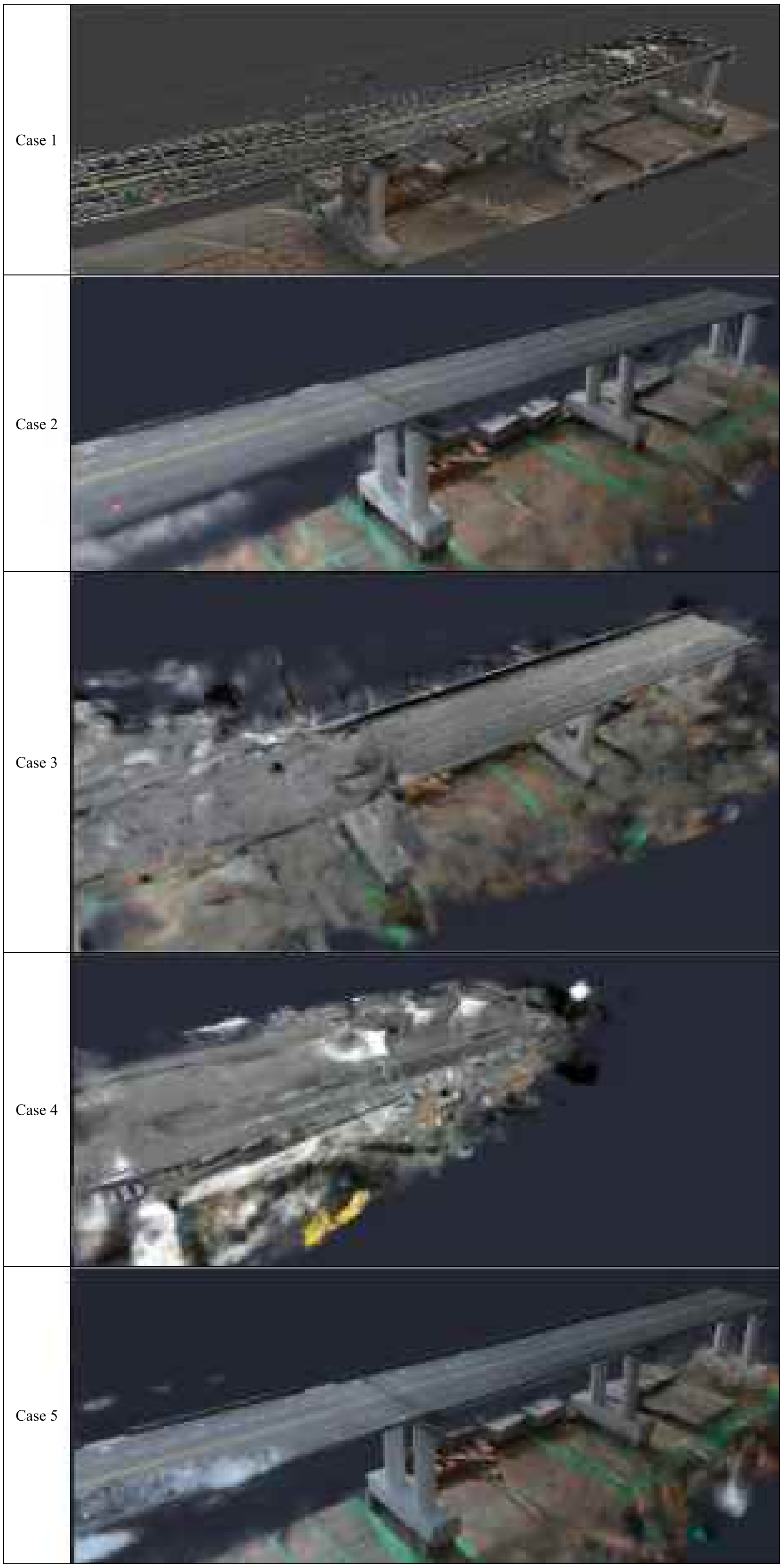


Fig. 13. 3D reconstruction results of each case.

