

3D Damage Mapping and Segmentation Using Neural Radiance Fields and Advanced Deep Learning Techniques

By

Geontae Kim

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfillment of the requirements of the degree of

Master of Science

Department of Civil Engineering,

University of Manitoba,

Winnipeg, Canada.

Copyright © 2024 by Geontae Kim

Abstract

This thesis introduces a novel approach to Structural Health Monitoring (SHM) by integrating deep learning with advanced 3D reconstruction techniques, focusing on the efficient analysis of bridge structures for achieving digital twin in SHM. Traditional photogrammetry faces challenges in accurately reconstructing flat surfaces and rapidly assessing structural health. Addressing these issues, this research adopts the Nerfacto model from the Neural Radiance Fields (NeRF) within the Nerfstudio framework to enhance 3D reconstruction fidelity and utilizes strategically placed markers to improve camera pose accuracy. Additionally, the integration of the STRNet and Test Time Augmentation (TTA) significantly enhances crack detection capabilities. This approach allows for precise mapping of segmented cracks onto a 3D model of a bridge, offering a detailed and quantifiable assessment of structural damage. By combining these innovative technologies, the research provides a scalable, cost-effective solution for comprehensive structural assessments, paving the way for proactive maintenance strategies that ensure the longevity and safety of critical infrastructure. The integration of digital twin technology and detailed damage mapping in 3D also sets a new standard in the field, demonstrating substantial potential for future SHM applications.

Acknowledgement

I would like to express my gratitude to all who supported me during this journey of completing my thesis.

Firstly, I am grateful to my advisor, Dr. Young-Jin Cha, for providing me with the opportunity to embark on this M.Sc. program and pursue this research. Throughout the program, I feel I have truly evolved into a researcher by independently studying and gathering the necessary resources. This experience has empowered me to tackle any challenges that may come my way. I am also thankful to the thesis committee members, Dr. Nan Wu and Dr. Graziano Fiorillo, for their valuable and insightful feedback on my thesis, and participating in my thesis defense. I would also like to extend my thanks to the faculty and staff of the Department of Civil Engineering, especially Jennifer Merrell, who has consistently assisted me with graduate student matters.

My gratitude also goes to my former lab mates—Ali Rahmat, Ali Waqas, John Lewis, and Alireza Mostafavi—and my school friend Bo, Shazaib, and Taimur. I cherish the great memories we have created together, which made my time in University of Manitoba enjoyable.

Lastly, I would like to share this joy with my family: my mother who always loves me, my father who is always proud of me, and my sister who is like a friend. I owe a profound thank you to my partner, Sherry, who has believed in and supported me unconditionally. Without you, I could not have completed this journey. I am also grateful to Sherry's parents, who have always welcomed me in Vancouver, and to Danny and Tuesday, for their companionship.

Table of Contents

Abstract.....	2
Acknowledgement	3
Table of Contents.....	4
List of Tables	6
List of Figures	7
Copyright	9
Terminology Abbreviations	10
1 Introduction	12
1.1 Background	12
1.2 Problem statement.....	16
1.3 Research objective.....	17
1.4 Thesis organization	18
2 Literature review.....	20
2.1 SHM	20
2.2 Computer vision in SHM	21
2.3 Deep learning in SHM.....	22
2.4 3D reconstruction techniques	24
2.5 NeRF	26
2.6 Digital twin in SHM.....	29
3 Methodology.....	33
3.1 Image deblurring with NAFNet	34
3.2 Obtaining camera parameters for NeRF input	37
3.2.1 Camera parameters information with pinhole camera model	37
3.2.2 SfM	41
3.2.3 Camera poses extraction with Metashape.....	44
3.3 TTA-STRNet crack segmentation.....	49
3.3.1 TTA	51
3.3.2 STRNet	53
3.4 NeRF for 3D reconstruction.....	55
3.4.1 NeRF overview	56
3.4.2 NeRF model structure	59

3.4.3	Network structure.....	61
3.4.4	Nerfstudio framework.....	63
3.5	Damage mapping.....	67
4	Data collection.....	70
4.1	Image data collection	71
4.2	Camera specifications and setup	72
4.3	Controlled environment and setup	73
4.4	Marker setting	75
4.5	Image capture methodology	77
5	Case studies	81
5.1	Experimental detail	82
5.2	STRNet + TTA evaluation.....	82
5.3	3D reconstruction	86
5.4	Damage segmented 3d reconstruction result.....	92
6	Conclusion and future work	94
6.1	Conclusion.....	94
6.2	Future work	94
7	References	97

List of Tables

Table 1. Changed hyperparameter values.	64
Table 2. Datasets in each case.....	72
Table 3. Camera settings.....	73
Table 4. Comparison mIoU with original STRNet and STRNet + TTA.....	83
Table 5. The mIoU results of each different TTA approached case.	84
Table 6. Training performance assessment (Kim and Cha, 2024).....	91

List of Figures

Figure 1. Flowchart of this research (Kim and Cha, 2024).	34
Figure 2. NAFNET's block (Chen et al., 2022).	35
Figure 3. Image deblurring results: (a) Blurred images caused by motion blur, (b) Results after deblurring using NAFNet.	36
Figure 4. Pinhole camera model.	38
Figure 5. Camera intrinsic.....	38
Figure 6. Different coordinate system.	40
Figure 7. SfM process flowchart.....	41
Figure 8. Epipolar geometry.	43
Figure 9. Markers placed on the bridge model (Kim and Cha, 2024).	45
Figure 10. Marker detection in Metashape.	46
Figure 11. Photo alignment and camera poses estimation process in Metashape.	47
Figure 12. Transformation matrices information. (a): XML format extracted from Metashape, (b): JSON format converted from XML.	49
Figure 13. STRNet + TTA workflow diagram (Kim and Cha, 2024).	52
Figure 14. Code for TTA.	53
Figure 15. The STRNet architecture (Kang and Cha, 2022).	55
Figure 16. Overview of the NeRF model from the original NeRF paper (Mildenhall et al., 2021).	57
Figure 17. NeRF model structure.....	59
Figure 18. Ray casting process.	60
Figure 19. Positional encoding.	61
Figure 20. Stratified sampling.....	62
Figure 21. Training progress within the nerfstudio real-time visualizer module web URL. Clockwise from top left, the progression of the training (Kim and Cha, 2024).	65
Figure 22. Camera pose alignment visualization (Kim and Cha, 2024).	65
Figure 23. Crop adjustment in NeRF (Kim and Cha, 2024).	66
Figure 24. Sequential workflow of damage mapping. (a) our method: camera pose estimation before damage segmentation, (b) alternative approach: camera pose estimation with crack segmented images (Kim and Cha, 2024).	68
Figure 25. Aligned images based on camera poses (Kim and Cha, 2024).	69
Figure 26. Camera setting. (a) Nikon digital camera (b) Image quality setting, (c) Image size setting.	73
Figure 27. Mitigation of reflective floor surfaces in the indoor lab (a) shows the reflective floor surface (b) shows coverage of reflective surfaces by putting boxes on the floor.	74
Figure 28. Image quality variation (a) Daytime with sufficient sunlight enhancing color visibility. (b) Nighttime showing diminished light and color quality.	75
Figure 29. Lighting effects on top side (a) Unwanted lines and glare from direct sunlight. (b) Preferred diffuse lighting on a cloudy day or during nighttime.....	75
Figure 30. Markers utilized in the experiment.	76

Figure 31. Aligned Markers on the bridge (Kim and Cha, 2024).....	77
Figure 32. example for 80% image overlap of 6 images in a row.	78
Figure 33. Photography route methods (Kim and Cha, 2024).....	80
Figure 34. Crack segmentation results of each case. (a) The result of detection after resizing the image to 1280×720. (b) The result of detection after resizing the image to 2048×1024. (c) The result of combining the detections from (a) and (b). (d) The result of applying a combination of resizing to 1280×720 and 2048×1024, flipping horizontally for both sizes, flipping vertically for 1280×720, and rotating (90°, 180°, 270°) for (1280×720). (e) The Ground Truth mask.	85
Figure 35. The 3D reconstruction results of each case (Kim and Cha, 2024).	89
Figure 36. Comparison of an image and the 3D reconstruction result of the bridge.	92
Figure 37. Segmented crack mapped 3d bridge model.	92
Figure 38. Comparative visualization of an unsegmented crack and segmented crack on the 3d model.....	93

Copyright

This thesis reproduces and provides detailed explanations of the recently submitted paper to the *Automation In Construction*, an Elsevier journal, titled “Kim, G., and Cha, Y.J. (2024). Neural radiance fields (NeRF-based) 3D damage mapping with advanced deep learning damage segmentation.”

Terminology Abbreviations

Augmented Reality (AR)

Camera Coordinate System (CCS)

Computerized Tomography (CT)

Convolutional Neural Networks (CNNs)

Finite Element Analysis (FEA)

Fully Convolutional Networks (FCNs)

Internet of Things (IoT)

Mean Intersection over Union (mIoU)

Mean Squared Error (MSE)

Multi-Layer Perceptrons (MLPs)

Neural Radiance Fields (NeRF)

Nonlinear Activation Free Network (NAFNet)

Peak Signal-to-Noise Ratio (PSNR)

Pixel Coordinate System (PCS)

Random Sample Consensus (RANSAC)

Scale-Invariant Feature Transform (SIFT)

Semantic Transformer (STR)

Semantic Transformer Representation Network (STRNet)

Singular Value Decomposition (SVD)

Structural Similarity Index (SSIM)

Structure from Motion (SfM)

Support Vector Machines (SVM)

Test Time Augmentation (TTA)

Virtual Reality (VR)

World Coordinate System (WCS)

1 Introduction

SHM plays a vital role in assessing the condition and predicting the lifespan of infrastructure (Cha et al., 2024). Essential for ensuring the safety and extending the service life of structures, SHM also facilitates cost-effective management practices. Particularly in Canada, the aging bridge infrastructure accentuates the importance of SHM, showcasing its necessity in maintaining structural integrity. This chapter details the methodologies utilized throughout this thesis.

1.1 Background

Since 1973, the average age of bridges in Canada has steadily increased, a trend driven by insufficient investment (Hammad et al., 2007). By 2003, many federal and provincial bridges exceeded half of their projected 46-year lifespan. Federal bridges, constituting just 3% of the total, had an average age of 26.4 years. Provincial bridges, making up 57% of the total, saw their average age increase from 15.4 years in 1963 to 24.6 years in 2003, primarily due to minimal provincial funding. Meanwhile, municipal bridges, which accounted for 39% of the total, were younger on average at 19.0 years (Gaudreault and Lemire, 2006). This prevalent aging pattern, particularly among provincial bridges, underscores the necessity of regular inspections to ensure safety and extend their service life. Consistent application of SHM not only secures structural safety but also reduces maintenance costs, establishing it as a critical practice within the realms of civil engineering and infrastructure management (Giurgiutiu, 2015).

Traditional SHM systems rely on a variety of sensor technologies, such as piezoelectric (Giurgiutiu, 2007; Qing et al., 2008), fiber optic (Tennyson et al., 2001), and vibration sensors (Deraemaeker et al., 2010). However, these methods encounter significant challenges when implemented in large-scale civil infrastructure. Numerical techniques frequently suffer from

uncertainties and the influence of non-uniform environmental factors (Cha and Buyukozturk, 2015). A primary issue with these sensor-based approaches is their inability to clearly differentiate between actual structural damages, sensor malfunctions, and false signals from noise. This complication often demands on-site validation of the sensing systems and the structure itself, requiring the presence of specialized technicians, thus consuming considerable time and resources.

Cracks serve as significant indicators of structural deficiencies and commonly result from various factors such as cyclical loading, drying shrinkage, thermal expansion and contraction, and overall structural wear (Abdel-Jaber and Glisic, 2015; Sanchez-Silva et al., 2011). Timely addressing of these damages is crucial for maintaining long-term structural durability and preventing further deterioration. Effective monitoring and timely repair of cracks are thus essential to preserve structural integrity and extend the lifespan of infrastructure assets (Mahaidi and Kalfat, 2018).

The advancement of vision-based equipment and image-processing technologies has significantly propelled the field of Structural Health Monitoring (SHM) forward. Computer vision has become a pivotal tool for damage detection, providing a more dependable alternative to traditional human expert evaluations, which are typically labor-intensive and prone to inconsistencies (Cha et al., 2017; Feng and Feng, 2018).

Research on image-based crack detection, classification, and segmentation has been thorough (Zakeri et al., 2017). Traditional methods such as thresholding, edge-detection, and data-driven approaches have increasingly been viewed as inadequate due to their limitations in accuracy and the challenges they face in segmenting cracks against complex backgrounds (Kheradmandi and Mehranfar, 2022; Deng et al., 2022). However, the advent of deep learning, especially Convolutional Neural Networks (CNNs), has significantly transformed the field of damage

detection. CNNs are proficient in extracting complex features from images and adaptively refining their internal parameters for improved automated recognition (Cha et al., 2017; Cha et al., 2018). A significant innovation within this area is the U-net architecture, originally developed for medical image analysis (Ronneberger et al., 2015), and successfully adapted for both medical and non-medical image segmentation tasks, including crack detection (Jenkins et al., 2018; Liu et al., 2019).

Recent innovations have incorporated the Deeplabv3+ and Xception architectures for enhanced crack segmentation in bridges (Fu et al., 2021). However, the application of U-net in complex scenes containing diverse visual elements continues to pose challenges. To overcome these limitations, Kang and Cha (2022) introduced STRNet, an attention-based encoder-decoder model specifically designed for crack segmentation, which has shown improved efficacy in complex settings. These advancements in computer vision and deep learning underscore a significant evolution in SHM, marking a departure from traditional methods towards more sophisticated, automated techniques that enhance both the precision and efficiency of detecting structural damage.

A significant obstacle in deploying CNNs for damage detection in actual structures is the need for extensive training datasets (Dryden et al., 2019). Collecting a diverse array of damage scenarios to adequately train these models is challenging, and often unfeasible for in-service structures. This scarcity of comprehensive training data can limit the practical use of CNNs in real-world structural health monitoring applications (Ye et al., 2019).

To overcome this challenge, various strategies such as transfer learning and semi-supervised learning have been investigated. Transfer learning, outlined by Teng et al. (2023), entails adapting an existing pre-trained model to tackle a related but new problem. Semi-supervised learning,

detailed by Xiang et al. (2023), utilizes a mix of labeled and unlabeled data to enhance the training process.

While detecting damage in a single image for SHM can be informative, it presents challenges, particularly with large structures where a single image captures only a small section of the entire structure. This makes it difficult to fully understand the extent of the damage (Zhao et al., 2021). Therefore, accurately translating detected damages from 2D images to 3D world coordinates is essential, especially for expansive structures like bridges. This translation provides a more complete view of the damage's location and severity.

Moreover, incorporating 3D reconstruction into the digital twin framework marks a significant advancement in SHM. A digital twin, an advanced technological construct, operates on data and models to perform monitoring, simulation, prediction, and optimization functions (Tao et al., 2022). This method is gaining traction in research for its utility in creating digital replicas of bridges, thus revolutionizing intelligent SHM (Ye et al., 2019). The development of digital twins not only improves the precision of damage assessments but also facilitates smarter, predictive maintenance strategies (Chen et al., 2023).

3D reconstruction is crucial in developing digital twins for SHM, necessitating that 3D models feature rich surface textures and high spatial resolutions (Kong and Hucks, 2023). This process involves generating a three-dimensional representation of an object, utilizing data gathered via photogrammetric or laser scanning techniques. In the realm of computer vision, various techniques employ multiple images rather than just relying on stereoscopic imagery (Hartley and Zisserman, 2003). Structure from Motion (SfM) is one such technique that has proven effective in creating intricate 3D models of both objects and landscapes (Schonberger & Frahm, 2016; Hartley &

Zisserman 2003). For example, Xue et al. (2022) applied a deep learning enhanced SfM method to evaluate tunnel defects, showcasing the technique's suitability for extensive monitoring tasks.

Photogrammetry has been utilized to overlay detected concrete cracks onto reconstructed 3D models, with the accuracy of these overlays enhanced by CNN-based methods such as UNet (Chaiyasarn et al., 2022). However, 3D reconstruction via photogrammetry presents its challenges; it is often time-consuming and demands significant computational resources (Remondino et al., 2012). Moreover, accurately reconstructing planar surfaces using solely photogrammetry can be problematic, which is why laser scanning is sometimes employed to mitigate these issues (Schenk et al., 2001). This underscores the need for more efficient and effective 3D reconstruction methods within SHM.

1.2 Problem statement

Addressing the challenge of preparing extensive training samples for deep learning models, both semi-supervised and transfer learning methods require re-training. This process is often time-consuming and resource-intensive. Moreover, these approaches typically demand larger datasets.

In the domain of SHM, where rapid monitoring and assessment are critical (Hoshyarmanesh et al., 2017), Test Time Augmentation (TTA) proves invaluable. TTA enhances model performance by applying various augmentations to test images and aggregating the predictions (Kimura, 2021). This method does not necessitate model retraining but instead boosts prediction accuracy at testing time by utilizing diverse variations of the input data. According to Wang et al. (2019), TTA functions like an ensemble technique by amalgamating results from multiple augmentations to deliver a robust and precise outcome, making it exceptionally suitable for SHM applications that demand swift and dependable damage evaluation.

Reconstructing planar surfaces and reflective materials through traditional photogrammetry often presents significant challenges. Neural Radiance Fields (NeRF), introduced by Mildenhall et al. (2021), represent a revolutionary shift, offering an effective alternative to conventional photogrammetry methods. Unlike traditional techniques that rely on manual texture mapping and geometric primitives like triangles and voxels, NeRF uses a neural network to model a scene's visual characteristics directly. This method excels in scanning objects with sparse feature points, providing a more adaptable approach to 3D reconstruction. NeRF uniquely captures complex lighting effects such as reflections and transparencies, creating detailed representations of points from diverse viewing angles. Consequently, NeRF delivers more precise and detailed scene renderings (Gao et al., 2022), making it a promising technique for enhancing Structural Health Monitoring (SHM) capabilities.

1.3 Research objective

This research explores the application of advanced deep learning methods for damage segmentation and 3D reconstruction in the mapping of civil infrastructure. The objectives are detailed as follows:

1. Enhancement of STRNet using TTA: This aims to improve the accuracy of crack detection by refining the STRNet model with TTA, which enhances detection capabilities under varying conditions.
2. Adoption of the NeRF-based Nerfacto model: This addresses the limitations of traditional photogrammetry by leveraging the Nerfacto model, which enhances depth and detail in 3D reconstructions.
3. Development of a damage mapping method: This focuses on innovating a method to accurately map and quantify structural damages.

4. Validation of the pipeline on a bridge model for SHM: The goal is to confirm that the proposed methods are effective for practical applications in SHM.

By successfully achieving these objectives, the research enhances the capabilities of visualization of damage in 3d model for SHM systems. This progress is crucial for the proactive maintenance and management of infrastructure, ultimately contributing to the safety, longevity, and economic efficiency of civil structures.

1.4 Thesis organization

This thesis is organized into several chapters, each designed to elaborate on distinct aspects of the research. The structure is as follows:

1. Introduction: This opening chapter sets the stage for the research by outlining the problem statement, research objectives, and the significance of the study within the field of SHM and deep learning applications for damage detection and traditional 3d reconstruction method applied in SHM.
2. Literature review: This chapter reviews existing literature related to computer vision and deep learning methods for damage detection in SHM. Also, 3D reconstruction and NeRF technology. It highlights the gaps that this research aims to address.
3. Methodology: This chapter details the methodologies applied in the research, starting from data collection for NeRF, the application of STRNet with TTA, the internal workings of NeRF, and the techniques for 3D damage mapping.
4. Data collection: This chapter describes the methods used to collect high-quality image data for 3D reconstruction, ensuring the accuracy and effectiveness of the applied techniques.

5. Case studies: This chapter presents practical applications of the methodologies through various case studies. It includes the improvements in accuracy of STRNet with TTA, the quality of 3D reconstruction under different setups, and the visualization of damage on 3D models.
6. Conclusion and future work: The final chapter summarizes the findings of the research, discusses the implications of these results, and suggests directions for future research to further advance the field.

2 Literature review

This chapter presents a comprehensive review of the existing literature relevant to SHM, encompassing journal articles, and publications. Our objective is to provide a detailed overview of the current state-of-the-art in SHM, focusing specifically on the integration of computer vision and deep learning technologies in detecting structural damage. Additionally, we explore advancements in 3D reconstruction techniques, the application of NeRF, and the development and implementation of digital twins in the context of SHM.

2.1 SHM

SHM is designed to continuously assess a structure's condition throughout its lifecycle, ensuring it remains within established safety and performance parameters (Farrar and Worden, 2007). This process involves diagnosing the state of the materials, components, and overall assembly of structures subjected to normal aging, environmental impacts, and accidental events. SHM has attracted attention across various engineering disciplines due to its capability to address complex issues related to structural deterioration, which can arise from substandard materials, poor construction practices, inadequate inspections, or extreme conditions. By leveraging historical data, SHM not only assesses the current health of structures like buildings, bridges, tunnels, dams, and various transportation modes but also predicts damage progression and remaining useful life, enhancing long-term safety, efficiency, and sustainability of infrastructure.

2.2 Computer vision in SHM

While many SHM approaches traditionally focus on analyzing structural acceleration responses, displacement responses are often more indicative of overall structural stiffness and can provide more precise assessments of structural conditions. Recent advancements in camera technology and computer vision have introduced vision-based sensing as a viable alternative to traditional sensors for measuring structural dynamic displacements and conducting health monitoring. Cameras capture video images of structural features, such as rivets and edges, which are then processed using measurement software to derive displacement time histories at multiple points simultaneously (Feng and Feng, 2021). In the specific approach of Computer vision-SHM at the local level, images serve as the primary data source for analysis and assessment, with the expectation that these processes will soon be fully automated. Assessments are ultimately based on condition maps derived from the images (Cha et al., 2024).

In the realm of computer vision, damage detection is primarily categorized into two approaches: one is patch-based detection, the other one is pixel-based detection. The patched-based damage detection approach focuses on identifying the presence of cracks within specific segments or patches of an image. Abdel-Qader et al. (2006) explored various Principal Components Analysis applied methods for this purpose to classify the crack status within image patches. Additionally, some researchers have employed SVM, which is supervised learning techniques, for classifying cracks more accurately (Lin et al., 2002).

Unlike patch-based detection, pixel-based methods segment crack morphology at the individual pixelwise level. Edge detection is commonly utilized here, involving the use of various filters to delineate edge shapes within images. Also, several edge detection algorithms were explored by Abdel-Qader et al. (2006) and resulted in fast Haar transform generally provided more

trustful results among fast Haar transform, fast Fourier transform, Sobel, and Canny detectors. Further developments in this area include the application of the Sobel detector by Yu et al. (2007) to identify edges, and the integration of a wavelet-based algorithm with the Canny edge detector by Rong et al. (2014) to enhance crack segmentation.