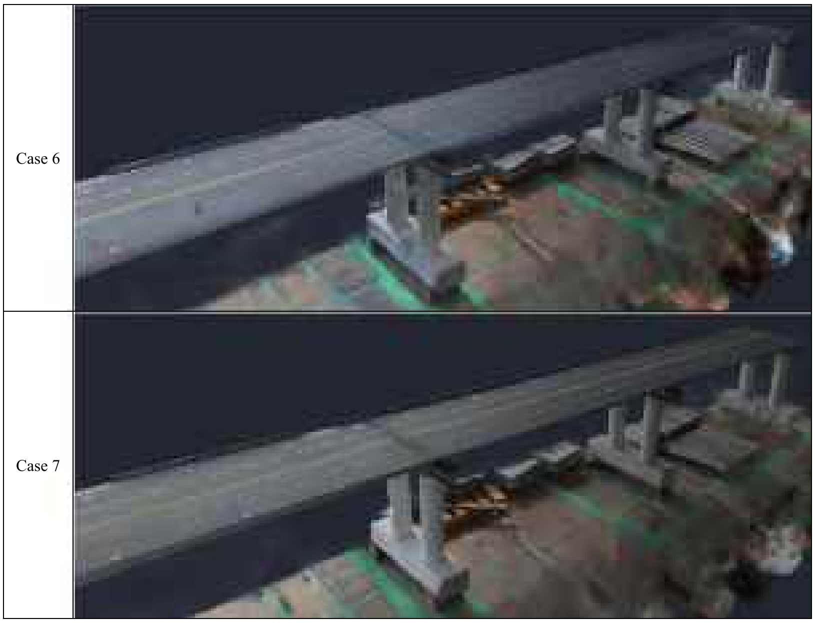


Fig. 13. (*continued*).

more faithfully. Table 3 presents a limited set of parametric studies based on trial-and-error approaches to identify near-optimal architectures and parameters for the proposed ABM-Nerfacto.

In the second case, the original Nerfacto model was implemented without parametric studies, as depicted in Fig. 13. Compared to the first case, it completely reconstructed the entire three-spans bridge, as evident from the visual inspection of the generated 3D reconstruction. The resulting PSNR, SSIM, and MSE values were 22.424, 0.8069, and 0.00672, respectively as presented in Table 3. These values are relatively lower than the third and fourth cases, which are those achieved by the original Nerfacto model with extensive modifications detailed in Table 4. For each subsequent case, significant enhancements were made to the model's hyperparameters and its neural network architecture, DANN-A and DANN-B. These enhancements included increasing the number of iterations, training batch size, 3D sample points, neurons in hidden layers of DANN-A and DANN-B, and the appearance embedding feature vector size. These extensive modifications resulted in a marked improvement in the visualization quality of the 3D reconstructions.

The results of the final case ABM-Nerfacto (Case 7) were drastically improved compared to the original Nerfacto model (Case 2) in terms of the three evaluation metrics (PSNR, SSIM, and MSE) and the visualized results shown in Fig. 13. The significant improvement in the model architecture and hyperparameters with the attention module have led to a substantial enhancement in the quality of the 3D reconstruction.

Table 4
Hyperparametric settings for modified Nerfacto of C4.

	Default Nerfacto	Modified Nerfacto #1	Modified Nerfacto #2
Number of iterations for training	30,000	100,000	100,000
Number of rays (pixels) processed per training batch	4096	8192	16,384
Number of 3D sample points per ray using uniform random sampling	256	256	512
Number of 3D sample points per ray for proposal sampling	256 →96 →48	256 →128 →64	512 →512 →64
Number of 3D sample points per ray for main DANN-A	48	64	64
Hidden layers' dimension of main DANN-A	64	256	256
Hidden layers' dimension of main DANN-B.	64	128	256
Appearance embedding	32	32	128

Through extensive parametric studies and modifications to the original Nerfacto model, including the adoption of multi-headed attention modules in DANN-A and DANN-B, the highest quality of 3D reconstruction was achieved, as evidenced by the three evaluation metrics (PSNR, SSIM, and MSE) presented in Table 3 and the visual inspection presented in Fig. 13.

Fig. 14 presents a comparative visualization of crack details on the first pier across all cases. Notably, Case 7 exhibits the most detailed and accurately reconstructed cracks, significantly enhancing the potential for precise visual damage assessment. This holistic rendering of the bridge system enables engineers to inspect and monitor the structure more systematically and cost-effectively, without the need for on-site visits.

The ability to collect this data through autonomous Unmanned Aerial Vehicles (UAVs) installed in the bridge system is a promising future direction. Such a setup would allow for continuous monitoring and inspection of the bridge structure, reducing the reliance on manual, on-site assessments. The combination of advanced 3D reconstruction techniques, such as the modified Nerfacto model with attention mechanisms, and the integration of autonomous data collection platforms, opens up new opportunities for more efficient and comprehensive structural monitoring. This approach can lead to early detection of potential issues, better-informed maintenance decisions, and ultimately, enhanced safety and longevity of critical infrastructure like the bridge system under investigation.

4.2. Crack segmentation using STRNet

The successful 3D reconstruction of the bridge system has paved the way for structural damage mapping. To demonstrate this capability, the STRNet (Structural Damage Segmentation Network) was employed as an example method for detecting concrete cracks, a common type of structural damage. STRNet was trained and tested using the data presented in Table 2. The training was performed over 1000 epochs, with the loss function being tracked and visualized in Fig. 15. Two separate plots are presented to offer a detailed view of the training loss progression. The first graph represents the training loss across the entire 1000 epochs. The x-axis displays the training iterations, and the y-axis

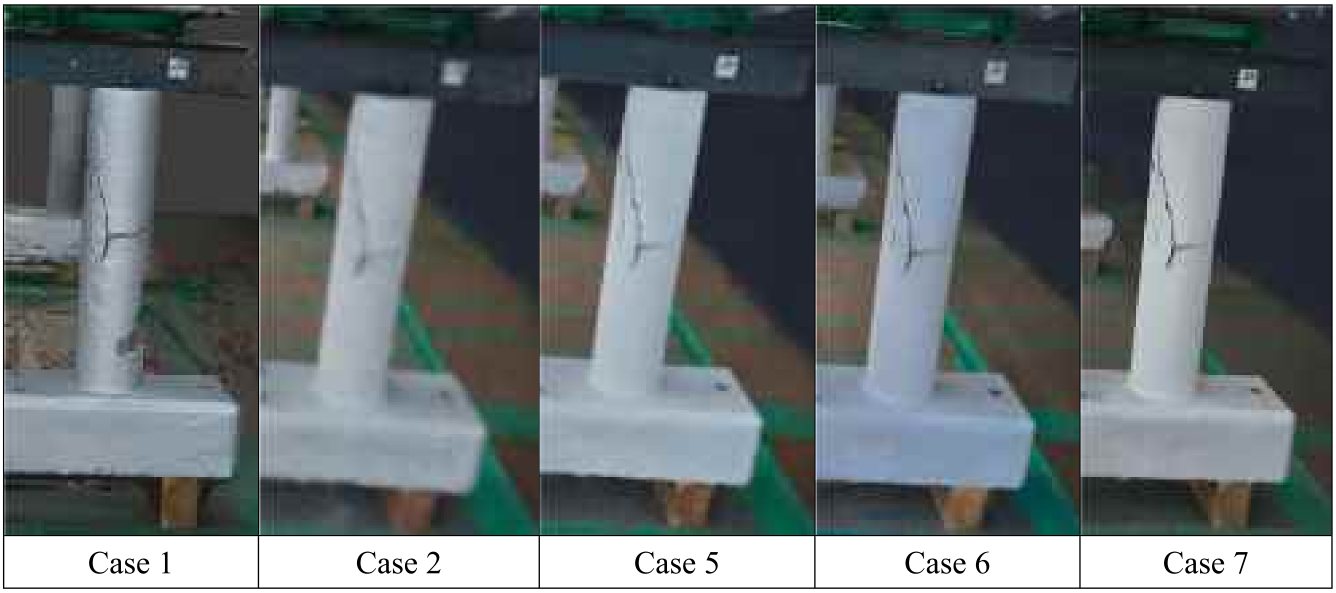


Fig. 14. 3D reconstruction of cracks of each case.

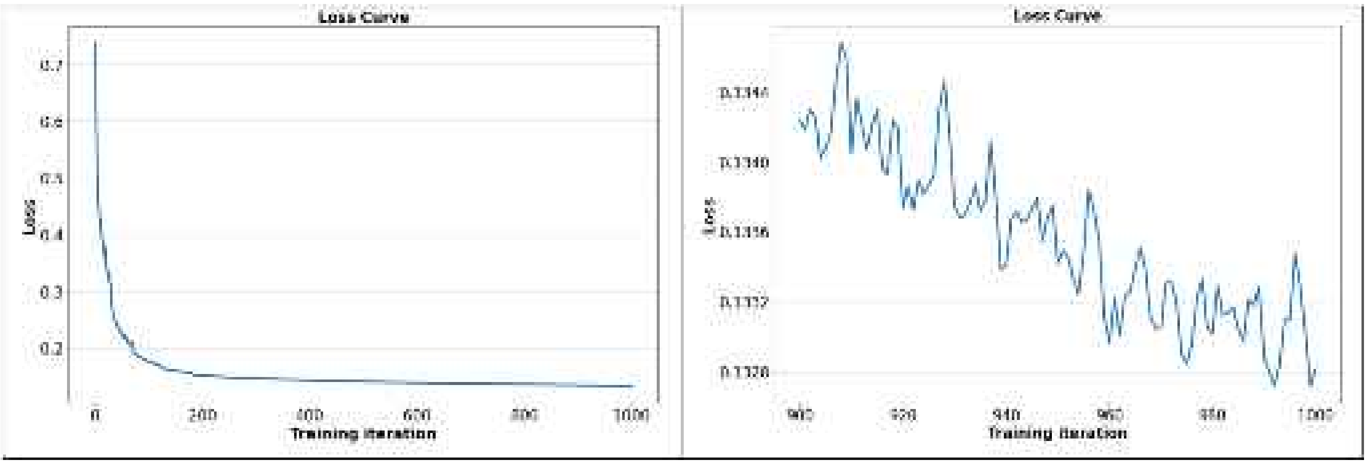


Fig. 15. Training loss function curve.

shows the corresponding loss values. The curve illustrates a rapid decrease in loss during the initial epochs, particularly between epochs 0 and 100. This rapid drop indicates that the model was able to learn a significant portion of the data representation early in the training process. The second graph focuses on the training from epoch 900 to 1000. The loss values decreased more gradually after epoch 500, indicating that the model was approaching its optimal performance. The final loss captured at epoch 1000 is 0.1328, reflects the model's ability to minimize the difference between the predicted and actual data.