Beyond these specific methods, there are additional general image processing techniques. Filtering, morphology, and binarization with thresholding are commonly employed. For example, Sinha and Fieguth (2006) utilized statistical filters to define and link global cracks, while Yamaguchi and Hashimoto (2010) employed percolation-based methods for crack extraction in large concrete structures. Fujita and Hamamoto (2011) used a combination of median filters and image subtraction to differentiate cracks from background noise.

However, these computer vision techniques face significant challenges under poor imaging conditions such as blur (Lee and Lee et al., 2013), uneven lighting (Brownjohn et al., 2017), or temperature (Zhou et al., 2017). Furthermore, Kang and Cha (2022) noted that these methods often struggle to detect multiple types of damage simultaneously, while Ribeiro et al. (2014) demonstrated the quantification of induced errors based on mathematical models in vision-based measurement tests.

2.3 Deep learning in SHM

Recent advancements have seen a significant incorporation of learning-based methods in computer vision studies to enhance the performance of applications. Deep learning technology is the representative method. Deep learning methods, particularly CNNs, are pivotal in detecting

multiple types of local damages simultaneously through end-to-end frameworks. This capability facilitates more effective and automated condition assessments (Xu et al., 2019).

CNNs are utilized to extract features from raw images through convolutional operations. Cha et al. (2017) introduced CNN based model to identify crack within small image patches (256×256 pixels) and employed a sliding window technique on tested images to predict the presence of cracks. Dorafshan et al. (2018) adapted AlexNet through transfer learning to develop a specialized CNN classifier for crack classification. Karaaslan et al. (2019) modified the VGG-16 network weights within a Single Shot MultiBox Detector framework for detecting crack regions delineated by bounding boxes and utilized the SegNet model for crack segmentation within these boxes. Similarly, Ali et al. (2021) utilized Faster R-CNN for detecting various damages such as cracks, corrosion, and loosened bolts on steel bridges in real-time.

For pixel-level crack detection, deep learning methods have shown promise. Dung and Anh (2019) implemented FCNs delineate cracks and provide detailed maps showing the shape and distribution of these defects. This could achieve semantic segmentation for the crack status. Choi and Cha (2019) developed a novel deep learning model, SDDNet, for effective crack segmentation against complex backgrounds. Furthermore, Kang and Cha (2022) applied an attention-based encoder and decoder model that achieved a state-of-the-art mIoU of 92.6%.

These developments underline the growing reliance on deep learning to address complex challenges in SHM, providing accurate, real-time assessments of structural health and aiding in the maintenance and safety of critical infrastructure.

2.4 3D reconstruction techniques

3D reconstruction from multiple images involves creating three-dimensional models by reversing the process typically used to capture 2D images from 3D scenes. This technology is increasingly used to generate detailed representations of objects in various formats, including point clouds, mesh models, and geometric models, with point clouds often serving as the foundational structure.

A 3D model comprises both geometric (position, orientation) and photometric (color, texture) data. This comprehensive data allows for the rendering of synthetic images by simulating the path of light rays from their sources to the camera, reflecting off the object's surface. 3D scanning, which includes the subproblems of 3D reconstruction (geometry estimation) and appearance estimation (photometric properties), effectively reverses this light trajectory to characterize the scene's surface (Durou et al., 2020).

The applications of 3D reconstruction are extensive and diverse, spanning across various fields such as robotic mapping and navigation, city planning, gaming, virtual tourism, medical field, archaeology, augmented reality, motion capture, gesture recognition, and hand tracking (Aharchi and Kbir, 2020).

In the context of SHM, 3D reconstruction techniques have also been pivotal. They have been utilized for the preservation of historical buildings (Brunetaud et al., 2012), dam inspections (Zhao et al., 2021), and the detailed visualization of reinforced concrete structures (Choi et al., 2016), showcasing their versatility in maintaining and monitoring structural integrity.

A variety of equipment is employed for 3D reconstruction, each offering different capabilities and suited to specific applications. Commonly used devices include monocular

cameras, binocular cameras, video cameras, and laser scanners (LiDAR). Less frequently used technologies in civil engineering include CT, ultrasonic tomography, Kinect, and total stations (Ma and Liu, 2018).

Equipment specific Methods:

- Monocular cameras: capture single images at a time.
- Binocular cameras: capture stereo images, providing distance information.
- Video cameras: capture a series of monocular images over time.
- Laser scanners: directly generate point clouds containing 3D spatial information.

The 3D reconstruction process varies based on the equipment used, typically involving several stages:

1. Feature extraction: identifying notable points or edges within images (Liu et al., 2016).
2. Feature matching: associating features across multiple images (Khaloo and Lattanzi, 2017).
3. Camera motion estimation: determining the movement of the camera between shots (Jog et al., 2011) — this step can be skipped when using stereo cameras.
4. Sparse 3D reconstruction: creating a preliminary 3D model from matched features (Liu et al., 2016).
5. Model parameters correction: refining the model for accuracy (Agarwal et al., 2010).
6. Absolute scale recovery: establishing the real-world dimensions of the model (Zollmann et al., 2014).
7. Dense 3D reconstruction: developing a detailed and complete 3D model (Yang et al., 2013).

After that, post-reconstruction, point cloud processing is employed to refine the raw data into a usable format, followed by mesh reconstruction to create a mesh model from the point cloud (Torok et al., 2012).

While laser scanners provide a quick and detailed collection of 3D data, they can be prohibitively expensive for large objects. Consequently, monocular cameras remain popular due to their cost-effectiveness. Combined with advances in camera resolution and computational power, vision-based technologies are increasingly viewed as an efficient approach for SHM (Adhikari et al., 2014).

2.5 NeRF

NeRF emerged as a ground-breaking approach in the field of view synthesis, introduced by Mildenhall et al. at ECCV 2020. This innovative method utilizes volume rendering combined with implicit neural scene representations through MLPs to capture the geometry and lighting of 3D scenes with unprecedented detail.

At its core, a NeRF model conceptualizes three-dimensional scenes as a radiance field, which is effectively approximated by a neural network. The radiance field encapsulates both color and volume density at every point in the scene, across all viewing directions. This relationship is formalized as:

$$F_{\Theta}: (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$$

where $\mathbf{x} = (x, y, z)$ denotes the coordinates within the scene, $\mathbf{d} = (\theta, \phi)$ represents the azimuthal and polar viewing angles, $\mathbf{c} = (r, g, b)$ indicates the RGB color values, σ denotes the volume density. This five-dimensional function is represented using one or more Multi-Layer Perception (MLP), denoted as F_{Θ} .

NeRF has not only set new benchmarks in visual quality but has also inspired a plethora of subsequent innovations. Recently, applications of NeRF models have expanded into diverse areas such as photo editing, 3D surface extraction, human avatar modeling, and extensive urban-scale 3D representation and view synthesis (Gao et al., 2022).

Since its introduction, numerous specialized adaptations of NeRF have emerged, each designed to address specific challenges in 3D rendering and computer vision. These variations include: (Müller et al., 2022):

1. NSVF (Neural Sparse Voxel Fields) (Liu et al., 2020):
 - Optimizes rendering by utilizing sparse voxel fields, significantly speeding up the process by efficiently representing empty spaces and focusing computational resources on regions with dense geometry.
2. Instant-NGP (Instant Neural Graphics Primitives) (Müller et al., 2022):
 - Greatly accelerates training and inference processes using a hash table-based approach for storing and accessing neural network parameters, facilitating rapid updates and real-time rendering capabilities.
3. FastNeRF (Garbin et al., 2021):

- Improves inference speed by adopting a factorized representation of radiance fields, which precomputes and caches scene representations for quicker rendering, albeit at the cost of higher memory usage.

4. DietNeRF (Jain et al., 2021):

- Addresses the challenge of working with limited input images by incorporating semantic consistency losses, which enhance the model's ability to generalize from sparse and limited data sets.

5. NeRF++ (Zhang et al., 2020):

- An advancement over the original NeRF that improves handling of unbounded scenes. It employs two separate models to accurately represent both the bounded foreground and the infinite background, enhancing realism in expansive environments.

6. Ref-NeRF (Verbin et al., 2022):

- Specializes in managing reflective surfaces by more accurately modeling specular reflections and highlights, beneficial in scenarios with reflective objects where traditional NeRF models may falter.

Compared to traditional methods like photogrammetry, NeRF demonstrates superior performance in handling objects with non-collaborative, texture-less, reflective, and refractive surfaces due to its sophisticated ability to interpret complex light interactions and surface properties (Remondino et al., 2023). NeRF is especially useful in extensive mapping of large areas with limited images, ideal for emergency situations where rapid and extensive data collection is needed but challenging. Moreover, NeRF is capable of accurately describing material characteristics and rendering homogeneous textures due to its advanced handling of lighting and surface properties, which traditional photogrammetry might not capture as effectively. Also, NeRF

is effective at reconstructing planar surfaces, whereas photogrammetry is not well suited for this task. Despite achieving higher fidelity and visual accuracy, NeRF is more prone to noise effects compared to photogrammetry, indicating a trade-off between visual quality and noise levels (Croce et al., 2023).

2.6 Digital twin in SHM

Digital twin technology replicates real-world objects or systems into highly accurate digital models. This virtual representation integrates attributes, structures, states, performances, and functions of physical entities, enabling simulations, analyses, and visualizations of real-world data to predict future scenarios (Tao et al., 2022).

Advancements in robotics and automation are being propelled by innovations in sensors, unmanned aerial vehicles (UAVs), information networks, and machine learning, promising enhancements across various sectors including building, healthcare, and transportation. Modern research is focusing on cutting-edge developments in cloud computing, big data, open-source software, and industry initiatives like the IoT, Smart Cities, and Industry 4.0. These technologies facilitate access to extensive datasets and computing resources, enabling the integration and optimization of data across systems, which significantly improves performance over time (Oztemel et al., 2020).

Digital twin technology offers numerous applications and benefits within SHM, particularly by mapping the comprehensive attributes, structures, states, performances, and functions of physical systems into a virtual environment (Boyes and Watson, 2022). This technology is increasingly adopted for long-term monitoring of large infrastructure, such as long-span bridges,

which are now frequently equipped with SHM systems to enhance predictive maintenance and operational planning (Zhang et al., 2024).

In bridge engineering, for example, Jiang et al. (2021) developed a digital twin framework that uses a non-deterministic method to predict the multiscale fatigue life of steel bridges, thereby deepening the understanding of fatigue processes throughout the bridge's lifecycle. Similarly, Cheng et al. (2019) created a digital twin for a self-anchored suspension bridge, using SHM data to derive fragility curves that proved more accurate than those generated from traditional finite element models.

Digital twins also prove invaluable in the preservation of historic structures. Angjeliu et al. (2020) employed digital twins to assess the structural system integrity of historic masonry buildings, aiding in system response analysis, maintenance, and strengthening measures. Furthermore, Xu et al. (2023) managed and analyzed monitoring sensors and data from a building's internal components to dynamically update its health state, illustrating the capacity of digital twins to integrate and synthesize complex datasets in real-time.

Lin et al. (2021) used a digital twin and a nonlinear model updating method to evaluate the collapse fragility of a cable-stayed bridge, highlighting the model's utility in critical structural assessments. For simulations involving complex structural behaviors, such as vortex-induced vibrations, Zhang et al. (2024) applied SHM-based digital twin technology to provide insightful predictive analyses.

Moreover, digital twins facilitate the integration of FEA within SHM practices. Funari et al. (2021) showcased a parametric modeling framework within a visual programming environment,

aimed at creating FEM of historic masonry buildings, thereby extending digital twin applications to detailed structural analyses and conservation efforts.

Seeing damage in 3D models represents a significant advantage over traditional 2D imaging techniques. While 2D imaging is effective for identifying surface damage, it does not provide insights into the depth and severity of the damage. In contrast, 3D data derived from advanced reconstruction methods offers detailed information about damage depth, enhancing the accuracy of assessments and updates to existing models. Digital twin technology, integrated within SHM, facilitates this process by allowing for clear visualization of damage quantification, localization, and patterns, making it easier to intuitively inspect infrastructure compared to traditional 2D images, which can obscure the precise location and extent of damage (Grieves and Vickers, 2017; Rashidi et al., 2020).

Several studies illustrate the practical applications of this technology in SHM. Yeum et al (2017) performed an inspection based on the structure of highway signs, using the SfM technique to localize and quantify regions of interest. Chen et al. (2019) used a UAV to capture bridge images, from which they conducted 3D reconstructions to extract and quantify damage areas. Zhao et al. (2022) developed a methodology that combines the YOLOv5s-HSC algorithm with 3D photogrammetric reconstruction to identify and localize damage on concrete dams. Majidi et al. (2023) integrated automated crack detection using DeepLabv3+ with Xception and Squeeze-and-Excitation into 3D reconstruction processes for civil infrastructures. This approach allows that predicted damage masks can be unified with original images for SfM, facilitating precise 3D crack mapping.

As discussed above, 3D models provide significant advantages in SHM. However, they are generally created through the post-processing of raw 3D data from laser scanners and

photogrammetry pipelines (Masciotta et al., 2023), a process that presents challenges such as time efficiency and accurate representation of planar surfaces (Croce et al., 2023). Additionally, the accuracy of mapped damages can vary depending on the methods used to capture damage geometry, which poses another significant concern.

3 Methodology

This thesis introduces a novel methodology for 3D damage mapping by combining pixel-wise damage segmentation through deep learning with 3D image reconstruction techniques. The process, depicted in Figure 1, begins with the collection of 2D RGB images of the structure of interest. These images are first processed using NAFNet for deblurring, preparing the images for accurate damage detection and 3D modeling. Then, SfM via the Agisoft Metashape software tool is used to extract camera pose estimations (Intrinsic and extrinsic camera parameter) for each image, represented in the form of a transform matrix $[\mathbf{R}|\mathbf{t}]$, which are crucial input data for the subsequent NeRF based 3D reconstruction. Concurrently, the images are input into our enhanced TTA-STRNet model for precise pixel-wise damage segmentation. This segmented damage information is then integrated into NeRF, facilitating the reconstruction of a 3D damage map. The final output, a damage-inclusive digital twin of the structure, can be visualized with a 3D visualization tool, offering a comprehensive tool for SHM. By synthesizing the capabilities of NAFNet, SfM, STRNet, TTA, and NeRF, this thesis presents a comprehensive approach to creating detailed and accurate 3D reconstructions of structural damage. This methodology not only facilitates the detection and visualization of damage in bridge structures but also contributes to the advancement of SHM by providing a digital twin that serves as an asset for maintenance and safety assessments.

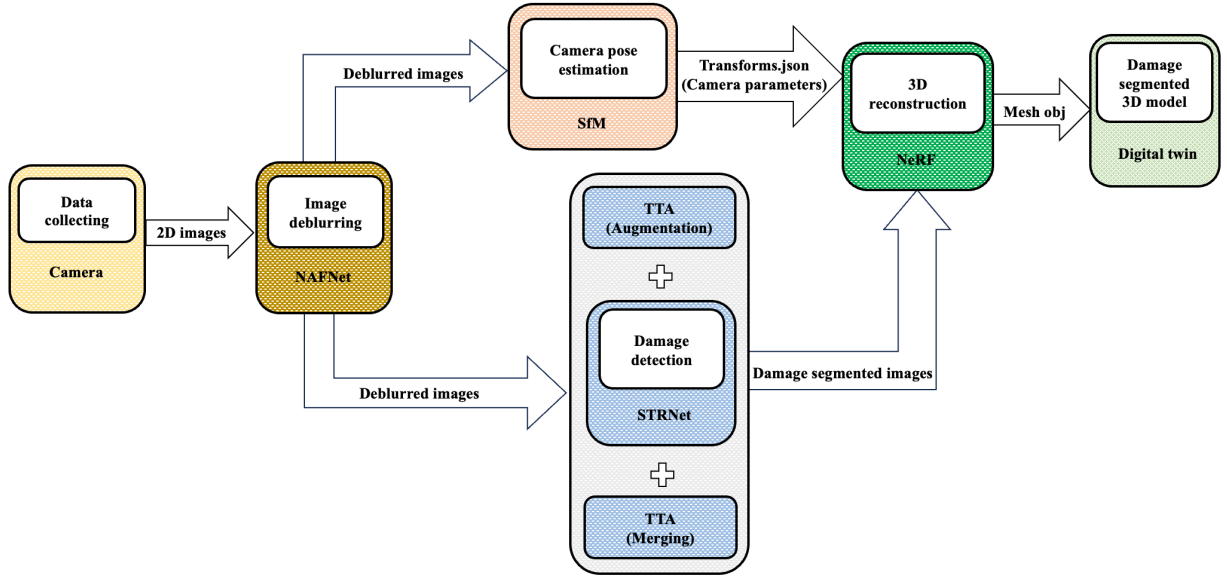


Figure 1. Flowchart of this research (Kim and Cha, 2024).

3.1 Image deblurring with NAFNet

In our pipeline, we integrated the NAFNet model (Chen et al., 2022), a deep learning-based solution for image deblurring, to enhance the quality of images used in 3D reconstruction. The challenge of image blurring, particularly prevalent in large-scale structures and when images are extracted from drone-captured videos, necessitated a robust solution. Manual deletion of blurred images is feasible for small datasets, but impractical for the extensive image collections required for detailed structures like buildings or bridges.

NAFNet, as described in 'Simple Baselines for Image Restoration', stands out for its simplicity and computational efficiency, surpassing state-of-the-art methods in image restoration. The model's architecture is characterized by a departure from conventional nonlinear activation functions like Sigmoid, ReLU, or GELU. Instead, it employs a Nonlinear Activation Free Network approach, which simplifies the processing pipeline while maintaining, or even enhancing, the

restoration quality. Its performance on challenging benchmarks, such as achieving 33.69 dB PSNR on the GoPro dataset for image deblurring with only 8.4% of the computational costs of previous state-of-the-art methods, demonstrates its efficacy.

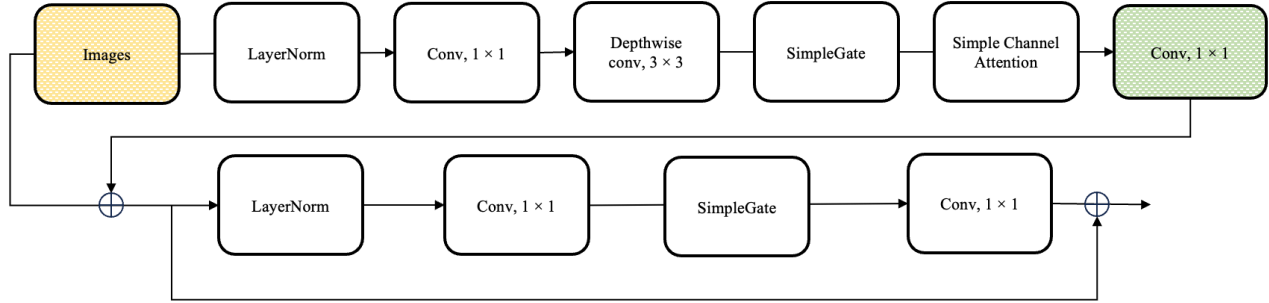


Figure 2. NAFNet's block (Chen et al., 2022).

The application of NAFNet in our project improved the quality of blurred images for NeRF input. By reducing blur, NAFNet enhanced the accuracy of the subsequent 3D reconstruction process. Furthermore, the model's efficiency in computational resource usage made it an ideal choice for our project, where large volumes of data required processing.

For the implementation of NAFNet, a virtual anaconda environment configured with Python 3.9.5, PyTorch 1.11.0, and CUDA 11.3 was utilized. By executing the command line with a pre-trained configuration file, the dataset of blurred images was processed, resulting in deblurred outputs ready for further processing. Figure 3 illustrates the deblurring effectiveness of NAFNet, showcasing enhanced sharpness and clarity of the images. This improvement is crucial for avoiding artifacts in the 3D reconstruction process. The impact of using deblurred images versus blurred images on the quality of 3D reconstruction is further explored in the "4. Experiment and Results" section, demonstrating the benefits of integrating NAFNet into the pipeline.



(a) Before NAFNet

(b) After NAFNet

Figure 3. Image deblurring results: (a) Blurred images caused by motion blur, (b) Results after deblurring using NAFNet.

The integration of NAFNet into our workflow not only saved time but also ensured that the quality of the 3D reconstructions was not compromised by the inherent blurring in the source images. This approach is expected to be particularly beneficial for 3D reconstruction of large-scale structures, where managing the quality of a huge number of images is crucial.

3.2 Obtaining camera parameters for NeRF input

To effectively employ NeRF in the methodology, accurate 3D coordinates and viewing directions associated with each image are indispensable. These critical pieces of information are derived from the camera's intrinsic $[K]$ and extrinsic $[R|t]$ parameters. These parameters are essential for translating 2D image data into a 3D spatial context, serving as a foundational step in the 3D reconstruction process. In the subsections of 3.2, the role of the camera parameters and the process of obtaining these camera parameters will be introduced, highlighting their pivotal role in accurately translating 2D image data into the 3D space.

3.2.1 Camera parameters information with pinhole camera model

The pinhole camera model serves as a foundational concept in the field of computer vision, simulating the way light travels through a small aperture (pinhole) to project an inverted image onto the camera's sensor. This model provides a simplified yet effective framework for understanding the geometric relationships between objects in the 3D world and their 2D representations captured by a camera.

As depicted in Figure 4, the pinhole camera model illustrates how an image is formed when light rays passing through the pinhole converge on the camera sensor, resulting in an upside-down and reversed image.

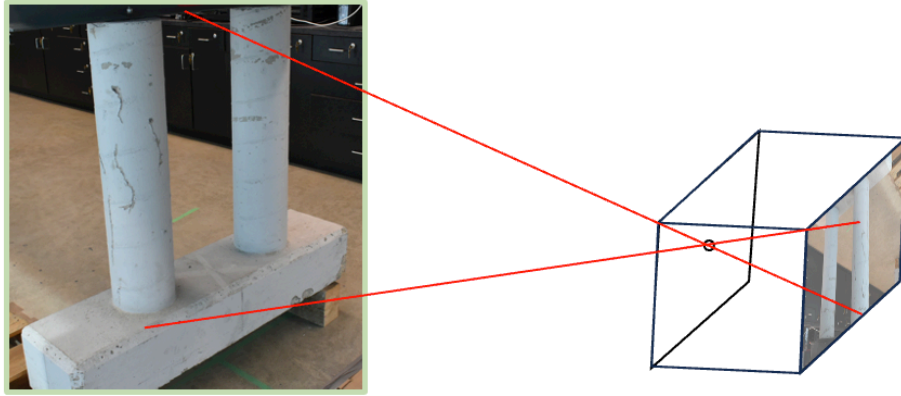


Figure 4. Pinhole camera model.