The performance of the damage segmentation model was evaluated using various metrics, including:

$$Recall = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{TP_i}{TP_i + FN_i}\right) \tag{21}$$

$$Precision = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{TP_i}{TP_i + FP_i}\right) \tag{22}$$

$$F1 - score = \left(\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \tag{23}$$

$$mIoU = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{TP_i}{TP_i + FP_i + FN_i}\right) \tag{24}$$

where N is the number of objects, TP represents True Positives, FP represents False Positives, and FN represents False Negatives. The test results in Table 5 shows that the STRNet achieved a high mIoU (mean Intersection over Union) percentage, indicating excellent performance in segmenting cracks on the new bridge system. This high-performance crack segmentation is presented in the results shown in Fig. 16, where all the cracks are well-identified by the model.

4.3. Crack mapping on the 3D reconstruction model

The successful 3D reconstruction of the bridge system enabled the mapping of structural damage on the 3D model. To achieve this, the

Table 5
Comparative studies STRNetTTA.

Metrics (%)	Recall	Precision	F1-score	mIoU (%)
STRNetTTA (Test)	94.1	88.1	90.9	91.6

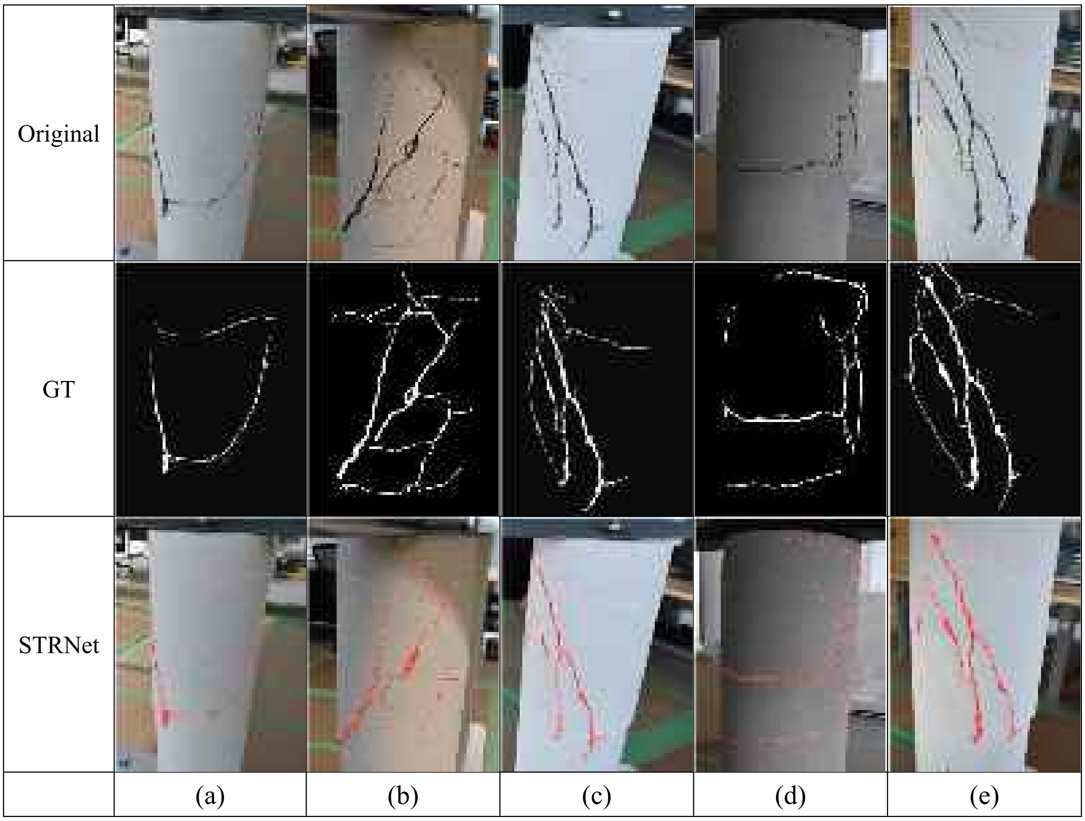


Fig. 16. Crack segmentation results using STRNet.

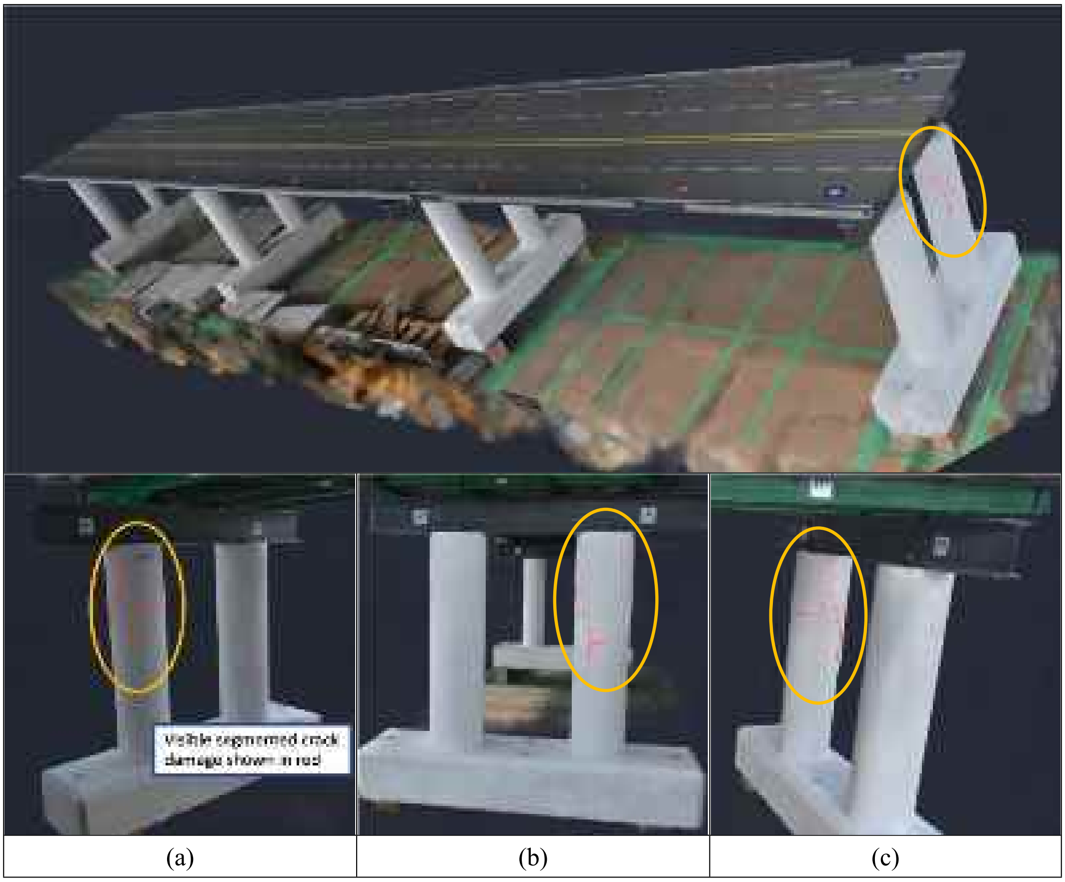


Fig. 17. 3D mapping of the segmented cracks.

images containing cracks from the dataset presented in Table 1 were replaced with the crack segmentation results generated by the well-trained STRNet (Structural Damage Segmentation Network). The ABM-Nerfacto model was then used to generate a 3D reconstruction that incorporated the identified crack damage information, as shown in Fig. 17. This 3D reconstruction accurately mapped all the cracks present on the bridge structure.

The high performance of both the ABM-Nerfacto model for 3D reconstruction and the STRNet for crack segmentation made this integrated approach possible. By leveraging these advanced techniques, the bridge's structural condition can be comprehensively assessed, paving the way for more systematic and automated infrastructure inspection and monitoring.

Building upon these achievements, autonomous UAV systems can be deployed to collect RGB image data of the bridge, which can then be transferred to ground station computers through the Internet of Things (IoT) infrastructure [3,20,34]. The collected data can be processed by trained deep learning networks for damage segmentation, and the resulting damage-annotated images can be used to generate a 3D reconstruction that highlights the structural issues.

This integrated framework, combining autonomous data collection, damage segmentation, and 3D reconstruction, enables a more systematic and efficient approach to infrastructure inspection, monitoring, and management. By automating these processes, bridge owners and operators can make more informed, data-driven decisions regarding maintenance and repair, ultimately ensuring the long-term safety and reliability of critical assets.

5. Conclusions

The paper presents a novel approach to 3D damage mapping on reconstructed 3D models. This technique integrates a neural radiance field-based 3D reconstruction method, specifically the ABM-Nerfacto model, with an advanced deep learning-based structural damage segmentation method, the STRNet. The key findings of this study are:

1) Traditional photogrammetry-based techniques were not suitable for the structure of interest because they failed to provide meaning quality of 3D reconstruction model.
2) Original NeRF model required excessive required computational cost for the structure of interest compared to the proposed ABM-Nerfacto.
3) The original Nerfacto model produced better-quality 3D reconstructions compared to the traditional methods, thanks to its advanced operations such as multiresolution hash encoding, spherical harmonics encoding, and appearance embedding. These techniques effectively encoded the collected RGB images and enabled more computationally efficient and accurate 3D rendering.
4) Despite the improvements offered by the original Nerfacto model, there was still significant room for further enhancement of its performance.
5) Extensive modifications to the original Nerfacto model, including adjustments to hyperparameters, integration of a multi-headed attention module, and comprehensive changes to the embedded DANN-A and DANN-B components, resulted in a much higher-quality 3D reconstruction and corresponding damage mapping.