Figure 5 further explains the relationships between world 3D coordinates (X, Y, Z) , normalized image coordinates, and pixel coordinates.

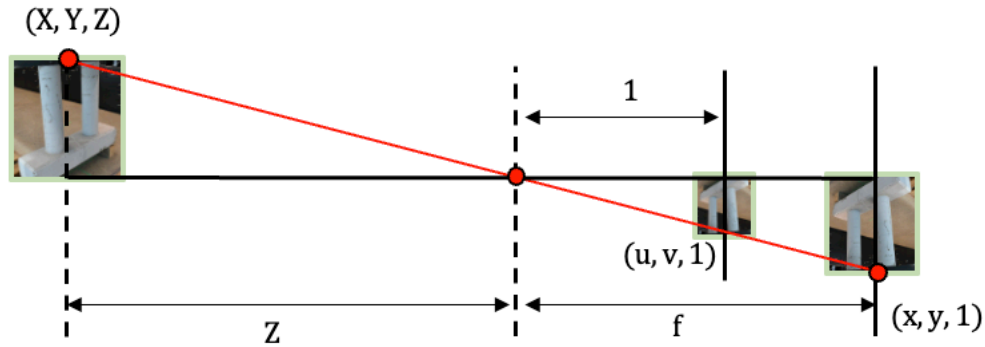


Figure 5. Camera intrinsic.

The focal point, represented by a dot along the solid line in the diagram, is the location where the camera's lens focuses light rays. The distance between this focal point and the object being photographed Z directly influences the size and position of the projected image. Additionally, the distance from the pinhole to the image plane f , known as the focal length, is a critical parameter that affects the field of view and the scale of the image projection. The intrinsic camera parameters can translate these geometric relationships, as the Equation (1) depicted.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = Z \mathbf{K}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

where intrinsic parameter matrix is represented by $\mathbf{K}$, as given by equation (2).

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The internal parameters refer to the camera's intrinsic qualities that influence image formation. These include the focal lengths (f_x, f_y), the principal point coordinates (c_x, c_y) that denote the image sensor's size, and the lens distortion coefficient (s), which is often negligible in modern cameras due to advanced sensor technology. Understanding these equations allowed me to discern how each pixel in a 2D image corresponds to a specific point projected from the WCS.

However, it's important to recognize that the CCS may differ with each shot due to variations in camera position and angle, affecting the 3D coordinates $[X, Y, Z]$ representation across images. Figure 6 shows how differences in camera angles and positions can cause the coordinates to be represented differently, as $[X', Y', Z']$, in another CCS.

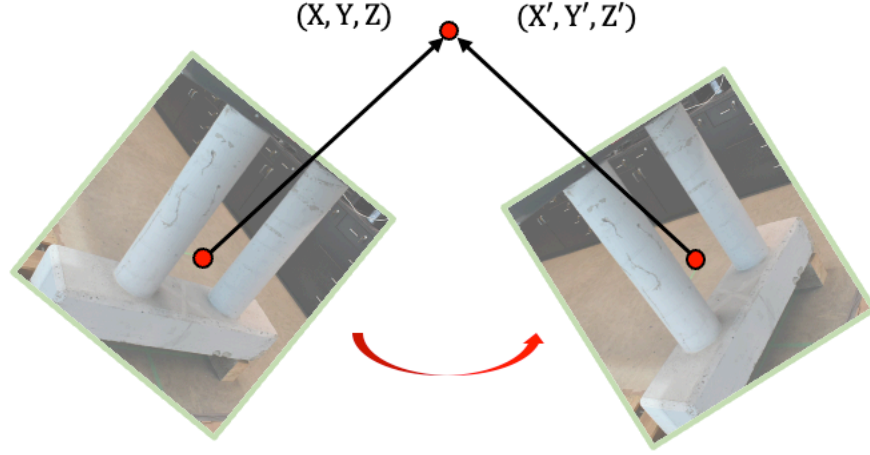


Figure 6. Different coordinate system.

To address these discrepancies and align the CCS with the WCS, Equation (3) can be applied, utilizing the extrinsic parameters, matrix $[R|t]$.

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = [R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3)$$

These extrinsic parameters are for defining how a scene's 3D structure is projected onto the camera's 2D image plane, encompassing rotation (R) and translation (t) that describe the camera's viewpoint. The extrinsic camera matrix, detailed in Equation (4)

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

The final step involves converting all 3D coordinates from the WCS to the PCS and vice versa, as outlined in Equation (5).

$$Z' \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (5)$$

This conversion is important for the effective deployment of NeRF in our 3D reconstruction process. To derive these essential parameters, we employ SfM, a technique in computer vision, which will be further elaborated upon in Section 3.2.2.

3.2.2 SfM

To extract camera parameters essential for accurate 3D reconstruction, we employ the SfM, a technique in the field of computer vision. This subsection visually outlines the SfM process through a flowchart, as shown in Figure 7 that highlights the progression from initial image capture to the creation of a sparse point cloud. This visualization emphasizes the steps particularly relevant to NeRF, such as the derivation of transformation matrices.

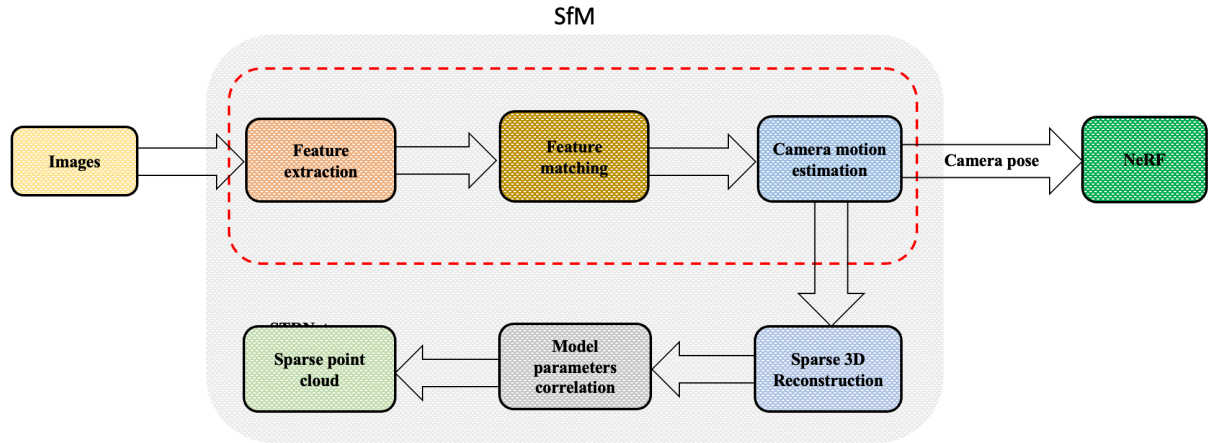


Figure 7. SfM process flowchart.

The steps enclosed within the red dotted box in the flowchart—feature extraction, feature matching, and camera motion estimation—are the procedures for deriving camera pose within SfM for NeRF.

To provide an overview of how the process unfolds, the flowchart begins with image acquisition, followed by feature extraction using the SIFT (Lowe, 2004) algorithm, which identifies key points that are invariant to scale and rotation. These features are matched across images to establish correspondences, forming the basis for estimating camera motion.

To ensure the reliability of these matches, the RANSAC algorithm is applied to remove outliers by randomly sampling data and selecting the maximum first-order value. Subsequently, eight matched feature points are extracted for each pair of images to compute the fundamental matrix, F , using SVD. The fundamental matrix is a matrix that describes the geometric relationship between the actual pixel coordinates of the two images, including the camera parameters. This can be expressed with the Equation (6).

$$p'_{img}{}^T F p_{img} = 0, \quad (6)$$

where p_{img} and p'_{img} represent the pixel coordinates in the respective images.

Then $p'_{img}{}^T F p_{img} = 0$ is then used. Epipolar geometry explores the geometric relationship between two images taken at different times from different viewpoints. As illustrated in Figure 8, a point P in 3D space projects onto p in image A and p' in image B. The epipoles e and e' are where the line connecting the camera origins intersects the image planes, and the epilines l and l' connect the projection points to the epipoles.

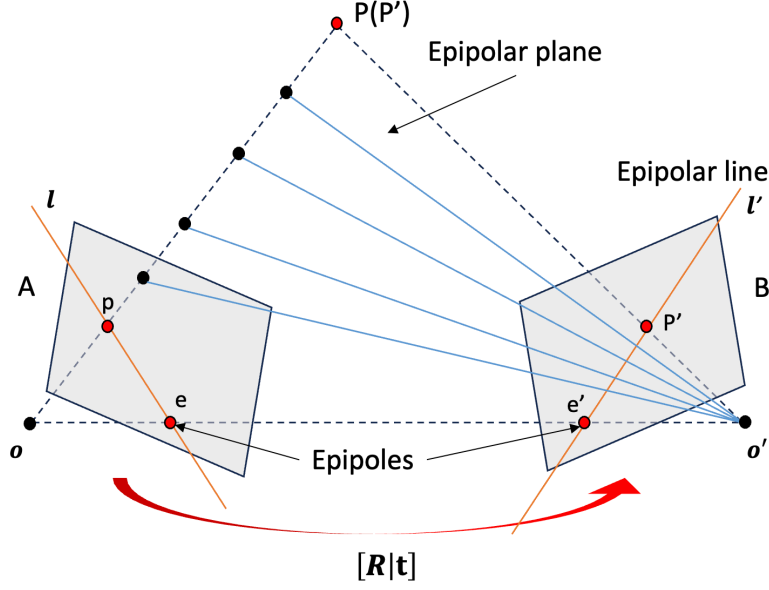


Figure 8. Epipolar geometry (Hata and Savarese, 2017).

With essential matrix E , we can move p to p' it can be expressed with the equation (7) due to the property that the orthogonal inner product is zero. The Essential Matrix is a matrix that describes the geometric relationship between matching pairs in the normalized image plane. Without depth information, the 3D coordinates of point P cannot be reconstructed solely from its image coordinates. However, because point P lies on a line connecting the origin of camera A to p , it projects onto image B along this line, forming the epiline l' . This projection is governed by the essential matrix E , which facilitates the transformation from p to p' (Normalized image coordinates). This relationship can be mathematically expressed as follows,

$$p'^T E p = 0, \quad (7)$$

where p is the normalized image coordinate, and E is the Essential Matrix, incorporating the rotation matrix R and the translation vector t .

With the intrinsic camera parameters known ($\mathbf{K}$), as defined in Equation (2), we can establish the relationship $\mathbf{p}_{img} = \mathbf{K}\mathbf{p}$ between pixel and normalized image coordinates. Substituting this into Equation (7), this leads to the derivation of Equation (8):

$$p'_{img}{}^T (\mathbf{K}'^T)^{-1} \mathbf{E} \mathbf{K}^{-1} p_{img} = p'_{img}{}^T \mathbf{F} p_{img} \quad (8)$$

This transformation illustrates how the essential matrix, and the fundamental matrix are interlinked through the intrinsic parameters, leading to Equation (9):

$$\mathbf{E} = \mathbf{K}^T \mathbf{F} \mathbf{K} \quad (9)$$

From this, the essential matrix can be extracted using the intrinsic parameters and the fundamental matrix. Subsequently, the extrinsic parameters, which consist of the rotation matrix and translation vector, are calculated from the essential matrix. This sequence of calculations is important for deriving the camera parameters necessary for accurate 3D reconstruction.

The necessity of capturing a comprehensive set of images to maximize feature matching opportunities is underscored by NeRF's reliance on precise camera parameters for realistic 3D rendering. To further enhance the precision of these parameters, Section 3.2.3 will introduce the use of the Metashape software tool.

3.2.3 Camera poses extraction with Metashape

Among the various SfM tools available, we selected Metashape software tool for its excellent robustness and precision in extracting camera parameters. Metashape provides its own proprietary markers, each identified by an ID. Once strategically placed within the scene, these

markers enable Metashape to precisely identify the location and orientation from which each photograph is taken.

Initially, markers created in the Metashape software are printed and strategically placed on the bridge as shown in Figure 9. Employing many markers is advantageous, particularly in environments like bridges where repetitive structural elements can make unique feature detection challenging. After positioning the markers, photographs of the bridge are captured and uploaded to Metashape.

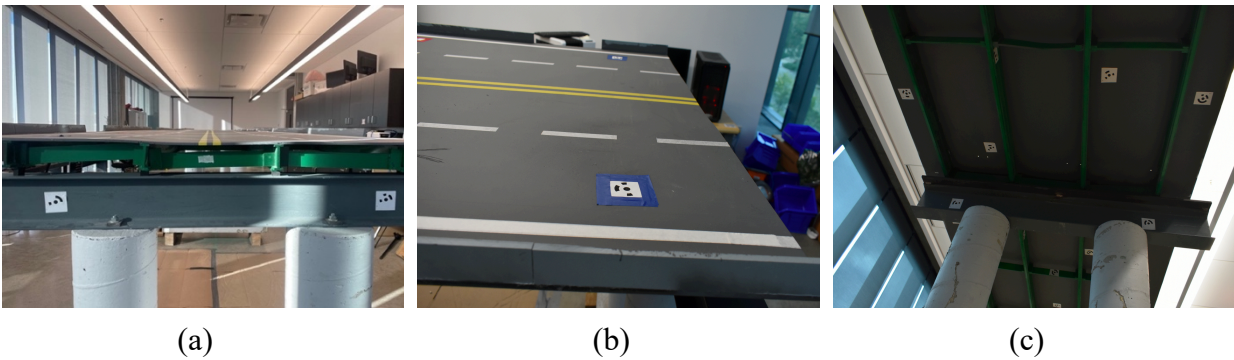
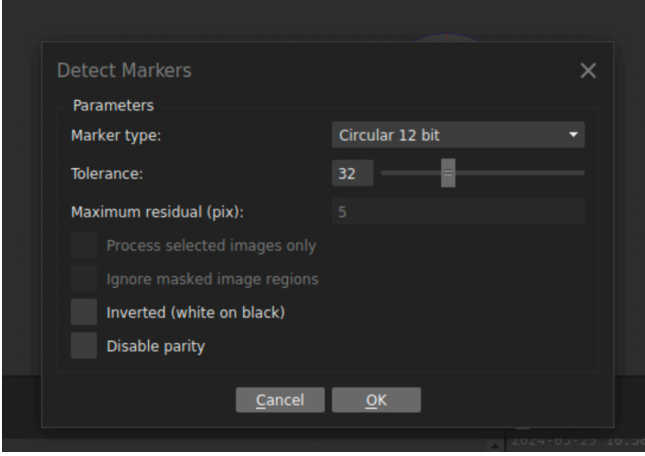
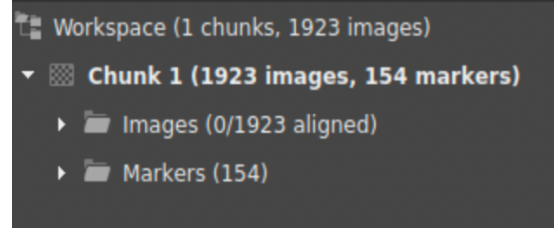


Figure 9. Markers placed on the bridge model (Kim and Cha, 2024).

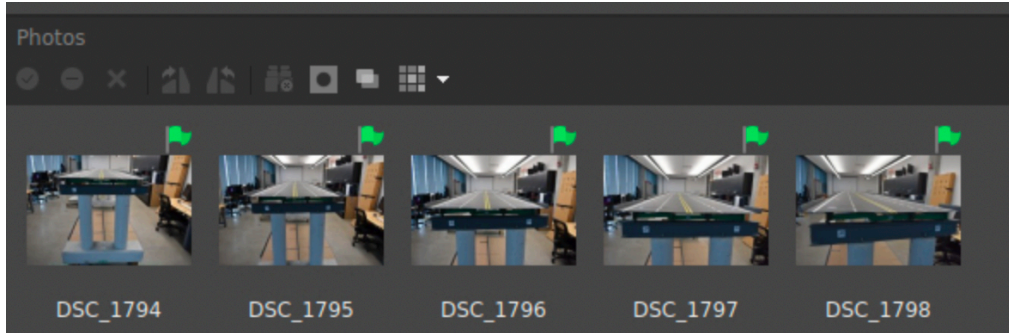
Within the software, we initiate the ‘detect markers’ process with an adjusted tolerance setting of 32, as demonstrated in Figure 10(a). This setting effectively balances sensitivity and specificity in marker detection. Post-processing, we verify that 154 markers have been successfully identified, as illustrated in Figure 10 (b). Each image's marker detection status is also displayed, providing a comprehensive overview of the process effectiveness, as shown in Figure 10 (c).



(a)



(b)

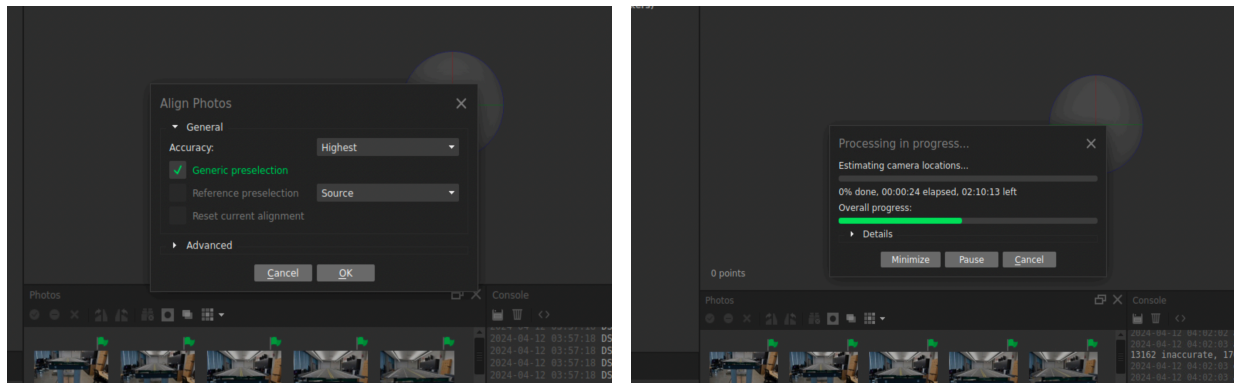


(c)

Figure 10. Marker detection in Metashape.

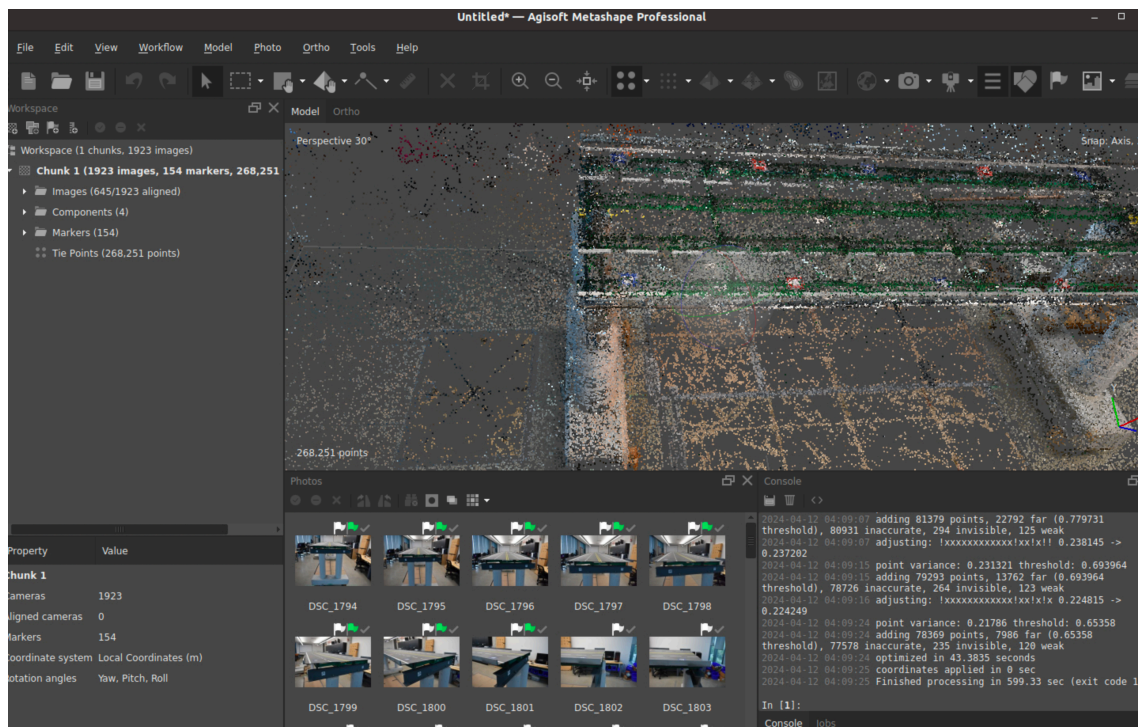
This marker-based tracking significantly enhances the precision in determining the extrinsic camera parameters, which are pivotal for constructing accurate transformation matrices. Subsequently, we align the photographs using images where markers were detected with the highest accuracy, as depicted in Figure 11 (a). This method significantly improves feature extraction, leading to enhanced accuracy, relying on the computational methods discussed in the previous subsection (3.2.2). The process also involves estimating the camera's location, with a real-time display of the estimated time remaining for this calculation, as shown in Figure 11 (b).

Following this, the aligned 3D point cloud is visualized, confirming that it is correctly positioned according to the calculated camera parameters, as shown in Figure 11 (c).



(a)

(b)



(c)

Figure 11. Photo alignment and camera poses estimation process in Metashape.