The proposed ABM-Nerfacto model can be further applied to real-world outdoor infrastructure, which involves larger scales and varying lighting conditions. Both manual and autonomous UAV-based image collection methods can be considered for future work [34]. Furthermore, a more accurate evaluation of metrics for 3D reconstruction based on NeRF, Nerfacto, and their variants should be explored.

This research demonstrates the potential of integrating advanced deep learning techniques with neural radiance field-based 3D reconstruction to enable accurate and comprehensive damage mapping on digital twin models. The findings contribute to the development of more systematic and automated infrastructure inspection and monitoring approaches, ultimately supporting efficient asset management and maintenance.

Funding

The research was supported by the Mitacs L2M program IT34014, Research Manitoba Proof-of-Concept Grant (4914), NSERC Discovery Grant (RGPIN-2022-04120) and the CFI JELF grant, (37394).

Statement

This manuscript (3D Pixelwise Damage Mapping Using a Deep Attention based modified Nerfacto) is our original unpublished work, the manuscript or any variation of it has not been submitted to another publication previously, and there is no conflict of interest.

CRediT authorship contribution statement

Geontae Kim: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Youngjin Cha:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

None.

Data availability

Data will be made available on request.

References

[9] Y.J. Cha, K. You, W. Choi, Vision-based detection of loosened bolts using the Hough transform and support vector machines, in: Automation in Construction, 2016-11 vol.71, Elsevier B.V., Amsterdam, 2016, pp. 181–188, https://doi.org/10.1016/j.autcon.2016.06.008. ISSN: 0926-5805, EISSN: 1872-7891.

[6] Y.J. Cha, R. Ali, J. Lewis, O. Büyüköztürk, Deep learning-based structural health monitoring, in: Automation in construction, 2024-05 vol.161, Elsevier B.V., Amsterdam, 2024, p. 105328. ISSN: 0926-5805, EISSN: 1872-7891. article 105328, https://doi.org/10.1016/j.autcon.2024.105328.

[7] Y.J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, in: Computer-aided Civil and Infrastructure Engineering, 2017-05 vol.32, 2017, pp. 361–378, 5. ISSN: 1093-9687, EISSN: 1467-8667, https://doi.org/10.1111/mice.12263.

[8] Y.J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, O. Büyüköztürk, Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types, in: Computer-aided Civil and Infrastructure Engineering, 2017-05 vol.32, Wiley subscription services, Inc., Hoboken, 2018, pp. 361–378, https://doi.org/10.1111/mice.12263, 5. ISSN: 1093-9687, EISSN: 1467-8667.

[20] D. Kang, Y.J. Cha, Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging, in: Computer-Aided Civil and Infrastructure Engineering, 2018–10 vol.33, Wiley Subscription Services, Inc., Hoboken, 2018, pp. 885–902, https://doi.org/10.1111/mice.12375, 10. ISSN: 1093–9687, EISSN: 1467–8667.

[3] R. Ali, D. Kang, G. Suh, Y.J. Cha, Real-time multiple damage mapping using autonomous UAV and deep faster region-based neural networks for GPS-denied structures, in: Automation in construction, 2021-10 VOL.130, Elsevier B.V., Amsterdam, 2021, p. 103831. ISSN: 0926-5805, EISSN: 1872-7891. article 103831, https://doi.org/10.1016/j.autcon.2021.103831.

[22] S.S. Kumar, D.M. Abraham, M.R. Jahanshahi, T. Iseley, J. Starr, Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks, in: Automation in Construction, 2018-07 vol.91, Elsevier B.V., Amsterdam, 2018, pp. 273–283, https://doi.org/10.1016/j.autcon.2018.03.028. ISSN: 0926-5805, EISSN: 1872-7891.

[4] P. Arafin, A.M. Billah, A. Issa, Deep learning-based concrete defects classification and detection using semantic segmentation, in: Structural health monitoring, 2024-01 vol.23, SAGE publications, London, England, 2024, pp. 383–409, https://doi.org/10.1177/14759217231168212, 1. ISSN: 1475-9217, EISSN: 1741-3168.

[23] Z. Liu, Y. Cao, Y. Wang, W. Wang, Computer vision-based concrete crack detection using U-net fully convolutional networks, in: Automation in Construction, 2019-08 vol.104, Elsevier B.V., Amsterdam, 2019, pp. 129–139,

https://doi.org/10.1016/j.autcon.2019.04.005. ISSN: 0926-5805, EISSN: 1872-7891.

[29] G. Pan, Y. Zheng, S. Guo, Y. Lv, Automatic sewer pipe defect semantic segmentation based on improved U-net, in: Automation in Construction, 2020-11 vol.119, Elsevier B.V., Amsterdam, 2020, p. 103383. ISSN: 0926-5805, EISSN: 1872-7891. article 103383, https://doi.org/10.1016/j.autcon.2020.103383.

[2] N. Alfaz, A. Hasnat, A.M.R.N. Khan, N.S. Sayom, A. Bhowmik, Bridge crack detection using dense convolutional network (densenet), in: Proceedings of the 2nd International Conference on Computing Advancements, ACM, New York, NY, USA, 2022, March, pp. 509–515, https://doi.org/10.1145/3542954.3543027. ISBN: 9781450397346, 1450397344.

[33] Z. Wang, Z. Hu, X. Xiao, Crack detection of Brown Rice kernel based on optimized ResNet-18 network, in: IEEE Access, 2023 vol.11, IEEE, Piscataway, 2023, pp. 140701–140709, https://doi.org/10.1109/ACCESS.2023.3341350. ISSN: 2169-3536, EISSN: 2169-3536.

[21] D. Kang, S.S. Benipal, D.L. Gopal, Y.J. Cha, Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning, in: Automation in Construction, 2020-10 vol.118, Elsevier B.V., Amsterdam, 2020, p. 103291. ISSN: 0926-5805, EISSN: 1872-7891. article 103291, https://doi.org/10.1016/j.autcon.2020.103291.

[24] Z. Liu, J.K. Yeoh, X. Gu, Q. Dong, Y. Chen, W. Wu, et al., Automatic pixel-level detection of vertical cracks in asphalt pavement based on GPR investigation and improved mask R-CNN, in: Automation in Construction, 2023–02 vol.146, Elsevier B.V., 2023, p. 104689. ISSN: 0926–5805, EISSN: 1872–7891. Article 104689, https://doi.org/10.1016/j.autcon.2022.104689.

[11] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV), Cornell University library, Ithaca, 2018, pp. 801–818, https://doi.org/10.48550/arxiv.1802.02611, arXiv.org. EISSN: 2331-8422.

[18] A. Ji, X. Xue, Y. Wang, X. Luo, W. Xue, An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement, in: Automation in Construction, 2020-06 vol.114, Elsevier B.V., Amsterdam, 2020, p. 103176. ISSN: 0926-5805, EISSN: 1872-7891. article 103176, https://doi.org/10.1016/j.autcon.2020.103176.

[35] X. Wu, Z. Feng, J. Liu, H. Chen, Y. Liu, Predicting existing tunnel deformation from adjacent foundation pit construction using hybrid machine learning, Autom. Constr. 165 (2024) 105516, 2024-09. article 105516. Elsevier B.V. ISSN: 0926-5805, EISSN: 1872-7891, https://doi.org/10.1016/j.autcon.2024.105516.

[12] W. Choi, Y.J. Cha, SDDNet: real-time crack segmentation, in: IEEE Transactions on Industrial Electronics, 2020-09 vol.67, IEEE, New York, 2019, pp. 8016–8025, https://doi.org/10.1109/TIE.2019.2945265, 9. ISSN: 0278-0046, EISSN: 1557-9948.

[19] D.H. Kang, Y.J. Cha, Efficient attention-based deep encoder and decoder for automatic crack segmentation, in: Structural Health Monitoring, 2022-09 vol.21, SAGE publications, London, England, 2022, pp. 2190–2205, https://doi.org/10.1177/14759217211053776, 5. ISSN: 1475-9217, EISSN: 1741-3168.

[34] A. Waqas, D. Kang, Y.J. Cha, Deep learning-based obstacle-avoiding autonomous UAVs with fiducial marker-based localization for structural health monitoring, Struct. Health Monit. 23 (2) (2024) 971–990, 2024-03. London, England: SAGE publications. ISSN: 1475-9217, EISSN: 1741-3168, https://doi.org/10.1177/14759217231177314.

[14] W. Gao, C. Zhang, X. Lu, W. Lu, Concrete spalling damage detection and seismic performance evaluation for RC shear walls via 3D reconstruction technique and numerical model updating, in: Automation in Construction, 2023-12 vol.156, Elsevier B.V., Amsterdam, 2023, p. 105146. ISSN: 0926-5805, EISSN: 1872-7891. article 105146, https://doi.org/10.1016/j.autcon.2023.105146.

[38] S. Zhao, F. Kang, J. Li, Concrete dam damage detection and localisation based on YOLOv5s-HSC and photogrammetric 3D reconstruction, Autom. Constr. 143 (2022) 104555, 2022-11. article 104555. ISSN: 0926-5805, https://doi.org/10.1016/j.autcon.2022.104555.

[17] L. Hua, Y. Lu, J. Deng, Z. Shi, D. Shen, 3D reconstruction of concrete defects using optical laser triangulation and modified spacetime analysis, in: Automation in construction, 2022-10 vol.142, 2022, p. 104469, article 104469. ISSN: 0926-5805, https://doi.org/10.1016/j.autcon.2022.104469.

[16] L. Hua, J. Deng, Z. Shi, X. Wang, Y. Lu, Single-stripe-enhanced spacetime stereo reconstruction for concrete defect identification, in: Automation in Construction, 2023-12 vol.156, Elsevier B.V., 2023, p. 105136. ISSN: 0926-5805, EISSN: 1872-7891. article 105136, https://doi.org/10.1016/j.autcon.2023.105136.