These accurately estimated camera poses then can be extracted. Metashape outputs this data in XML format, which, while detailed, requires conversion to JSON format to be compatible with NeRF systems. For this conversion, we employ the `agi2nerf.py` script, resulting in the generation of a `transform.json` file as illustrated in Figure 12 from (a) XML format to (b) JSON format.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0"?>
<document version="2.0.0">
  <chunk label="Chunk 1" enabled="true">
    <sensors next_id="1">
      <sensor id="0" label="NIKON D7200 (18mm)" type="frame">
        <resolution width="2992" height="2000"/>
        <property name="pixel_width" value="0.0080147904546248282"/>
        <property name="pixel_height" value="0.0080147904546248282"/>
        <property name="focal_length" value="18"/>
        <property name="layer_index" value="0"/>
        <bands>
          <band label="Red"/>
          <band label="Green"/>
          <band label="Blue"/>
        </bands>
        <data type="uint8"/>
        <calibration type="frame" class="adjusted">
          <resolution width="2992" height="2000"/>
          <f>2299.0383375918486</f>
          <cx>16.341481278138424</cx>
          <cy>-3.1257796894780139</cy>
          <k1>-0.02783657890457856</k1>
          <k2>-0.03125226037138424</k2>
          <k3>-0.02175779188851349</k3>
          <p1>0.00025102018126486956</p1>
          <p2>0.00044247160749823596</p2>
        </calibration>
        <covariance>
          <params>f cx cy k1 k2 k3 p1 p2</params>
          <coeffs>0.038996779827710118 0.0010762615028576852 -0.012966257767583693 -7.9375639668785841e-06 4.500910313415952e-05 -4.3832413057462816e-05
            1.0432290999317618e-07 -1.1773011313525839e-06 0.0010762615028576852 0.088581897498510143 -0.0026320746180893261 3.0720154483903594e-07 -7.4870320322074727e-07
            1.2815421427782095e-06 1.0141707101651689e-05 -1.8616898874378875e-07 -0.012966257767583693 -0.0026320746180893261 0.083043637767403602 -5.1433097501925073e-06
            1.2306229493840732e-05 -1.2655083939946411e-05 -3.5727663797803296e-07 7.5642701552741879e-06 -7.9375639668785841e-06 3.0720154483903594e-07
            -5.1433097501925073e-06 1.0207668301320945e-07 -3.673001570171035e-07 3.948941423775939e-07 9.0959644892208075e-11 -8.0766092435275109e-10 4.500910313415952e-05
            -7.4870320322074727e-07 1.2306229493840732e-05 -3.673001570171035e-07 1.4551803011146206e-06 -1.6414928268565609e-06 -3.224956696467315e-10 1.9251421482596038e-09
            -4.3832413057462816e-05 1.2815421427782095e-06 -1.2655083939946411e-05 3.948941423775939e-07 -1.6414928268565609e-06 1.920869502617566e-06
            5.1388750215636359e-10 -1.8418603392915816e-09 1.0432290999317618e-07 1.0141707101651689e-05 -3.5727663797803296e-07 9.0959644892208075e-11 -3.224956696467315e-10
            5.1388750215636359e-10 1.3560802346439942e-09 -2.2548837516938284e-11 -1.1773011313525839e-06 -1.8616898874378875e-07 7.5642701552741879e-06
            -8.0766092435275109e-10 1.9251421482596038e-09 -1.8418603392915816e-09 -2.2548837516938284e-11 9.4801457611208284e-10</coeffs>
        </covariance>
        <meta>
          <property name="Exif/Make" value="NIKON CORPORATION"/>
          <property name="Exif/Model" value="NIKON D7200"/>
          <property name="Exif/Software" value="Ver.1.01 "/>
        </meta>
      </sensor>
    </sensors>
    <components next_id="1" active_id="0">
      <component id="0" label="Component 1">
        <region>
          <center>-3.6098921712237679 -0.5720861130369479 -3.5626499484005381</center>
          <size>28.510226384798685 21.735423318545024 12.609481159845988</size>
          <R>-0.45946701856520478 -0.41761027491866809 0.78389522076177764 0.88804972799159709 -0.23194892319755037 0.39694757543225728 0.016054266250836665
            0.87852225658195549 0.47743157124900742</R>
          </region>
        </component>
      </components>
    </chunk>
  </document>
```

(a)


```

1 {
2   "camera_angle_x": 1.153705207889192,
3   "camera_angle_y": 0.8205295762307923,
4   "fl_x": 2299.1357279765075,
5   "fl_y": 2299.1357279765075,
6   "k1": -0.02523829354043384,
7   "k2": 0.02615603813173497,
8   "p1": 0.00041249552637485264,
9   "p2": 0.00018614380211391207,
10  "cx": 1513.5448411265872,
11  "cy": 994.5325980664471,
12  "w": 2992.0,
13  "h": 2000.0,
14  "aabb_scale": 16,
15  "frames": [
16    {
17      "file_path": "./Images/DSC_1794.jpg",
18      "sharpness": 0,
19      "transform_matrix": [
20        [
21          0.5821194241594156,
22          0.5286518916080128,
23          -0.6177897324459826,
24          -15.670284994267199
25        ],
26        [
27          -0.7781185425890393,
28          0.1417174844780904,
29          -0.6119213088889056,
30          -15.461944709784367
31        ],
32        [
33          -0.23594175064074543,
34          0.8369249261985637,
35          0.49385033989267246,
36          6.132018719263941
37        ],
38        [
39          0.0,
40          0.0,
41          0.0,
42          1.0
43        ]
44      ],
45    },

```

(b)

Figure 12. Transformation matrices information. (a): XML format extracted from Metashape, (b): JSON format converted from XML.

This conversion process is not just a technical necessity, but a critical step in seamlessly integrating SfM data into NeRF’s workflow to accurately project 3D rays onto captured scene for good 3D reconstruction result.

3.3 TTA-STRNet crack segmentation

In our research, we utilized the STRNet for precise and real-time segmentation of structural damages, such as cracks. STRNet, a cutting-edge deep learning model, excels in pixel-level crack detection, achieving an impressive processing speed of 49 frames per second on 1024×512 pixels images. This high performance is backed by its training on an extensive dataset of 1,203 images that underwent synthesis-based augmentation to simulate a wide range of damage scenarios, thus

enhancing the model's generalization capability across diverse structural conditions (Kang and Cha, 2022).

During the evaluation phase, STRNet was rigorously tested on a separate dataset consisting of 545 images with resolutions of 1280×720 and 1024×512 . It was evaluated across several key metrics: precision, recall, F1 score, and mean Intersection over Union (mIoU). The results were outstanding, with STRNet achieving 91.7% precision, 92.7% recall, 92.2% F1 score, and 92.6% mIoU, confirming its efficacy in accurately identifying and delineating cracks.

Despite these strengths, STRNet faced challenges when applied to images of varying sizes (2992×2000) and different materials encountered in actual structural settings. When tested on 20 high-resolution images, the network's performance dropped, with an mIoU of just 59.9%, indicating a need for model adjustments to handle real-world diversity effectively. This decline is attributed to the limited variation within the training datasets, which are not sufficiently representative of the complexities found in natural environments (Dryden et al., 2019). Additionally, there is a concern regarding bias towards specific types of augmented data, which could affect the model's performance in practical applications (Smith and Ricanek, 2020).

To overcome these obstacles, we integrated TTA with STRNet. TTA is an approach that applies different augmentations to a test image to create many different versions of each image, allowing the model to predict these variations. The results are then aggregated to enhance the accuracy and robustness of the predictions (Kimura, 2021). Incorporating TTA not only bolsters STRNet's ability to maintain high performance without the need for retraining but also helps reduce the model's susceptibility to bias by evaluating several augmented versions of the identical image within test time. This strategy effectively uses ensemble techniques to refine the model's

output, ensuring more reliable and precise damage assessments—critical for the success of SHM (Wang et al., 2019).

3.3.1 TTA

TTA is a technique applied during the inference phase of model evaluation, distinct from traditional data augmentation, which typically occurs during the training phase. Applying TTA includes a series of transformations to the images—for instance, resizing, flipping, and rotating—at the time of testing. The results from these transformations are then aggregated to make a final version of prediction. In our study, implementing TTA led to a significant improvement, increasing the mean mIoU by 6.6%, from 73.7% without TTA to 80.3% with TTA.

For efficient and effective damage detection, a high resolution captured images of 2992×2000 pixels are resized to 1280×720 and 2048×1024 . After the damage detection process, these images are scaled back to their original dimensions to maintain the integrity of the data. To bolster the model's robustness and account for various orientations and perspectives, we also incorporate horizontal and vertical flips, as well as rotations at 90° , 180° , and 270° . This comprehensive augmentation strategy creates 12 unique test scenarios, enhancing our ability to assess the performance of the model across different image sizes and material properties, as illustrated in Figure 13.

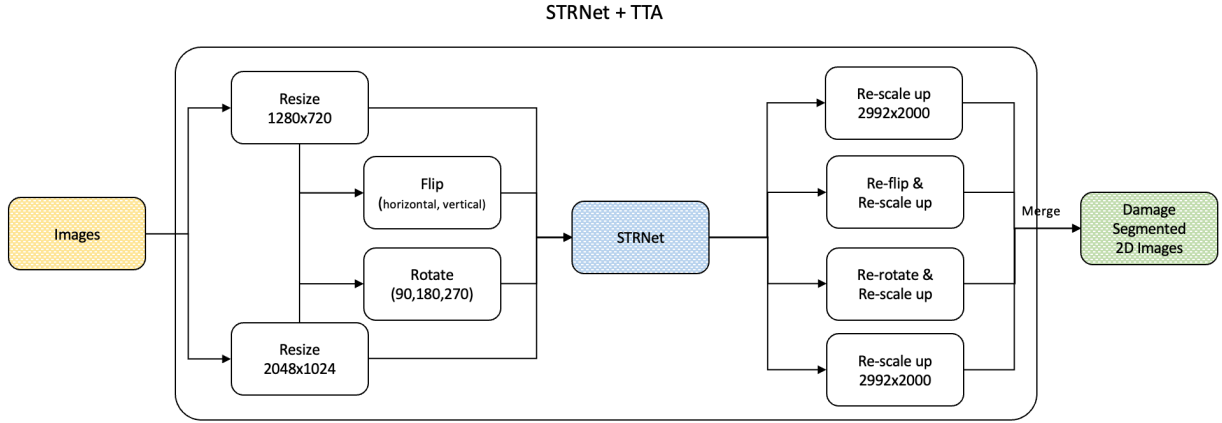


Figure 13. STRNet + TTA workflow diagram (Kim and Cha, 2024).

It is crucial to recognize that not all TTA techniques uniformly enhance accuracy. Certain transformations might even detract from the model's effectiveness, necessitating a selective approach to determine the most beneficial TTA methods for specific scenarios. In our experiments, the most effective combination included resizing images to 2048×1024 and applying horizontal flips for all resized images, vertical flips for those resized to 1280×720, and rotations at 90, 180, and 270 degrees for the same set of resized images. This configuration yielded the highest accuracy improvements in crack detection, with further elaboration available in the case study section.

To facilitate an efficient trial-and-error method for determining the best TTA configurations, we developed a flexible code structure. Figure 14 showcases the code for our TTA setup, defined in the `img_setup` function. This function allows for toggling each transformation on or off—such as `flip_leftright`, `flip_updown`, `rotate90`, `rotate180`, and `rotate270`—enabling us to systematically test and refine our approach.

This methodical application of TTA not only substantiates the flexibility and adaptability of our approach but also significantly enhances the accuracy and reliability of our damage detection model, ensuring that the structural assessments are both precise and consistent.

```

45 def img_setup(img, flip_leftright=True, flip_updown=True, rotate90=True, rotate180=True, rotate270=True):
46     img_resized = np.copy(img)
47     img = img.astype(np.float32)
48     img /= 255 # 0~1
49     img = img.transpose((2, 0, 1)) # HWC -> CHW #
50     img_tensor = torch.from_numpy(img) # numpy -> tensor # 3,1028,720
51     img_tensor = torch.unsqueeze(img_tensor, 0) # add a batch dimension # 1,3,1028,720
52
53     with torch.no_grad():
54         img_variable = torch.autograd.Variable(img_tensor) # type change to Variable
55         if torch.cuda.is_available():
56             img_variable = img_variable.cuda() #type change to cuda
57         img_out = model(img_variable)
58
59         Lovasz=True
60         if Lovasz:
61             # classMap_numpy = (img_out[0].data.cpu() > 0.001).numpy()[0] # 0.001 is the threshold.
62             classMap_numpy = (img_out[0].data.cpu() > 0.001).numpy()[0] # lower the threshold for more detection
63         else:
64             classMap_numpy = img_out[0].max(0)[1].byte().data.cpu().numpy()
65
66         # idx_fg = (classMap_numpy == 1).astype(np.float)
67         idx_fg = (classMap_numpy == 1).astype(float) # 1 channel mask
68         mask = idx_fg.astype(np.uint8)*255 # 1 channel mask 0~255
69
70         if flip_leftright:
71             mask=cv2.flip(mask,1)
72             mask2=mask.copy()
73
74         if flip_updown:
75             mask=cv2.flip(mask,0)
76             mask2=mask.copy()
77
78         if rotate90:
79             mask=cv2.rotate(mask, cv2.ROTATE_90_COUNTERCLOCKWISE)
80             mask2=mask.copy()
81
82         if rotate180:
83             mask=cv2.rotate(mask, cv2.ROTATE_180)
84             mask2=mask.copy()
85
86         if rotate270:
87             mask=cv2.rotate(mask, cv2.ROTATE_90_COUNTERCLOCKWISE)
88             mask2=mask.copy()
89
90         else:
91             mask2=mask.copy()
92         # mask resize to original size
93         mask=cv2.resize(mask, (int(w),int(h))) #value of mask is 0 or 255
94         mask2=mask.copy()
95
96

```

Figure 14. Code for TTA.

3.3.2 STRNet

STRNet is specifically engineered for the task of crack detection, leveraging a sophisticated architecture that combines an attention-based encoder with a multi-head attention-based decoder.

This design incorporates several innovative elements to streamline the network while maintaining its high-speed processing capabilities.

At the core of STRNet is its initial convolution block, which processes the input images using the Hswish activation function, a variant known for its efficiency in deep learning tasks (Avenash and Viswanath, 2019), coupled with batch normalization to stabilize learning (Ioffe and Szegedy, 2015). The processed feature map from this block then enters the STR module. This STR module, which significantly enhances the network's ability to discern relevant features for crack detection, is iterated 11 times to progressively refine the feature details.

Post-STR module processing, the output undergoes max pooling to reduce spatial dimensions while preserving important features, which is then channeled into the attention-based decoder. This decoder utilizes skipped connections, a technique essential for preserving spatial hierarchies of features, which facilitates more accurate reconstructions in the up-sampling phases. The data undergoes two types of up-sampling: standard and coarse. Standard up-sampling is performed twice to gradually recover the data to the size of the input image, while integrating the detailed features preserved by the skipped connections.

The culmination of the STRNet's processing pipeline is in the final convolution stage. Here, outputs of standard up-sampling block, the initial convolution block, and the coarse up-sampling block are concatenated. This comprehensive concatenation is crucial as it combines

detailed and broader spatial features. Next, the concatenated output is refined through pointwise convolution (PW), effectively segmenting cracks at a pixel level with high precision.

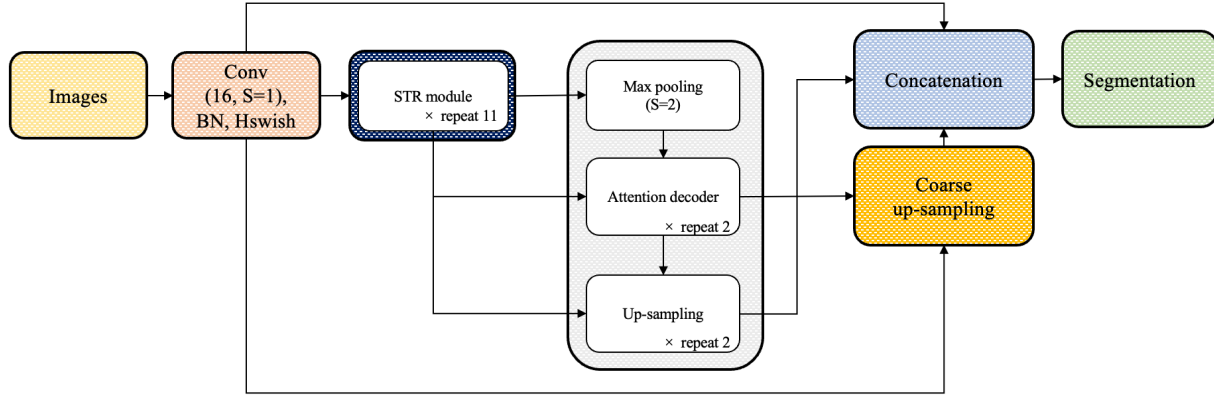


Figure 15. The STRNet architecture (Kang and Cha, 2022).

Figure 15 illustrates the complete STRNet architecture, highlighting each component from the initial convolution through to the final output stage. This visual representation underscores the complexity and the efficiency of the network, providing a clear view of how each section contributes to the task of high-precision crack detection.

3.4 NeRF for 3D reconstruction

The adoption of 3D reconstruction in our research enhances visualization, provides accurate damage assessment, and extends functionality beyond traditional 2D methodologies. This technology allows for the transformation of reconstructed models into mesh OBJ files, facilitating a variety of applications including digital twins and FEA.

At the forefront of 3D modeling technologies, NeRF employs a novel approach to synthesize lifelike 3D models from sets of 2D images. Unlike conventional methods, NeRF constructs scenes

using continuous volumetric functions, which have been shown to produce detailed and realistic renderings from any viewpoint (Mildenhall et al., 2021). Central to NeRF’s effectiveness is its processing of 5D coordinates, which combine spatial positions (x, y, z) with viewing directions (θ, φ) . This capability allows NeRF to generate outputs that include precise color and density values, thereby creating visually detailed 3d models.

NeRF distinguishes itself by its ability to reconstruct scenes from sparse and irregularly spaced images, which is particularly advantageous in data-limited scenarios (Müller et al., 2022). Its advanced modeling extends to the accurate representation of flat surfaces devoid of notable features—a significant challenge in traditional photogrammetry methods (Cheng et al., 2021). Furthermore, NeRF explicitly models complex lighting and shading effects, which dramatically improves the accuracy of reconstructing surfaces with intricate reflectance properties (Tancik et al., 2023). This robust capability not only enhances the visual quality but also ensures consistency across different reconstructions, marking a substantial improvement over older techniques.

NeRF's integration of deep learning algorithms also allows it to learn from a wide range of scenes, making it adaptable to various applications beyond mere visual reconstructions. This adaptability is underpinned by its capacity to interpolate and extrapolate from limited views, generating coherent 3D spaces from seemingly incomplete 2D data—a significant advantage for practical implementations in real-world settings.

3.4.1 NeRF overview

NeRF is an advanced method initially developed for view synthesis, which entails constructing 3D models from 2D images. This technique models scenes using continuous volumetric functions, which enable the creation of highly detailed and photorealistic renderings

from new, previously unseen viewpoints. As shown in Figure 16, the NeRF model employs a function, defined by Equation (10), which processes 5D coordinates—encompassing both spatial positions ($\mathbf{x}$) and directional vectors ($\mathbf{d}$):

$$F_{\Theta}: (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma) \quad (10)$$

Here, the function outputs color ($\mathbf{c}$) and density (σ), the latter being the reciprocal of transparency.

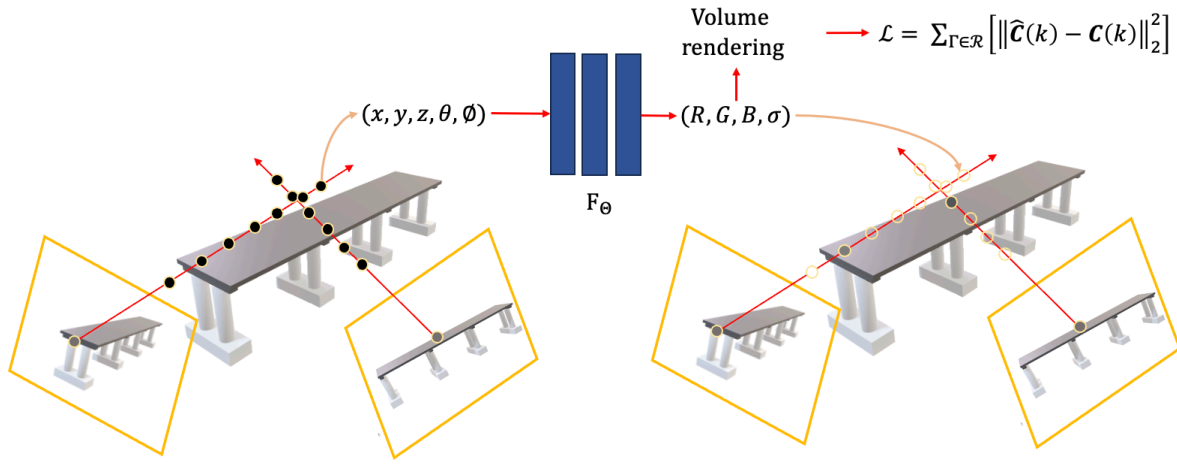


Figure 16. Overview of the NeRF model from the original NeRF paper (Mildenhall et al., 2021).

The RGB color value C at a pixel is determined through volume rendering, formulated in Equation (11):

$$\mathbf{C}(\mathbf{r}) = \int_{t_n}^{t_f} \mathbf{T}(t) \cdot \sigma(\mathbf{r}(t)) \cdot \mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt \quad (\text{Kajiya and Herzen, 1984}) \quad (11)$$

In this equation:

- $\mathbf{C}$ represents the integrated color along a ray.
- $\mathbf{c}$ and σ are the color and volume density at point $\mathbf{r}(t)$ along ray $\mathbf{r}$, $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$.

- $T(t)$, the transmittance, accumulates along ray r from t_n to t , decreasing as the ray passes through denser media.

$$T(t) = \exp \left(- \int_{t_n}^t \sigma(r(s)) ds \right) \quad (12)$$

This model calculates transmittance $T(t)$ to weigh the color contribution of each point along the ray, where a higher density suggests greater opacity and lower transmittance, affecting how much of the subsequent points' colors influence the final pixel color.

The practical implementation of NeRF approximates these integrals in discrete form, as shown in Equation (13):

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-s_i \delta_i)) \mathbf{c}_i, \text{ (Max, 1995)} \quad (13)$$

Where T_i is the transmittance computed up to segment i :

$$T_i = \exp \left(- \sum_{j=1}^{i-1} s_j \delta_j \right) \quad (14)$$

To evaluate the model's accuracy, the predicted color $\hat{\mathbf{C}}$ is compared against actual pixel values from a subset of images (1,717 out of 1,931) using a loss function:

$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2 \right] \quad (15)$$

This loss function guides the training process, optimizing the neural network to reduce discrepancies between the predicted and actual colors, thus refining the model's capability to render scenes with high fidelity.

3.4.2 NeRF model structure

The NeRF model is built on a fully connected neural network designed to interpret spatial and directional data to estimate the scene’s volume density and color. The architecture consists of nine layers, each with 256 channels, followed by a final layer with 128 channels, all utilizing ReLU activation functions. This configuration is depicted in Figure 17, titled “NeRF model structure,” illustrates these layers and their sequence.

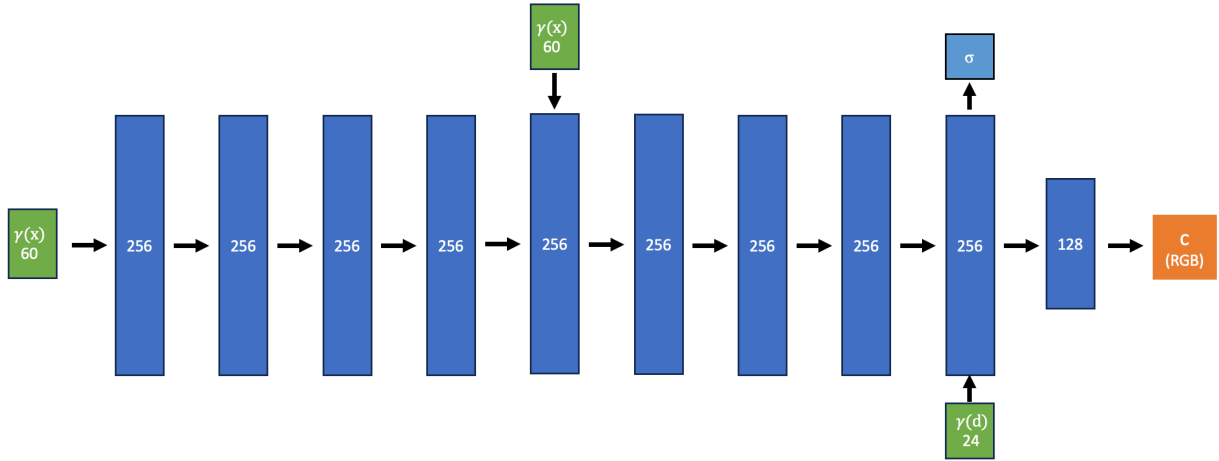


Figure 17. NeRF model structure.

Inputs to the network include (x, y, z) coordinates that represent each point on a ray, as depicted in Figure 18. This process involves shooting a ray in the viewing direction (θ, ϕ) at each 2D pixel coordinate and sampling N 3D point coordinates along the ray. Each i^{th} 3D point coordinate, along with the viewing direction, is fed into the MLP to calculate the respective volume density (σ) and radiance ($c_i = \text{RGB}$).

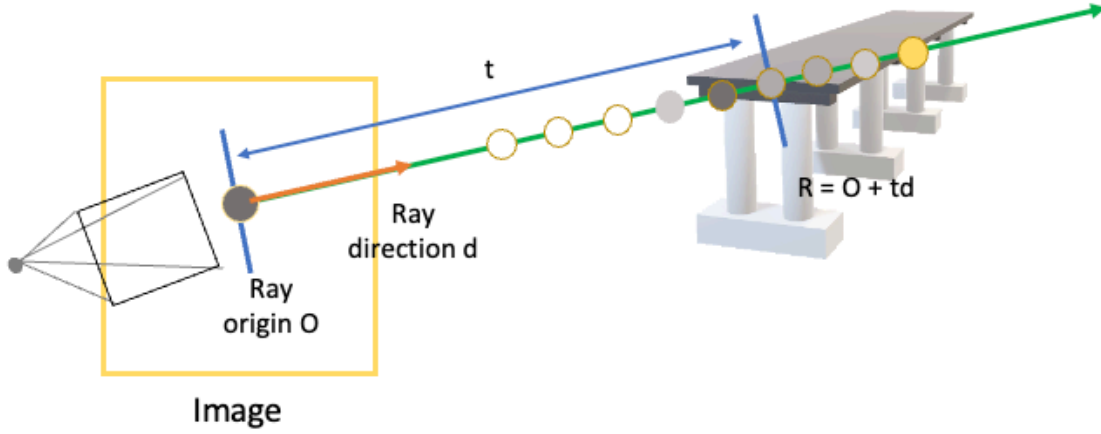


Figure 18. Ray casting process.

To enhance the model's performance to discern and reproduce high frequency details, positional encoding is used for input data, expanding it to 60 dimensions for 3D coordinates and 24 dimensions for viewing direction. Figure 19 visually illustrates this transformation of input data, showing how this step increases the input data's dimension of neural network for detail and accuracy. The equation governing positional encoding is shown below:

$$\boldsymbol{\gamma}(\mathbf{x}) = [\sin(\omega_1 x), \cos(\omega_1 x), \sin(\omega_2 x), \cos(\omega_2 x), \dots, \sin(\omega_n x), \cos(\omega_n x)] \quad (16)$$

where $\omega_1, \omega_2, \dots, \omega_n$ are frequencies chosen based on the data scale and model requirements. This expansion enables NeRF to accurately capture the intricacies of the scene's geometry and appearance, which are for rendering photorealistic images.

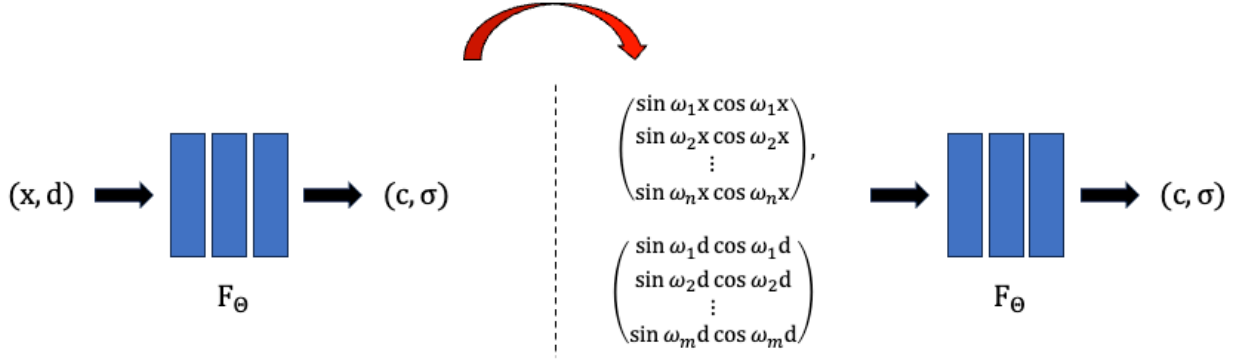


Figure 19. Positional encoding.

The design of the network ensures consistency across multiple views. It predicts volume density σ based on location alone, while RGB color c is determined by both the location and the viewing direction, allowing NeRF to maintain the color and structural integrity of the scene across various angles.

3.4.3 Network structure

NeRF employs a dual-network architecture comprising a coarse network and a fine network, each playing a critical role in the rendering process. The coarse network initially processes N_c samples along a ray, determining points of significant density, which suggest prominent features within the scene. These critical points, highlighted by the weights from the coarse network, guide the sampling strategy for the fine network. The fine network then combines

these weighted samples, N_f , with the original N_c samples for further refinement, enhancing the accuracy and detail of the final image.

Figure 20 depicts how data points are sampled along a ray. The sampling process in NeRF involves discretizing the continuous ray into N equal segments and performing random sampling within these segments. This method, termed stratified sampling, is represented by the Equation (17):

$$t_i \sim \mathcal{U} \left[t_n + \frac{i-1}{N} (t_f - t_n), \quad t_n + \frac{i}{N} (t_f - t_n) \right] \quad (17)$$

This stratified approach allows for variable distribution within each segment, reducing bias and improving the representativeness of the sample points along the ray compared to uniform sampling.

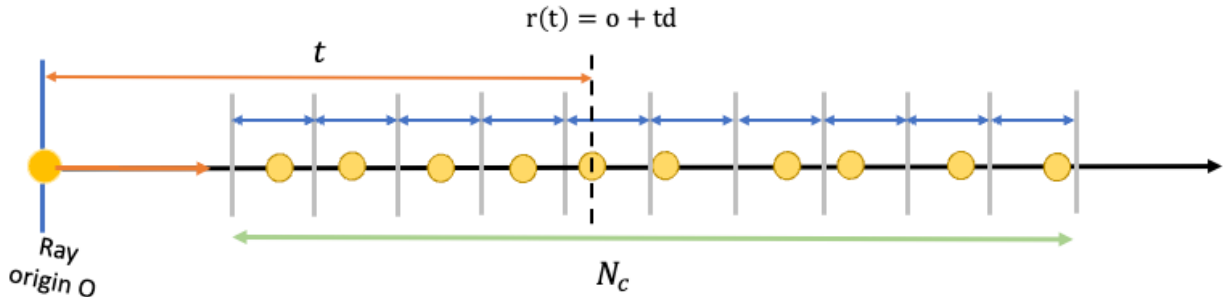


Figure 20. Stratified sampling.

The model's overall performance is evaluated by a composite loss function, which aggregates the discrepancies between the predicted and actual colors from both networks. This is formulated as:

$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\|\hat{\mathbf{C}}_c(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2 + \|\hat{\mathbf{C}}_f(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2 \right] \quad (18)$$

This dual-network loss calculation ensures that both the initial approximations by the coarse network and the refined predictions by the fine network are optimized, leading to more precise and consistent color renderings across different views.

Moreover, the greater the number of similar images of an object, the more likely it is that the model will consistently predict accurate colors. This consistency is vital as it enhances the training process, contributing significantly to the model's ability to faithfully reproduce the actual colors observed in the real world.

3.4.4 Nerfstudio framework

Our research employs the NeRF-based Nerfacto model from the versatile and modular Nerfstudio framework, recognized for integrating a variety of NeRF technologies into a cohesive system, such as Mip-NeRF 360 (Barron et al., 2022), Instant-NGP (Müller et al., 2022), NeRF-W (Martin-Brualla et al., 2021) and Ref-NeRF (Verbin et al., 2021). This framework facilitates real-time scene visualization and simplifies the creation of NeRFs from user-captured data, making it a prime choice for both academic and industry applications (Tancik et al., 2023). key advantage of using Nerfacto was computational efficiency and its capability to amalgamate images taken under diverse lighting conditions, essential for accurately reconstructing large structures photographed at different times of the day. The appearance embedding functionality of Nerfacto ensures consistent quality in the final reconstruction.

In our application of Nerfacto, we made several key adjustments to optimize the model's hyperparameters for our specific needs. Notable changes included increasing the maximum

number of iterations from 30,000 to 100,000 to capture deeper detail and expanding the number of rays processed per training batch from 4,096 to 16,384 to enhance data analysis per iteration. These adjustments, detailed in Table 1, significantly improved the quality of our 3D reconstructions.

Table 1. Changed hyperparameter values.

Parameter	Default setting	Our setting
The steps per evaluation batch	500	2,000
Frequency of saving progress	Every 500 steps	Every 2,000 steps
number of iterations	30,000	100,000
the number of rays processed per training batch	4,096	16,384
The number of rays per evaluation batch	4,096	4,096

Real-time training progress monitoring is facilitated through a web URL, where initial renderings gradually clarify as the model learns the scene's color details, shown in Figure 21. This visualization helps assess the evolving accuracy of the model.

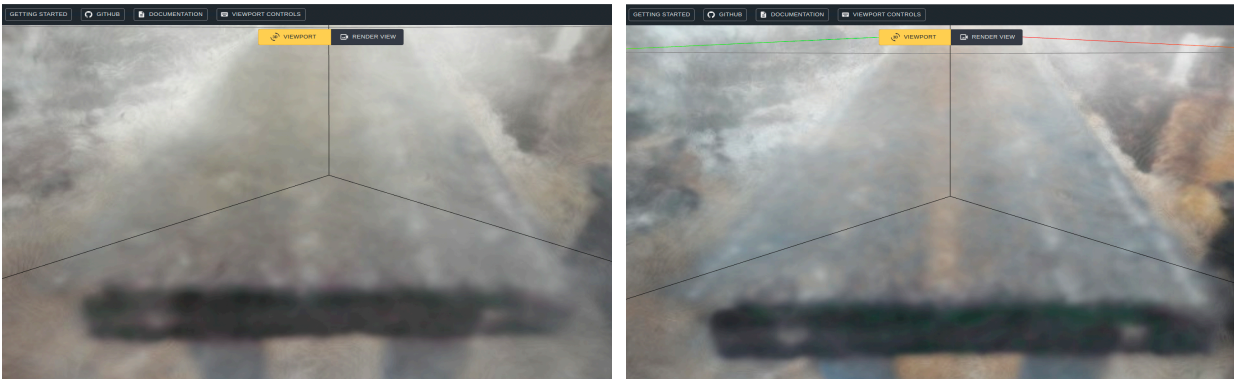




Figure 21. Training progress within the nerfstudio real-time visualizer module web URL.

Clockwise from top left, the progression of the training (Kim and Cha, 2024).

Following this, to demonstrate the practical results of these improvements, Figure 22 can be introduced to illustrate how effectively the model aligns images based on the estimated camera poses. This step is crucial for verifying that the model not only enhances visual quality over training iterations but also accurately aligns multiple views according to the reconstructed spatial geometry.

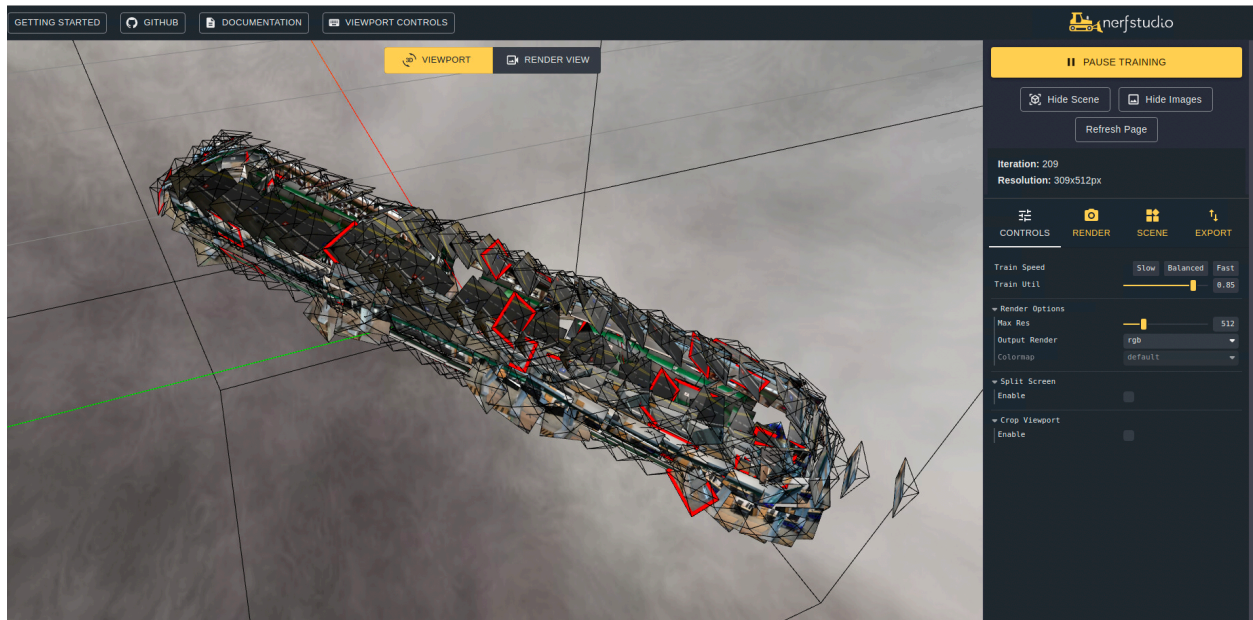


Figure 22. Camera pose alignment visualization (Kim and Cha, 2024).

This figure visually confirms the model’s ability to accurately estimate and utilize camera poses, which is essential for achieving consistent reconstruction quality across different views. Such alignment is indicative of the model's robust handling of spatial data and its effectiveness in synthesizing accurate 3D scenes from various camera angles.

Upon completion, the model supports various export formats, including video and point cloud files. For our purposes, exporting the model as a mesh OBJ file was particularly beneficial for integration into virtual environments and for future simulation research. Adjustments to the model before exporting, such as cropping to highlight areas of interest, can be performed directly via Nerfstudio’s interface, as shown in Figure 23.

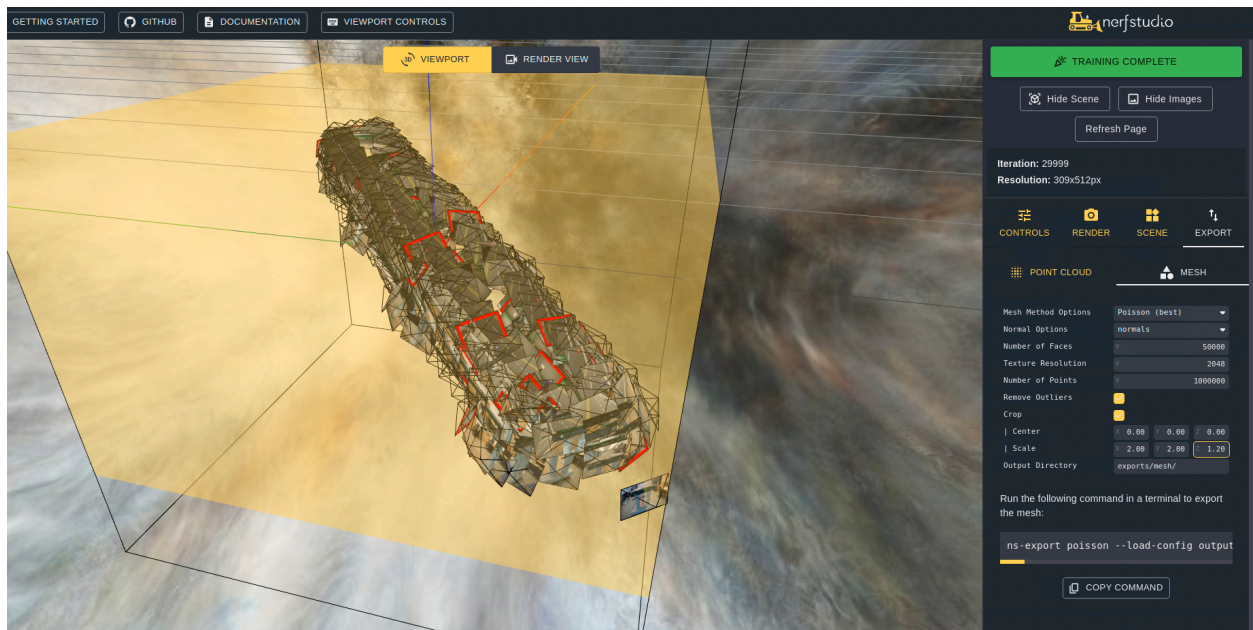


Figure 23. Crop adjustment in NeRF (Kim and Cha, 2024).

With the comprehensive processes outlined above, from the strategic adjustments in the Nerfacto model's hyperparameters to the meticulous alignment of camera poses, we successfully synthesize a highly detailed and accurate 3D model. This model is exported in OBJ file format, which is ideal for integrating into our digital twin framework. This format ensures compatibility with various visualization. It also enhances collaborative efforts and further simulations, laying a robust foundation for future research and applications.