[26] B. Mildenhall, P.P. Srinivasan, M. Tancik, J.T. Barron, R. Ramamoorthi, R. Ng, Nerf: representing scenes as neural radiance fields for view synthesis, in: Communications of the ACM, 2022-01 vol.65, ACM, New York, NY, USA, 2021, pp. 99–106, https://doi.org/10.1145/3503250, 1. ISSN: 0001-0782, EISSN: 1557-7317.

[28] A. Pal, J.J. Lin, S.H. Hsieh, M. Golparvar-Fard, Activity-level construction progress monitoring through semantic segmentation of 3D-informed orthographic images, in: Automation in Construction, 2024-01 vol.157, 2024, p. 105157, article 105157. ISSN: 0926-5805, https://doi.org/10.1016/j.autcon.2023.105157.

[37] Z. Yu, Y. Shen, Y. Zhang, Y. Xiang, Automatic crack detection and 3D reconstruction of structural appearance using underwater wall-climbing robot, Autom. Constr. 160 (2024) 105322, 2024-04. article 105322. Elsevier B.V. ISSN: 0926-5805, EISSN: 1872-7891, https://doi.org/10.1016/j.autcon.2024.105322.

[15] S.J. Garbin, M. Kowalski, M. Johnson, J. Shotton, J. Valentin, Fastnerf: high-fidelity neural rendering at 200fps, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, IEEE, 2021, pp. 14326–14335, https://doi.org/10.1109/ICCV48922.2021.01408. EISSN: 2380-7504, EISBN: 9781665428125, EISBN: 1665428120.

[10] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, H. Su, Mvsnerf: fast generalizable radiance field reconstruction from multi-view stereo, in: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), 2021, IEEE, 2021, pp. 14104–14113, https://doi.org/10.1109/ICCV48922.2021.01386. EISSN: 2380-7504, EISBN: 9781665428125.

[32] Q. Wang, Z. Wang, K. Genova, P.P. Srinivasan, H. Zhou, J.T. Barron, et al., Ibrnet: Learning multi-view image-based rendering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Cornell University Library, Ithaca, 2021, pp. 4690–4699, https://doi.org/10.48550/arxiv.2102.13090, arXiv.org. EISSN: 2331–8422.

[36] B. Xiong, W. Jiang, D. Li, M. Qi, Voxel grid-based fast registration of terrestrial point cloud, Remote Sens. (Basel, Switzerland) 13 (10) (2021) 1905, 2021-05. Basel: MDPI AG. ISSN: 2072-4292, EISSN: 2072-4292, https://doi.org/10.3390/rs13101905.

[30] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, et al., Nerfstudio: A modular framework for neural radiance field development, in: ACM SIGGRAPH 2023 Conference Proceedings, ACM, New York, NY, USA, 2023, July, pp. 1–12, https://doi.org/10.1145/3588432.3591516. ISBN: 9798400701597.

[13] Z. Dong, W. Lu, J. Chen, Neural rendering-based semantic point cloud retrieval for indoor construction progress monitoring, in: Automation in Construction, 2024-08 vol.164, Elsevier B.V., Amsterdam, 2024, p. 105448, https://doi.org/10.1016/j.autcon.2024.105448. ISSN: 0926-5805, EISSN: 1872-7891.

[5] J.T. Barron, B. Mildenhall, D. Verbin, P.P. Srinivasan, P. Hedman, Mip-nerf 360: unbounded anti-aliased neural radiance fields, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Cornell University library, Ithaca, 2022, pp. 5470–5479, https://doi.org/10.48550/arxiv.2111.12077. EISSN: 2331-8422.

[27] T. Müller, A. Evans, C. Schied, A. Keller, Instant neural graphics primitives with a multiresolution hash encoding, in: ACM Transactions on Graphics (ToG), 2022-07 vol.41, ACM, New York, NY, USA, 2022, pp. 1–15, 4. ISSN: 0730-0301, EISSN: 1557-7368. article 102, https://doi.org/10.1145/3528223.3530127.

[31] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J.T. Barron, P.P. Srinivasan, Ref-nerf: structured view-dependent appearance for neural radiance fields, in: 2022 IEEE/CVF Conference on Computer Vision and pattern Recognition (CVPR), IEEE, Ithaca, 2022, June, pp. 5481–5490, https://doi.org/10.48550/arxiv.2112.03907. Cornell University library, arXiv.org. EISSN: 2331-8422.

[25] R. Martin-Brualla, N. Radwan, M.S. Sajjadi, J.T. Barron, A. Dosovitskiy, D. Duckworth, Nerf in the wild: neural radiance fields for unconstrained photo collections, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Cornell University library, Ithaca, 2021, pp. 7210–7219, https://doi.org/10.48550/arxiv.2008.02268, arXiv.org. EISSN: 2331-8422.

[1] Agisoft Metashape Professional Edition (version 2.1), Retrieved from, https://www.agisoft.com, 2024.