3.5 Damage mapping

Our damage mapping methodology integrates sophisticated imaging and processing techniques to ensure detailed and precise localization of structural damage on 3D models, which is essential for assessing the integrity and planning the maintenance of infrastructure like bridges.

The process begins with the input of high-resolution image data into Metashape, a powerful tool for performing SfM. This initial step is crucial as it generates an XML file consisting of transformation matrices for each image, for encoding the spatial information of the images relative to one another, as illustrated in Figure 12 (a) for the XML format. To ensure compatibility with the Nerfstudio framework, this XML file is converted into JSON format, as shown in Figure 12 (b). This conversion, facilitated by the `agi2nerf.py` script, is critical for enabling the subsequent integration of segmented damage data into Nerfstudio.

Following this conversion, the image datasets, now prepared with transformation data, are integrated back into STRNet to utilize damage segmentation. This phase involves combining the images, now augmented with damage details, with the `transforms.json` file and inputting them into Nerfstudio. The sequential workflow of our damage mapping process is depicted in Figure 24,

where (a) shows our method of camera pose estimation before damage segmentation, while (b) illustrates an alternative approach where the transforms.json file is extracted alongside damage-segmented images.

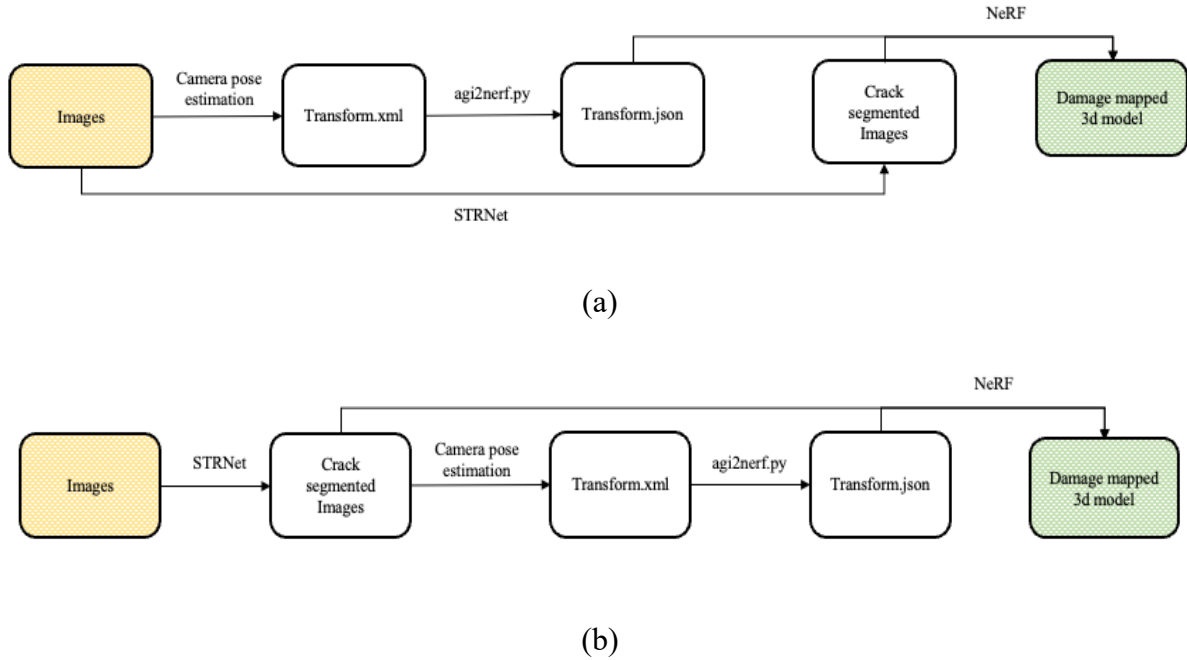
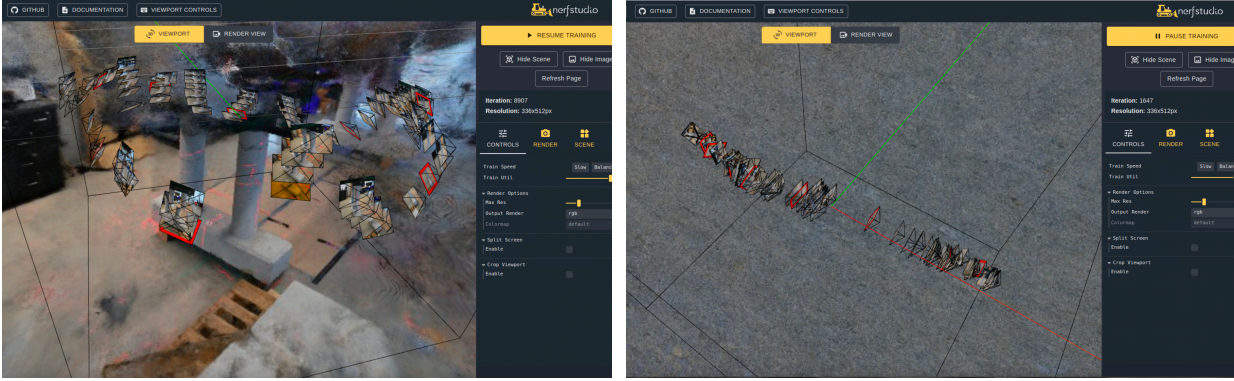


Figure 24. Sequential workflow of damage mapping. (a) our method: camera pose estimation before damage segmentation, (b) alternative approach: camera pose estimation with crack segmented images (Kim and Cha, 2024).

To prevent errors in camera pose estimation that could arise from uneven damage segmentation, we made the strategic decision to prepare the transforms.json file prior to segmenting damage. By prioritizing camera pose estimation before addressing damage, we safeguarded the accuracy of camera positioning against the potential inconsistencies in damage appearance. Figure 25 underscores this methodology: panel (a) shows well-aligned images, indicative of precise camera pose estimation using our approach. Conversely, panel (b) displays images with noticeable alignment issues due to the sequence of performing damage segmentation

before camera pose estimation. Such misalignment leads to flawed ray tracing across the images, adversely affecting the 3D reconstruction's quality. This specific issue was observed during our analysis of a dataset comprising 85 images dedicated to the reconstruction of two bridge columns.



(a) Well-aligned images (our method)

(b) Misaligned images

Figure 25. Aligned images based on camera poses (Kim and Cha, 2024).

Ultimately, our approach to damage mapping does more than just reconstruct 3D models; it effectively maps structural damages, like cracks, over the model. This process not only allows for the precise quantification of damage sizes but also aids in the intuitive localization of damages on the bridge model. More realistic view of the structural integrity of the bridge can be accomplished by expressing these damages in a 3D context, which is invaluable for maintenance planning and safety assessments.

4 Data collection

The success of any SHM and 3D reconstruction project fundamentally depends on the quality of the data collected. This chapter underscores the pivotal role of meticulous data collection within our methodological framework. Good quality data form the cornerstone on which we build to guarantee the precision of our outputs. Detailed and precise data are essential for enhancing the quality of 3D reconstructions as well as robust damage detection.

To achieve good quality in 3D reconstruction, there are a few key things to consider carefully.:

- **Lighting conditions:** Clear and consistent lighting is essential, as it significantly impacts the quality of the captured images, affecting both the visibility of features and the consistency of color (Dandois et al., 2015).
- **Image overlap:** We recommend maintaining at least 80% overlap between consecutive images to improve the accuracy and detail of the reconstructions, facilitating more reliable feature integration across images (Dandois et al., 2015).
- **Reflective surfaces:** Reflective objects can introduce glare and reflections that distort the data, posing challenges in photogrammetry (Karami et al., 2022).
- **Camera quality:** The resolution and capabilities of the camera sensor directly influence the detail captured in images. High-resolution cameras equipped with advanced sensors are crucial for capturing detailed nuances necessary for high-quality reconstructions (Karami et al., 2022).
- **Stability and clarity:** Ensuring stable conditions is crucial to minimize motion blur and maintain uniform image quality, which is essential for accurate camera pose estimation and image alignment (Lee and Lee et al., 2013).

- Use of markers: Employing trackable markers in the images greatly enhances the reconstruction process. These markers serve as precise reference points, aiding in the accurate alignment and stitching of the images, and consequently, the production of more accurate 3D models (Majidi et al., 2023).

In summary, optimizing the 3D photogrammetric reconstruction process involves meticulous attention to the interplay of lighting, image overlap, camera specifications, environmental conditions, and strategic use of markers. Each of these elements is critical for capturing the high-quality data required to produce reliable and detailed 3D models.

The subsequent subsections from 4.1 to 4.5 will delve into each of these areas in detail, providing a comprehensive discussion of the methods and techniques employed in our data collection efforts to ensure the highest quality of 3D reconstruction possible.

4.1 Image data collection

our approach to damage detection incorporates the use of a pretrained STRNet, developed and initially trained by Kang and Cha (2022). This module utilizes a diverse and extensive dataset specifically tailored for crack detection. The effectiveness of STRNet in identifying and segmenting cracks is crucial for assessing structural integrity, making this dataset invaluable to our efforts in SHM.

The datasets used for training and testing STRNet include a total of 1,748 images, segmented between the training and testing phases with resolutions designed to meet the task requirements. The training set consists of 1,203 images at a resolution of 1024×512 pixels, while the test set includes 545 images at two resolutions, 1024×512 and 1280×720 pixels. This configuration allows us to validate the network's effectiveness across different image qualities.

To further refine the model's accuracy under varied conditions, additional tests and pre-tests using TTA were conducted on high-resolution images (2992×2000 pixels). The distribution of images across these datasets is detailed in Table 2.

Table 2. Datasets in each case.

Cases	Image size (px)	Number of images
STRNet (Training)	1024×512	1203
STRNet (Test)	1024×512, 1280×720	545
STRNet + TTA (Pre-test)	2992×2000	33
STRNet + TTA (Test)	2992×2000	100
NeRF for 3D reconstruction	2992×2000	1931

For 3D reconstruction using NeRF, we compiled a substantial dataset of 1,931 high-resolution images (2992×2000 pixels). This large dataset is crucial for achieving detailed and accurate 3D models, providing multiple views and extensive coverage that enhance the depth and quality of the reconstruction.

4.2 Camera specifications and setup

For capturing high-resolution images essential for our SHM and 3D reconstruction efforts, we utilized a Nikon digital camera. The specific camera and settings employed are illustrated in Figure 26 and detailed in Table 3 below.

To optimize the balance between image quality and computational efficiency, we configured the camera to shoot at a resolution of 2992×2000 pixels. The focal length was fixed at 18mm to provide a wide field of view while capturing sufficient detail. We selected the JPEG fine format for our image files, which offers a good compromise between high image quality and manageable file size, keeping each image under 5MB. This consideration is crucial as it allows for efficient processing without sacrificing the quality necessary for detailed analysis.

The decision to set the resolution at 2992×2000 pixels emerged from multiple trials that considered the relationship between computational cost and the number of images needed for accurate 3D reconstruction. While higher pixel density is significant for enhancing model quality, it increases the computational load, which in turn limits the number of images that can be processed efficiently. However, capturing a higher number of images is also crucial as it allows for the extraction of consistent features across multiple views, improving the accuracy of camera pose estimations. Furthermore, frequent appearances of the same color pixels in the dataset enable NeRF to assign higher weights to these observations, leading to better color accuracy in the 3D model reconstruction.

Table 3. Camera settings

Model name	Image size (px)	Focal length	File format
Nikon D7200	2992×2000	18mm	JPEG



(a)



(b)



(c)

Figure 26. Camera setting. (a) Nikon digital camera (b) Image quality setting, (c) Image size setting.

4.3 Controlled environment and setup

The experimental setup was meticulously designed within an indoor laboratory to control for variables that could affect image quality. One significant challenge was the lab's reflective flooring, which could distort images taken at various angles due to unwanted reflections.

To mitigate reflections caused by the lab's reflective flooring, we strategically placed boxes and wooden panels beneath the bridge model, securing them with tape to ensure they remained immobile between shots. This setup was essential not only for blocking reflections but also for maintaining the positional consistency of these objects, critical for accurate camera pose estimation. Figure 27 illustrates our setup adjustments to minimize reflections. The blinds were lowered as shown in Figure 27 (a) to control light exposure further, ensuring uniform lighting conditions.



Figure 27. Mitigation of reflective floor surfaces in the indoor lab (a) shows the reflective floor surface (b) shows coverage of reflective surfaces by putting boxes on the floor.

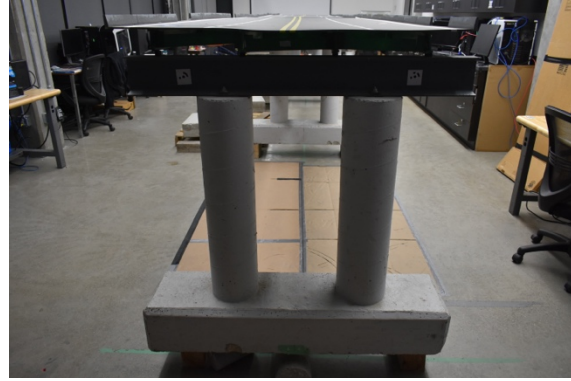
While LED lighting provided a stable light source, it was insufficient for illuminating the underside of the bridge model effectively. To compensate for this, images of the bridge's bottom were captured during times of sufficient sunlight, ensuring that the colors and details were adequately represented. Figure 28 demonstrates the variance in image quality between different lighting conditions: daylight brought out bright, clear colors, whereas nighttime images appeared significantly darker and less detailed.

On the other hand, for capturing images of the top side, it was ideal to take photos when the sun was at its zenith to avoid direct sunlight coming through the windows, or during cloudy

weather, which helps prevent inconsistencies caused by varying lighting conditions. Figure 29 illustrates how direct sunlight can create unwanted lines and glare when it enters through the windows.



(a)



(b)

Figure 28. Image quality variation (a) Daytime with sufficient sunlight enhancing color visibility. (b) Nighttime showing diminished light and color quality.



(a)



(b)

Figure 29. Lighting effects on top side (a) Unwanted lines and glare from direct sunlight. (b) Preferred diffuse lighting on a cloudy day or during nighttime.

4.4 Marker setting

In our experiments, the precision of camera pose estimation was significantly enhanced using 154 markers, as demonstrated in Figure 30. These markers are designed by Metashape software tool. We strategically placed these markers on the top, bottom, and sides of the bridge,

illustrated in Figure 31. Their dual function was essential: they provided accurate reference points for establishing the coordinate system and defining scale.

The markers are coded targets. These markers were strategically positioned in the scene before photography began. After that they were employed within the photogrammetry software, serving reference points and integral components of the image matching process, particularly through the 'Align Selected Cameras' feature. Before the implementation of these markers, issues with misaligned camera poses frequently occurred; these issues were substantially mitigated following their placement. We adhered to best practices by ensuring at least one marker is included in each captured image, significantly enhancing feature matching and spatial orientation.

The implementation of markers was shown to be particularly beneficial for the NeRF technique, offering reliable camera pose information, which are important for achieving accurate 3D reconstructions.



Figure 30. Markers utilized in the experiment (Agisoft Metashape Professional Edition, version 2.1).





(a) Front view of the bridge model and closeup

(b) Top view of the bridge model and closeup

(c) Bottom view of the bridge model and closeup.

Figure 31. Aligned Markers on the bridge (Kim and Cha, 2024).

4.5 Image capture methodology

Capturing images for 3D reconstruction demands meticulous attention to detail to ensure that each photograph adequately captures the desired features of the bridge model. Emphasis was placed on capturing close-up shots to maximize the number of image pixels representing the bridge, enhancing the quality and detail of the reconstruction. Figure 31 bottom part showcases examples of these close-up shots, demonstrating how such detailed captures contribute to a more precise and detailed 3D model.

To ensure consistency and continuity of the image data, strict guidelines were implemented to avoid sharp angles and minimize positional changes between shots. Significant emphasis was also placed on image overlap, ensuring that each photograph overlapped with its neighboring images by at least 80%. Figure 32 illustrates an example of this 80% image overlap, showing six images in a row, demonstrating how each image sufficiently overlaps with its adjacent images to ensure comprehensive coverage and detail accuracy. This careful approach not only facilitates the acquisition of accurate camera poses (viewing directions) through SfM but is also crucial for improve the quality of the final 3D model created employing NeRF.

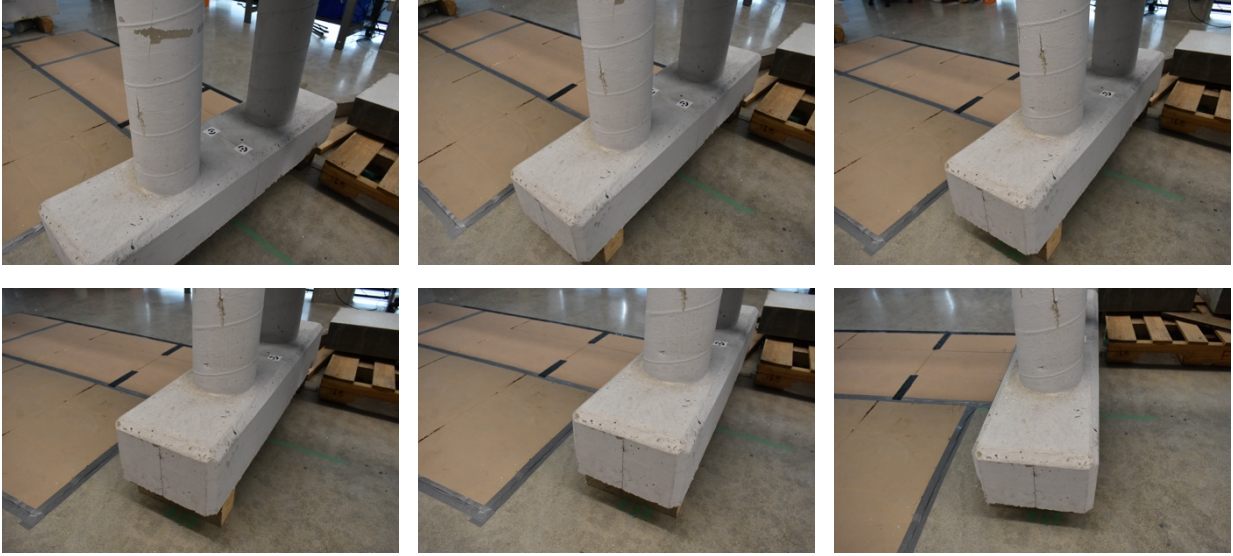
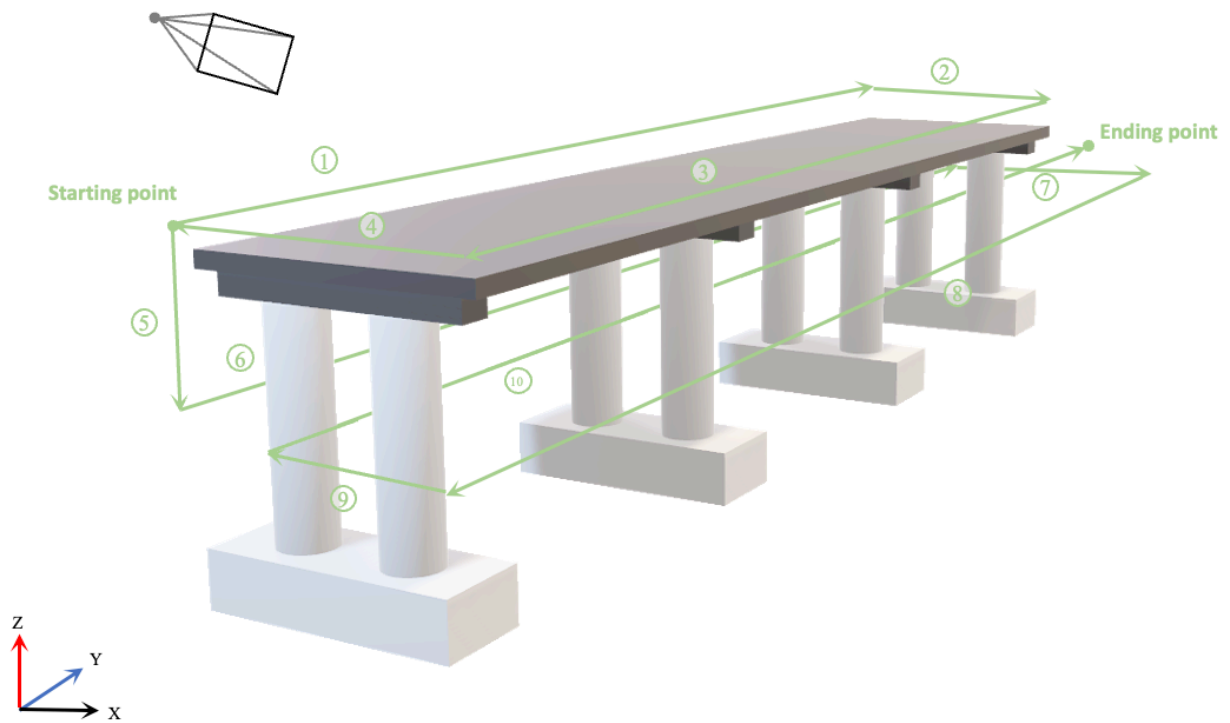


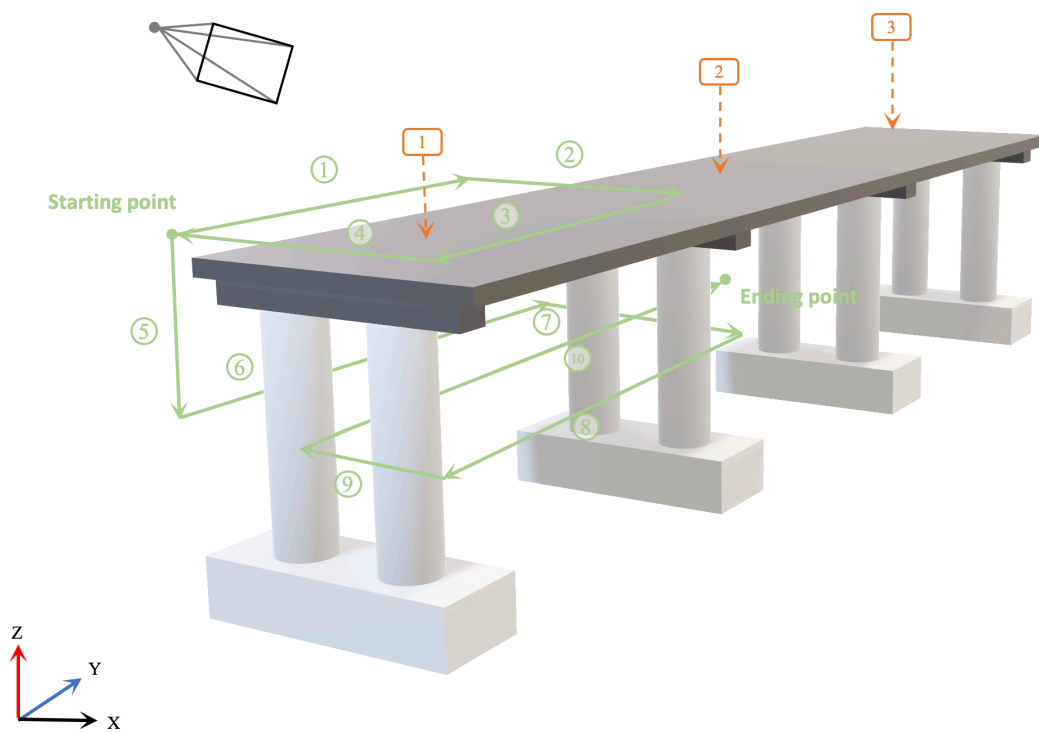
Figure 32. example for 80% image overlap of 6 images in a row.

Another challenge was the potential for motion blur, which can degrade the quality of 3D reconstructions by affecting camera pose estimation accuracy. To address this, we initially focused on meticulous capture techniques and constant quality checks during shoots. Recognizing the limitations of this method, especially for larger objects, we incorporated the use of NAFNet, a deblurring deep learning model, to enhance image clarity. This model and its application are detailed further in Section 3.1.

The methodology chosen for capturing images also played a significant role in the quality of the outcomes. Through experimentation, we determined that a sequential or circular method of capturing images of each bridge span individually yielded superior 3D reconstruction results compared to a method that attempted to capture three spans in one shot. Figure 33 depicts the different approaches we tested.



(a) Three spans at once route of photography



(b) Each span route of photography

Figure 33. Photography route methods (Kim and Cha, 2024).

This strategic approach to image capture provided multiple views of the same object, ensuring a comprehensive and consistent 3D model, which was essential for consistent color mapping within the NeRF system. The presence of more images capturing similar features increases the likelihood of consistent color observation, which, in turn, enhances the weight assigned to true colors during the model training phase.

5 Case studies

In this chapter, we introduce a collection of case studies that empirically validate our methods and illustrate the application of a digital twin in SHM. These studies highlight the use of advanced techniques like 3D reconstruction using NeRF and enhanced crack segmentation via STRNet with TTA. Each case study aims to underscore various facets of our methodology, ranging from the preliminary experimental setup to the refinement of processing techniques that enhance the outcomes. These case studies aim to offer an in-depth exploration of the diverse factors that influence the accuracy and quality of 3D reconstruction and damage detection in structural models.

Section 5.1 will detail the laboratory and computational setups used in the experiments. It covers descriptions of the environments where the bridge models were examined and outlines the specifications of the computer systems utilized for the data processing.

In Section 5.2, we evaluate the performance of crack segmentation using STRNet enhanced with TTA. This section will delve into the effectiveness of the technique and its implications for accurate damage detection.

Section 5.3 presents the details and results of the 3D reconstructions obtained through NeRF. We will explore how various factors such as the number of photos, image resolution, image deblurring, image capture route strategies, and hyperparameter adjustments affect the fidelity of the reconstructed models. Additionally, this section will include a quantitative evaluation of each scenario, utilizing metrics such as PSNR, SSIM, MSE to gauge the quality of the reconstructions.

Finally, Section 5.4 will showcase the results of incorporating segmented crack data into the 3D models, highlighting the practical application of our integrated approach to SHM.

Through these case studies, we demonstrate a real-world application of our method, combining cutting-edge deep learning techniques for damage detection with sophisticated 3D reconstruction to facilitate a digital twin in SHM.

5.1 Experimental detail

- **Research Location and Equipment:** Our experiments were conducted in a specialized indoor laboratory at the University of Manitoba, using a bridge model specifically crafted to include dimensions that facilitate both damage simulation and photogrammetry. Artificial cracks were methodically introduced to the bridge columns using a hammer and chisel to simulate various damage scenarios.
- **Computational Setup:** We utilized a high-performance desktop equipped with an AMD Ryzen 3990x CPU and an Nvidia RTX A6000 GPU, running on Ubuntu 20.04.6 LTS. This setup was crucial for handling the computationally intensive tasks associated with detailed 3D reconstruction and damage detection.
- **Software and Frameworks:** For 3D reconstruction, the Nerfacto model within the Nerfstudio framework was employed, renowned for its capability to handle complex volumetric reconstructions effectively. In addition, Metashape software was used for accurate photogrammetric processing and camera pose estimation, establishing a foundational dataset for further analysis in Nerfstudio.

5.2 STRNet + TTA evaluation

Initially, the STRNet model achieved a mIoU of 92% using a dataset of 545 images at resolutions of 1024×512 and 1280×720 , as shown in Table 4. However, when applied to a 33-

image dataset at the higher resolution of 2992×2000, used for 3D reconstruction, the mIoU significantly dropped to 73.5%. The mIoU is calculated as follows:

$$mIoU = mean\left(\frac{TP}{TP + FP + FN}\right) \quad (19)$$

Therefore, this was the challenge that the original STRNet model faced in effectively detecting cracks on high-resolution images (2992×2000). To overcome these limitations, we implemented TTA and assessed the improvements using the mIoU metric.

After the application of TTA, which involved combining results from various augmentation strategies developed through trial and error, the mIoU improved to 80.4%. Further validation was conducted using a larger dataset of 100 high-resolution images, where the mIoU for the original STRNet was 73.7%. Post-TTA application, the mIoU consistently increased to 80.4%, demonstrating the effectiveness of our augmentation techniques.

Table 4. Comparison mIoU with original STRNet and STRNet + TTA

Cases	Images size (px)	number	mIoU (%)
STRNet (Training)	1024×512	1,203	
STRNet (Test)	1024×512, 1280x720	545	92.5
STRNet (Pre-test)	2992×2000	33	73.5
STRNet (Pre-test) + TTA	2992×2000	33	80.4
STRNet (Test)	2992×2000	100	73.7
STRNet (Test) + TTA	2992×2000	100	80.3

As detailed in Table 5, we explored various augmentation combinations to determine their impact on mIoU. We defined several cases for this purpose: Case 1 involved resizing to 1280×720, Case 2 to 2048×1024, Cases 3 and 4 entailed flipping images left and right for both the 1280×720 and 2048x1024 sizes, respectively. Cases 5 and 6 included vertical flips (up and down) for each size. Rotation augmentations were applied in Cases 7 and 8 (rotate 90°), Cases 9 and 10 (rotate

180°), and Cases 11 and 12 (rotate 270°). For instance, the notation [1:2] denotes the combination of results from cases 1 and 2, while [1:12] represents the combination of all augmentation results.

Importantly, more combinations did not always yield better results, as observed in the slight mIoU drop from [1:7] to [1:8]. Therefore, examining all possible cases was crucial to identify the most effective augmentations. Our analysis concluded that the optimal result was achieved with the combination of [2:5] + [7] + [9] + [11]. This combination includes resizing images to 2048×1024, applying horizontal flip for both resized images (1280×720 and 2048×1024), vertical flip for resized images (1280×720), and rotating the resized (1280×720) images by 90, 180, and 270 degrees. This specific amalgamation of augmentations provided the most significant improvement in mIoU, as per our experimental findings.

Table 5. The mIoU results of each different TTA approached case.

Cases	mIoU (%)	Cases	mIoU (%)
[1:2]	76.1	[2:4]	79.2
[1:3]	78.7	[2:5]	79.6
[1:4]	79.2	[1:4] + [7]	80.1
[1:5]	79.6	[1:4] + [11]	80.0
[1:6]	79.7	[1:4] + [7] + [11]	80.1
[1:7]	80.1	[1:5] + [9]	80.0
[1:8]	80.0	[1:5] + [7] + [9] + [11]	80.4
[1:9]	80.1	[2:4] + [7] + [11]	80.1
[1:10]	80.2	[2:5] + [7]	80.2
[1:11]	80.2	[2:5] + [7] + [11]	80.2
[1:12]	80.2	[2:5] + [7] + [9] + [11]	80.4

Case 1 = resizing to (1280×720).	Case 7, 8 = 90 rotations for each of the two sizes.
Case 2 = resizing to (2048×1024).	Case 9, 10 = 180 rotations for each of the two sizes.
Case 3, 4 = horizontal flip (left, right) for each of the two sizes.	Case 11, 12 = 270 rotations for each of the two sizes.
Case 5, 6 = vertical flip (up, down) for each of the two sizes.	

Figure 34 visually demonstrates the impact of TTA on segmentation accuracy. Image (a) displays the segmentation result after resizing the original image from 2992×2000 to 1280×720, where the very right-side crack is only partially segmented. Image (b) shows the outcome after resizing to 2048×1024, successfully segmenting the very right-side crack fully, but the middle two cracks are less accurately segmented compared to (a). Image (c), resulting from the combination of (a) and (b), reveals improved segmentation of the middle two cracks and the very right-side crack. However, the two cracks on the very left-side remain unsegmented. Image (d), representing the combined strategy of [2:5] + [7] + [9] + [11], illustrates all five cracks fully segmented, aligning closely with the ground truth mask (e). Each step shows progressively more accurate crack segmentation, with image (d) showing all target cracks detected.

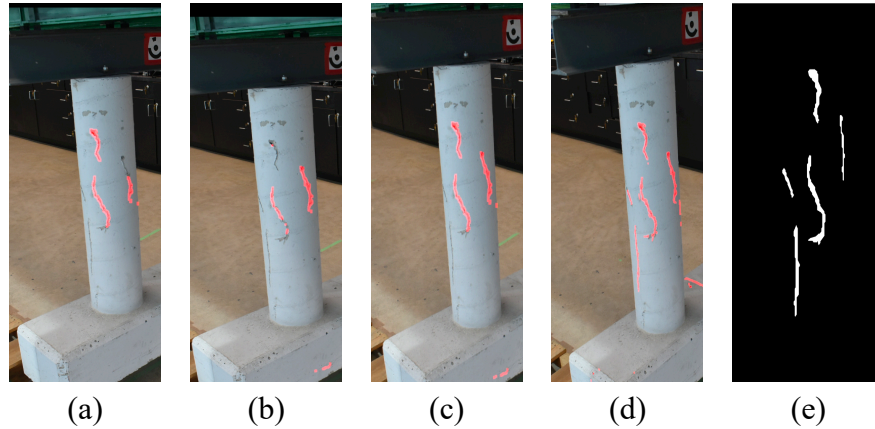


Figure 34. Crack segmentation results of each case. (a) The result of detection after resizing the image to 1280×720. (b) The result of detection after resizing the image to 2048×1024. (c) The result of combining the detections from (a) and (b). (d) The result of applying a combination of resizing to 1280×720 and 2048×1024, flipping horizontally for both sizes, flipping vertically for 1280×720, and rotating (90°, 180°, 270°) for (1280×720). (e) The Ground Truth mask.

5.3 3D reconstruction

The 3D reconstruction of the bridge was accomplished by capturing 1,931 images, each primarily capturing the bridge structure. These images were taken with an overlap exceeding 70~80%, in consistent and controlled lighting conditions, and were specifically arranged to minimize reflections. The images had a resolution of 2992×2000 pixels. We used these images, along with camera poses derived from Agisoft Metashape, to train the Nerfacto model for generating a 3D mesh of the bridge. We conducted multiple tests to determine the impact of different elements like image resolution, the number of images, clarity, photographic route, and hyperparameter adjustments on the quality of the 3D reconstruction.

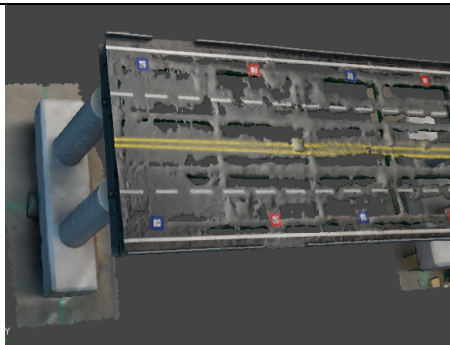
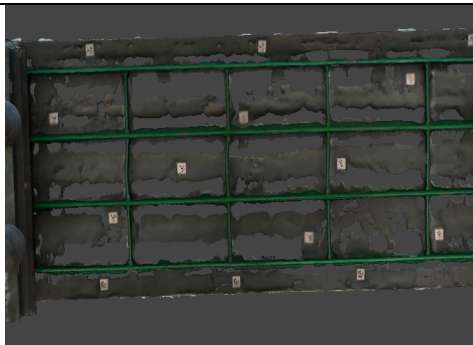
As depicted in Figure 35, eight different experimental setups were tested. The initial test employed a standard photogrammetry approach, which resulted in significant gaps in planar areas of the 3D model, leading us to abandon this method. The second test utilized 1,931 images of lower resolution (1024×720), which, while reducing gaps in planar surfaces, compromised the overall sharpness and quality of the reconstruction. This test highlighted the importance of resolution in capturing adequate features for high-quality 3D models.

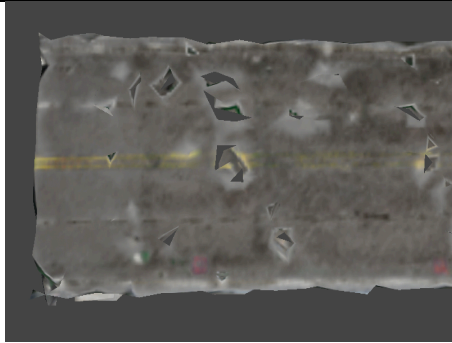
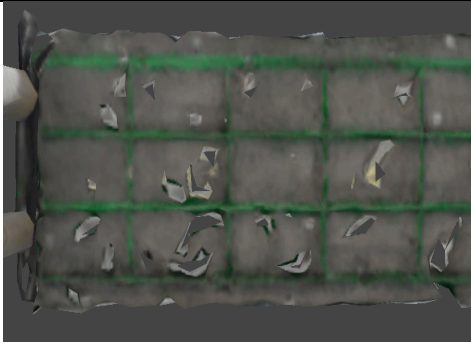
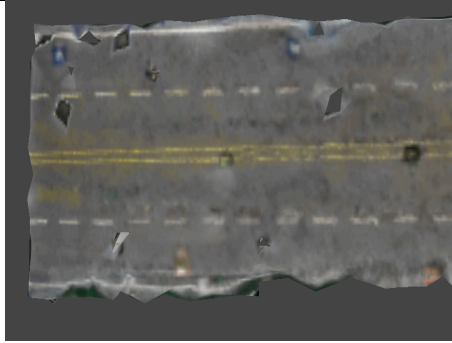
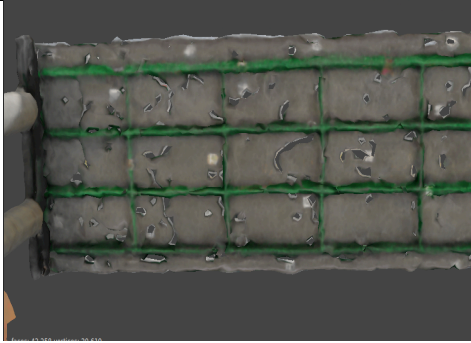
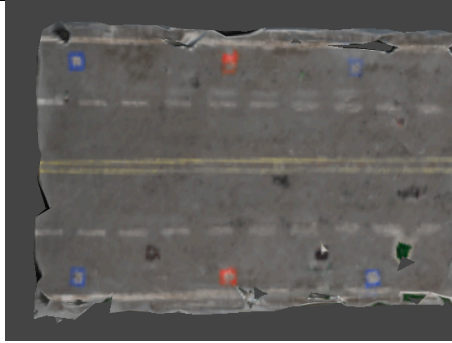
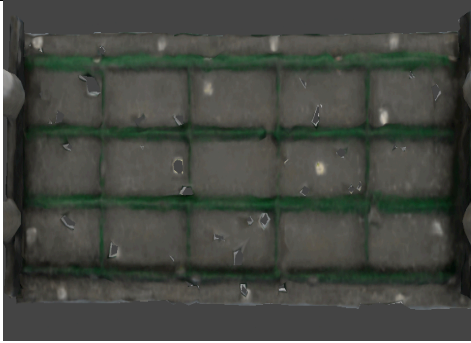
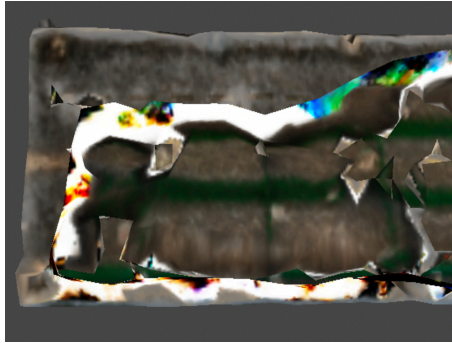
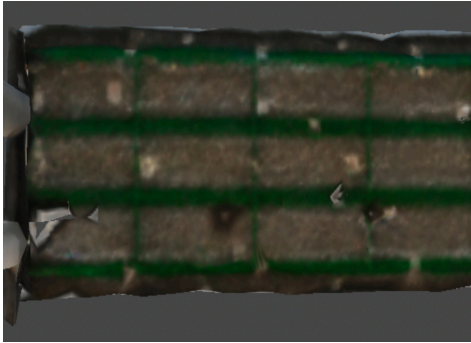
In the third experiment, we used fewer images (743) but at a higher resolution (2992×2000), which enhanced line clarity and reduced holes, though the limited number of images constrained the detail achievable across the dataset. This emphasized the need for a larger dataset when working with high-resolution images to ensure comprehensive training and learning.

Experiment five was setup with some blurred images within the dataset to check how the blurred image impact the overall 3d reconstruction quality. it introduced challenges with blurred

images of the bridge's upper surfaces, which introduced artifacts distorting the planar geometry. The sixth setup improved upon this by applying a deblurring algorithm, NAFNet, which significantly enhanced the sharpness of the 3D model, illustrating the detrimental effect of blur on reconstruction quality. Notably, motion blur often occurs when capturing large structures from a moving platform such as UAVs, and applying deblurring algorithms can be more practical than manual image inspection, especially with large datasets.

The photographic route also played a critical role; in experiment seven, capturing images of each bridge segment separately resulted in clearer and more distinct features compared to cases where multiple spans were captured in single shots. Finally, in the eighth experiment, applying default hyperparameter settings yielded clear and distinct features of the bridge. Subsequent optimization of hyperparameters, such as increasing the number of rays per batch from 4,096 to 16,384 and iterations from 30,000 to 100,000, further enhanced the detail and quality of the 3D model. This progressive optimization throughout the experiments culminated in a superior quality model by the final test.

	Top	Bottom
Experiment 1 • Photogrammetry software tool • Number of images: 1,931 • Resolution: 2992×2000 • Three spans at once route		
Experiment 2		

• NeRF • Number of images: 1,931 • Resolution: 1024×720 • Three spans at once route		
Experiment 3 • NeRF • Number of images: 743 • Resoulution: 2992×2000 • Three spans at once route		
Experiment 4 • NeRF • Number of images: 1,931 • Resoulution: 2992×2000 • Three spans at once route		
Experiment 5 • NeRF • Number of images: 1,931 • Resoulution: 2992×2000 • Three spans at once route • Blurred images		
Experiment 6 • NeRF • Number of images: 1,931		

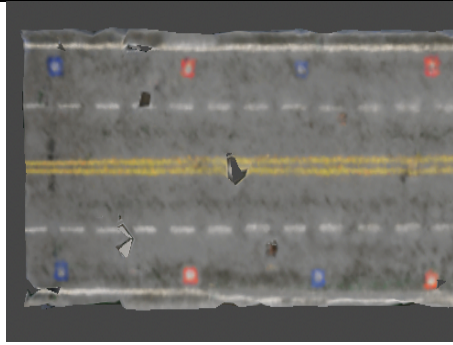
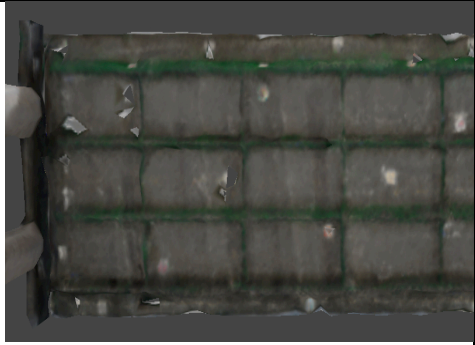
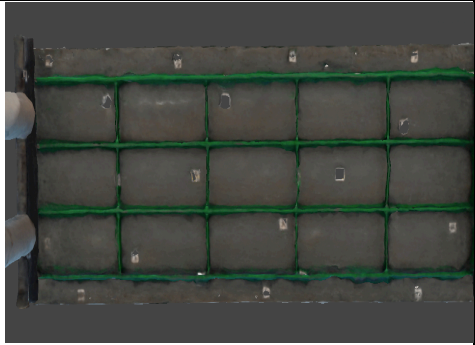
• Resolution: 2992×2000 • Three spans at once route • Deblurred images		
Experiment 7 • NeRF • Number of images: 1,931 • Resolution: 2992×2000 • Each span route • Default hyperparameters		
Experiment 8 • NeRF • Number of images: 1,931 • Resolution: 2992×2000 • Each span route • Hyperparameters changed		

Figure 35. The 3D reconstruction results of each case (Kim and Cha, 2024).

In assessing the effectiveness and accuracy of our NeRF model, we tracked critical performance metrics including MSE Loss, PSNR, and SSIM. These metrics are crucial for evaluating the image quality of our rendered models. Detailed results for each experimental case were compiled and are presented in Table 6. PSNR evaluates the maximum possible signal strength relative to the background noise and is quantified using Equation (20), providing insight into the fidelity of our 3D models. SSIM, which measures perceived image quality based on luminance,

contrast, and structural similarity, is calculated using Equation (21). While each metric individually has certain limitations, together they provide a robust framework for comprehensively assessing the quality of our 3D reconstructions.

$$\text{PSNR} = 10 \log_{10}(R^2 / \text{MSE}), (R=255\text{db}) \quad (20)$$

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \text{ (Wang et al., 2004)} \quad (21)$$

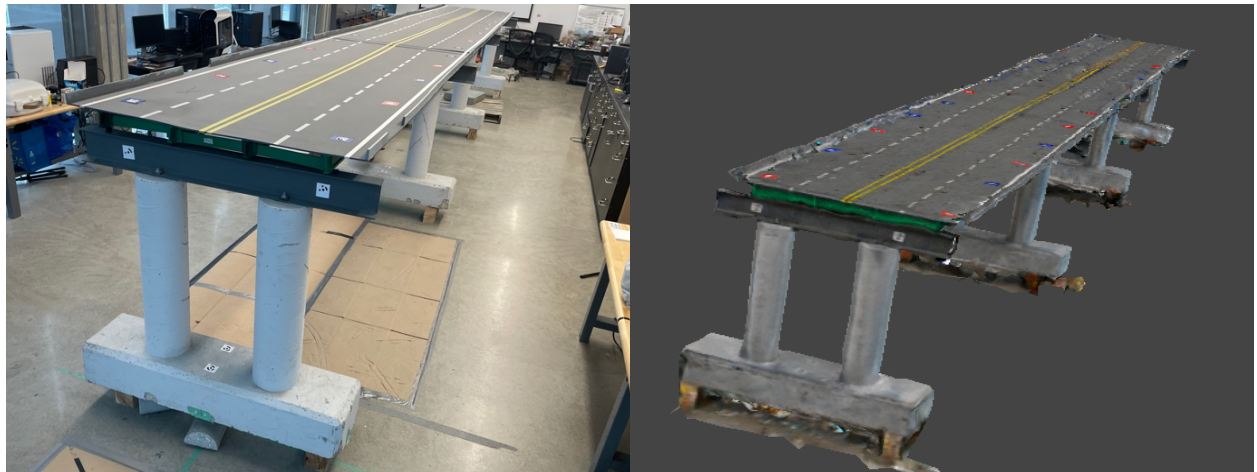
where μ_x indicates the mean of the pixel sample x , μ_y is the mean of pixel sample y , σ_x^2 is the variance of x , σ_y^2 is the variance of y , σ_{xy} is the covariance of x and y , $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$, L indicates the dynamic range of the pixel values ($2^{\text{bits per pixel}} - 1$), $k_1 = 0.01$, and $k_2 = 0.03$.

Tancik et al. (2023) who developed the nerfstudio framework caution that standard evaluation metrics are often based on images captured in the same manner as those used during training. This can lead to misleading results, particularly in practical scenarios where data is captured in unstructured environments or when rendering views that differ significantly from the training set. Solely depending on PSNR and SSIM may not comprehensively reflect the model's performance across varied environments. It is crucial to note that while metrics like PSNR are informative, they do not provide a comprehensive assessment of model performance, especially for views greatly different from the those in the training dataset. A more comprehensive evaluation method, incorporating qualitative analysis through an interactive viewer, can offer a deeper insight into the model's effectiveness under different conditions. Although these limitations exist, quantitative metrics maintain their usefulness for specific facets of our analysis. Notably, in our case studies, experiment 8 achieved the highest PSNR and the lowest MSE, generally indicating

better performance. These findings are summarized in Table 6 and Figure 36 displays the 3D reconstruction of the bridge model from experiment 8 case, showcasing the qualitative improvements.

Table 6. Training performance assessment (Kim and Cha, 2024).

Experiment Case	Image size (pixel)	Number of images	PSNR (db)	SSIM	Loss function (MSE)	Training time (Sec)
Ex.1	2992×2000	1,931	-	-	-	-
Ex.2	1024×720	1,931	21.883	0.7724	0.00813	1925.59
Ex.3	2992×2000	743	21.115	0.7382	0.00911	4071.744
Ex.4	2992×2000	1,931	22.660	0.8167	0.00610	4977.425
Ex.5	2992×2000	1,931	20.720	0.8147	0.00656	4817.612
Ex.6	2992×2000	1,931	23.090	0.8330	0.00643	4802.124
Ex.7	2992×2000	1,931	23.119	0.8029	0.00603	4926.702
Ex.8	2992×2000	1,931	26.570	0.8025	0.00278	13921.733



(a) Image of the bridge model

(b) 3D reconstructed model of the bridge model

Figure 36. Comparison of an image and the 3D reconstruction result of the bridge.

5.4 Damage segmented 3d reconstruction result

A notable feature of our research is depicted in Figure 37, where a segmented crack is clearly visible on the 3D model of the bridge. This image effectively demonstrates the precise localization and scale of damage on the structure. The visibility of the segmented cracks enhances our insight of the damage's location and severity. Additionally, Figure 38 provides a comparative visualization of the bridge model with and without crack segmentation. This side-by-side comparison clearly showcases the enhanced detail and clarity provided by our segmentation process, with the segmented cracks visibly more defined and informative about the structural integrity than the unsegmented version.

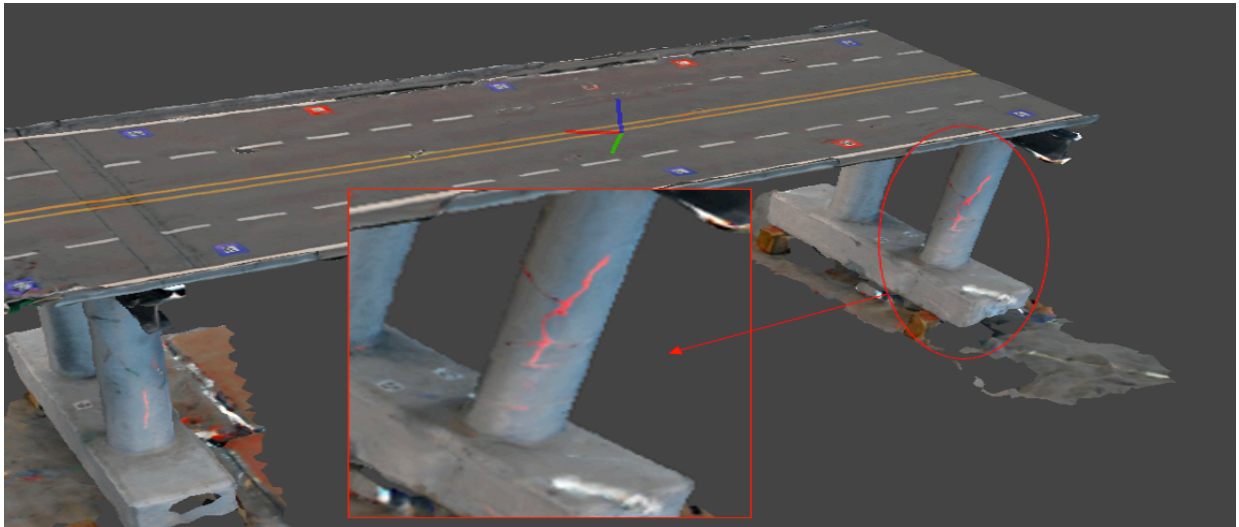
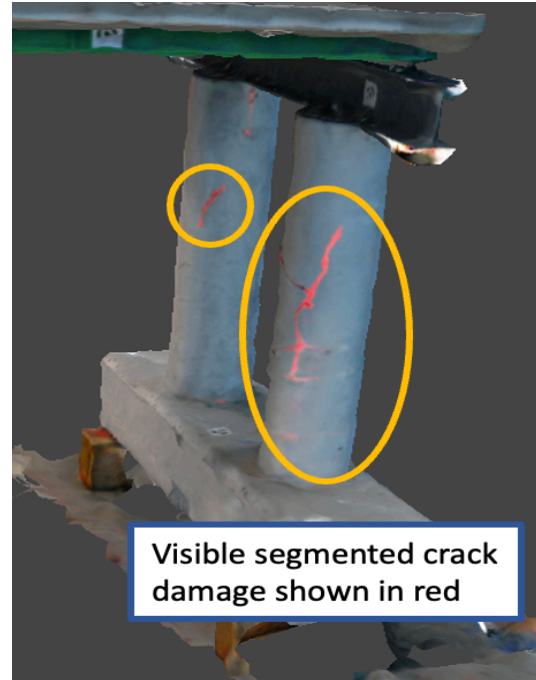


Figure 37. Segmented crack mapped 3d bridge model.



(a)



(b)

Figure 38. Comparative visualization of an unsegmented crack and segmented crack on the 3d model.

6 Conclusion and future work

This chapter concludes the research by summarizing the key findings and discussing the implications of these results. It also outlines potential future work, suggesting directions for further refinement and exploration to extend the capabilities and applications of the methodologies developed in this study.

6.1 Conclusion

This study introduced an novel approach to SHM by incorporating advanced deep learning techniques for damage segmentation and 3D reconstruction methods, with a particular focus on bridge monitoring. Our methodology combined STRNet and TTA to enhance crack detection accuracy, achieving a notable increase in mIoU by 6.6% (from 73.7% without TTA to 80.3% with TTA). Strategic image capture and marker usage significantly improved camera pose estimation, crucial for producing high-quality 3D reconstructions using the NeRF. This approach enabled intuitive and measurable damage mapping on the 3D model, significantly improving the assessment of structural integrity. Although quantitative metrics like PSNR, SSIM, and MSE provided insights, we placed greater emphasis on visual evaluations to confirm enhancements in reconstruction fidelity, due to the challenges in quantitatively assessing the quality of 3D model. The effectiveness of our training approach and the ability to export models in various formats, particularly mesh OBJ files, underscore the practical applications of our work for broader SHM applications.

6.2 Future work

As we move forward, expanding our dataset will be crucial, especially for convolution-based deep learning models that require extensive data to ensure robust outputs across various scenarios.

Additionally, capturing more images to improve color prediction in 3D reconstruction using NeRF is advisable, although hardware limitations restricted our trials to fewer than 2,000 high-resolution images. Further hardware enhancements will allow us to test configurations involving increased numbers of 3D sample points, MLP layers, and dimensions, which could significant potential to enhance the 3D model quality.

Future developments could also focus on refining the NeRF model to better accommodate large structures like bridges and buildings. Adjustments such as optimizing the range of 3D sample points could tailor the model for large-scale applications. Moreover, real-world implementations would need to address the presence of transient objects like humans and vehicles. Developing preprocessing steps to exclude these elements before employing NeRF will be essential for practical and accurate structural assessments.

Future research could also focus on several promising areas to enhance digital twin technology and its applications in SHM. One area involves refining the mesh quality of 3D models through advanced algorithms to ensure higher fidelity and accuracy, incorporating sophisticated surface smoothing techniques and higher resolution data inputs. Another significant advancement could be the integration of digital twin models with VR and AR technologies, providing interactive environments for structural analysis. This would greatly enhance how engineers and maintenance teams visualize and interact with structural data, offering a deeper understanding of structural issues and more effective maintenance planning. Additionally, developing collaborative platforms that allow multiple users to interact with the digital twin models in real-time from remote locations would facilitate a more cohesive approach to infrastructure management. Such platforms could serve as invaluable tools for training, remote diagnostics, and collaborative decision-making in maintenance and emergency response strategies, enabling teams to respond more swiftly and

effectively to structural issues. These advancements will contribute to more proactive maintenance strategies, improving the safety and longevity of critical infrastructure.

7 References

- [1] Abdel-Jaber, H., & Glisic, B. (2015). Analysis of the status of pre-release cracks in prestressed concrete structures using long-gauge sensors. *Smart Materials and Structures*, 24(2), 025038.
- [2] Abdel-Qader, I., Pashaie-Rad, S., Abudayyeh, O., & Yehia, S. (2006). PCA-based algorithm for unsupervised bridge crack detection. *Advances in Engineering Software*, 37(12), 771-778.
- [3] Adhikari, R. S., Moselhi, O., & Bagchi, A. (2014). Image-based retrieval of concrete crack properties for bridge inspection. *Automation in construction*, 39, 180-194.
- [4] Agarwal, S., Snavely, N., Seitz, S. M., & Szeliski, R. (2010). Bundle adjustment in the large. In *Computer Vision—ECCV 2010: 11th European Conference on Computer Vision*, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11 (pp. 29-42). Springer Berlin Heidelberg.
- [5] Agisoft Metashape Professional Edition (version 2.1) (2024). Retrieved from <https://www.agisoft.com>.
- [6] Aharchi, M., & Ait Kbir, M. (2020). A review on 3D reconstruction techniques from 2D images. In *Innovations in Smart Cities Applications Edition 3: The Proceedings of the 4th International Conference on Smart City Applications 4* (pp. 510-522). Springer International Publishing.
- [7] Al-Mahaidi, R., & Kalfat, R. (2018). *Rehabilitation of concrete structures with fiber-reinforced polymer*. Butterworth-Heinemann.

- [8] Ali, R., Kang, D., Suh, G., & Cha, Y. J. (2021). Real-time multiple damage mapping using autonomous UAV and deep faster region-based neural networks for GPS-denied structures. *Automation in Construction*, 130, 103831.
- [9] Angjeliu, G., Coronelli, D., & Cardani, G. (2020). Development of the simulation model for Digital Twin applications in historical masonry buildings: The integration between numerical and experimental reality. *Computers & Structures*, 238, 106282.
- [10] Avenash, R., & Viswanath, P. (2019, February). Semantic Segmentation of Satellite Images using a Modified CNN with Hard-Swish Activation Function. In *VISIGRAPP (4: VISAPP)* (pp. 413-420).
- [11] Barron, J. T., Mildenhall, B., Verbin, D., Srinivasan, P. P., & Hedman, P. (2022). Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5470-5479).
- [12] Boyes, H., & Watson, T. (2022). Digital twins: An analysis framework and open issues. *Computers in Industry*, 143, 103763.
- [13] Brownjohn, J. M. W., Xu, Y., & Hester, D. (2017). Vision-based bridge deformation monitoring. *Frontiers in Built Environment*, 3, 23.
- [14] Brunetaud, X., Luca, L. D., Janvier-Badosa, S., Beck, K., & Al-Mukhtar, M. (2012). Application of digital techniques in monument preservation. *European Journal of Environmental and Civil Engineering*, 16(5), 543-556.
- [15] Cha, Y. J., & Buyukozturk, O. (2015). Structural damage detection using modal strain energy and hybrid multiobjective optimization. *Computer-Aided Civil and Infrastructure Engineering*, 30(5), 347-358.

- [16] Cha, Y. J., Ali, R., Lewis, J., & Büyüköztürk, O. (2024). Deep learning-based structural health monitoring. *Automation in Construction*, 161, 105328.
- [17] Cha, Y. J., Chen, J. G., & Büyüköztürk, O. (2017). Output-only computer vision based damage detection using phase-based optical flow and unscented Kalman filters. *Engineering Structures*, 132, 300-313.
- [18] Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5), 361-378.
- [19] Cha, Y. J., Kim, Y., & You, T. (2018). Advanced sensing and structural health monitoring. *Journal of Sensors*, 2018.
- [20] Chaibyasarn, K., Buatik, A., Mohamad, H., Zhou, M., Kongsilp, S., & Poovarodom, N. (2022). Integrated pixel-level CNN-FCN crack detection via photogrammetric 3D texture mapping of concrete structures. *Automation in Construction*, 140, 104388.
- [21] Chen, C., Fu, H., Zheng, Y., Tao, F., & Liu, Y. (2023). The advance of digital twin for predictive maintenance: The role and function of machine learning. *Journal of Manufacturing Systems*, 71, 581-594.
- [22] Chen, L., Chu, X., Zhang, X., & Sun, J. (2022, October). Simple baselines for image restoration. In *European Conference on Computer Vision* (pp. 17-33). Cham: Springer Nature Switzerland.
- [23] Chen, S., Laefer, D. F., Mangina, E., Zolanvari, S. I., & Byrne, J. (2019). UAV bridge inspection through evaluated 3D reconstructions. *Journal of Bridge Engineering*, 24(4), 05019001.

- [24] Cheng, Y., Zhang, J., & Wu, J. (2019). Fragility analysis of a self-anchored suspension bridge based on structural health monitoring data. *Advances in Civil Engineering*, 2019, 1-19.
- [25] Cheng, Z., Li, H., Asano, Y., Zheng, Y., & Sato, I. (2021). Multi-view 3d reconstruction of a texture-less smooth surface of unknown generic reflectance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 16226-16235).
- [26] Choi, H., Ham, Y., & Popovics, J. S. (2016). Integrated visualization for reinforced concrete using ultrasonic tomography and image-based 3-D reconstruction. *Construction and building materials*, 123, 384-393.
- [27] Choi, W., & Cha, Y. J. (2019). SDDNet: Real-time crack segmentation. *IEEE Transactions on Industrial Electronics*, 67(9), 8016-8025.
- [28] Croce, V., Billi, D., Caroti, G., Piemonte, A., De Luca, L., & Véron, P. (2023). Comparative Assessment of NeRF and Photogrammetry in Digital Heritage: Impact of Varying Image Conditions on 3D Reconstruction.
- [29] Dandois, J. P., Olano, M., & Ellis, E. C. (2015). Optimal altitude, overlap, and weather conditions for computer vision UAV estimates of forest structure. *Remote sensing*, 7(10), 13895-13920.
- [30] Deng, J., Singh, A., Zhou, Y., Lu, Y., & Lee, V. C. S. (2022). Review on computer vision-based crack detection and quantification methodologies for civil structures. *Construction and Building Materials*, 356, 129238.

- [31] Deraemaeker, A., Preumont, A., Reynders, E., De Roeck, G., Kullaa, J., Lamsa, V., ... & Wandowski, T. (2010). Vibration-based structural health monitoring using large sensor networks. *Smart Structures and Systems*, 6(3), 335-347.
- [32] Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, 186, 1031-1045.
- [33] Dryden, N., Maruyama, N., Benson, T., Moon, T., Snir, M., & Van Essen, B. (2019, May). Improving strong-scaling of CNN training by exploiting finer-grained parallelism. In *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (pp. 210-220). IEEE.
- [34] Dung, C. V. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, 99, 52-58.
- [35] Durou, J. D., Falcone, M., Quéau, Y., & Tozza, S. (2020). A comprehensive introduction to photometric 3d-reconstruction. *Advances in Photometric 3D-Reconstruction*, 1-29.
- [36] Farrar, C. R., & Worden, K. (2007). An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851), 303-315.
- [37] Feng, D., & Feng, M. Q. (2018). Computer vision for SHM of civil infrastructure: From dynamic response measurement to damage detection—A review. *Engineering Structures*, 156, 105-117.
- [38] Feng, D., & Feng, M. Q. (2021). *Computer vision for structural dynamics and health monitoring*. John Wiley & Sons.

- [39] Fu, H., Meng, D., Li, W., & Wang, Y. (2021). Bridge crack semantic segmentation based on improved Deeplabv3+. *Journal of Marine Science and Engineering*, 9(6), 671.
- [40] Fujita, Y., & Hamamoto, Y. (2011). A robust automatic crack detection method from noisy concrete surfaces. *Machine Vision and Applications*, 22, 245-254.
- [41] Funari, M. F., Hajjat, A. E., Masciotta, M. G., Oliveira, D. V., & Lourenço, P. B. (2021). A parametric scan-to-FEM framework for the digital twin generation of historic masonry structures. *Sustainability*, 13(19), 11088.
- [42] Gao, K., Gao, Y., He, H., Lu, D., Xu, L., & Li, J. (2022). Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*.
- [43] Garbin, S. J., Kowalski, M., Johnson, M., Shotton, J., & Valentin, J. (2021). Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 14346-14355).
- [44] Gaudreault, V., & Lemire, P. (2006). *The age of public infrastructure in Canada*. Ottawa, ON: Statistics Canada.
- [45] Giurgiutiu, V. (2007). *Structural health monitoring: with piezoelectric wafer active sensors*. Elsevier.
- [46] Giurgiutiu, V. (2015). *Structural health monitoring of aerospace composites*.
- [47] Grieves, M., & Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. *Transdisciplinary perspectives on complex systems: New findings and approaches*, 85-113.
- [48] Hammad, A. W., Yan, J., & Mostofi, B. (2007). Recent development of bridge management systems in Canada. In *2007 Annual Conference and Exhibition of the*

Transportation Association of Canada: Transportation-An Economic Enabler (Les Transports: Un Levier Economique) Transportation Association of Canada (TAC).

- [49] Hartley, R., & Zisserman, A. (2003). Multiple view geometry in computer vision. Cambridge university press.
- [50] Hata, K., & Savarese, S. (2017). Cs231a course notes 3: Epipolar geometry. 18c.)/K. Hata, S. Savarese.–2018–режим доступа: http://web.stanford.edu/class/cs231a/course_notes/03-epipolar-geometry.pdf.
- [51] Hoshyarmanesh, H., Abbasi, A., Moein, P., Ghodsi, M., & Zareinia, K. (2017). Design and implementation of an accurate, portable, and time-efficient impedance-based transceiver for structural health monitoring. IEEE/ASME Transactions on Mechatronics, 22(6), 2809-2814.
- [52] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). pmlr.
- [53] Jain, A., Tancik, M., & Abbeel, P. (2021). Putting nerf on a diet: Semantically consistent few-shot view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5885-5894).
- [54] Jenkins, M. D., Carr, T. A., Iglesias, M. I., Buggy, T., & Morison, G. (2018, September). A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks. In 2018 26th European signal processing conference (EUSIPCO) (pp. 2120-2124). IEEE.
- [55] Jiang, F., Ding, Y., Song, Y., Geng, F., & Wang, Z. (2021). Digital Twin-driven framework for fatigue life prediction of steel bridges using a probabilistic multiscale

model: Application to segmental orthotropic steel deck specimen. *Engineering Structures*, 241, 112461.

- [56] Jog, G. M., Fathi, H., & Brilakis, I. (2011). Automated computation of the fundamental matrix for vision based construction site applications. *Advanced Engineering Informatics*, 25(4), 725-735.
- [57] Kajiya, J. T., & Von Herzen, B. P. (1984). Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3), 165-174.
- [58] Kang, D. H., & Cha, Y. J. (2022). Efficient attention-based deep encoder and decoder for automatic crack segmentation. *Structural Health Monitoring*, 21(5), 2190-2205.
- [59] Karaaslan, E., Bagci, U., & Catbas, F. N. (2019). Artificial intelligence assisted infrastructure assessment using mixed reality systems. *Transportation Research Record*, 2673(12), 413-424.
- [60] Karami, A., Menna, F., & Remondino, F. (2022). Combining Photogrammetry and Photometric Stereo to Achieve Precise and Complete 3D Reconstruction. *Sensors*, 22(21), 8172.
- [61] Khaloo, A., & Lattanzi, D. (2017). Hierarchical dense structure-from-motion reconstructions for infrastructure condition assessment. *Journal of Computing in Civil Engineering*, 31(1), 04016047.
- [62] Kheradmandi, N., & Mehranfar, V. (2022). A critical review and comparative study on image segmentation-based techniques for pavement crack detection. *Construction and Building Materials*, 321, 126162.

- [63] Kim, G., and Cha, Y.J. (2024). Neural radiance fields (NeRF-based) 3D damage mapping with advanced deep learning damage segmentation. *Mechanical Systems and Signal Processing*, submitted.
- [64] Kimura, M. (2021, December). Understanding test-time augmentation. In *International Conference on Neural Information Processing* (pp. 558-569). Cham: Springer International Publishing.
- [65] Kong, X., & Hucks, R. G. (2023). Preserving our heritage: A photogrammetry-based digital twin framework for monitoring deteriorations of historic structures. *Automation in Construction*, 152, 104928.
- [66] Lin, H. H., Liu, T. L., & Chuang, J. H. (2002, September). A probabilistic SVM approach for background scene initialization. In *Proceedings. International conference on image processing* (Vol. 3, pp. 893-896). IEEE.
- [67] Lin, K., Xu, Y. L., Lu, X., Guan, Z., & Li, J. (2021). Digital twin-based collapse fragility assessment of a long-span cable-stayed bridge under strong earthquakes. *Automation in Construction*, 123, 103547.
- [68] Liu, L., Gu, J., Zaw Lin, K., Chua, T. S., & Theobalt, C. (2020). Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33, 15651-15663.
- [69] Liu, Y. F., Cho, S., Spencer Jr, B. F., & Fan, J. S. (2016). Concrete crack assessment using digital image processing and 3D scene reconstruction. *Journal of Computing in Civil Engineering*, 30(1), 04014124.
- [70] Liu, Z., Cao, Y., Wang, Y., & Wang, W. (2019). Computer vision-based concrete crack detection using U-net fully convolutional networks. *Automation in Construction*, 104, 129-139

- [71] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 91-110.
- [72] Ma, Z., & Liu, S. (2018). A review of 3D reconstruction techniques in civil engineering and their applications. *Advanced Engineering Informatics*, 37, 163-174.
- [73] Majidi, S., Omidalizarandi, M., & Sharifi, M. A. (2023). Intelligent 3d Crack Reconstruction Using Close Range Photogrammetry Imagery. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10, 443-450.
- [74] Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., & Duckworth, D. (2021). Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7210-7219).
- [75] Masciotta, M. G., Sanchez-Aparicio, L. J., Oliveira, D. V., & Gonzalez-Aguilera, D. (2023). Integration of laser scanning technologies and 360° photography for the digital documentation and management of cultural heritage buildings. *International Journal of Architectural Heritage*, 17(1), 56-75.
- [76] Max, N. (1995). Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2), 99-108.
- [77] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1), 99-106.
- [78] Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4), 1-15.

- [79] Oztemel, E., & Gursev, S. (2020). Literature review of Industry 4.0 and related technologies. *Journal of intelligent manufacturing*, 31(1), 127-182.
- [80] Qing, X. P., Beard, S. J., Kumar, A., Sullivan, K., Aguilar, R., Merchant, M., & Taniguchi, M. (2008). The performance of a piezoelectric-sensor-based SHM system under a combined cryogenic temperature and vibration environment. *Smart Materials and Structures*, 17(5), 055010.
- [81] Rashidi, M., Mohammadi, M., Sadeghlou Kivi, S., Abdolvand, M. M., Truong-Hong, L., & Samali, B. (2020). A decade of modern bridge monitoring using terrestrial laser scanning: Review and future directions. *Remote Sensing*, 12(22), 3796.
- [82] Remondino, F., Barazzetti, L., Nex, F., Scaioni, M., & Sarazzi, D. (2012). UAV photogrammetry for mapping and 3d modeling—current status and future perspectives. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38, 25-31.
- [83] Remondino, F., Karami, A., Yan, Z., Mazzacca, G., Rigon, S., & Qin, R. (2023). A critical analysis of NeRF-based 3d reconstruction. *Remote Sensing*, 15(14), 3585.
- [84] Ribeiro, D., Calçada, R., Ferreira, J., & Martins, T. (2014). Non-contact measurement of the dynamic displacement of railway bridges using an advanced video-based system. *Engineering Structures*, 75, 164-180.
- [85] Rong, W., Li, Z., Zhang, W., & Sun, L. (2014, August). An improved CANNY edge detection algorithm. In *2014 IEEE international conference on mechatronics and automation* (pp. 577-582). IEEE.
- [86] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted*

Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

- [87] Sanchez-Silva, M., Klutke, G. A., & Rosowsky, D. V. (2011). Life-cycle performance of structures subject to multiple deterioration mechanisms. *Structural Safety*, 33(3), 206-217.
- [88] Schenk, T., Seo, S., & Csathó, B. (2001). Accuracy study of airborne laser scanning data with photogrammetry. *International Archives of Photogrammetry and Remote Sensing*, 34(part 3), W4.
- [89] Schonberger, J. L., & Frahm, J. M. (2016). Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4104-4113).
- [90] Seok Lee, H., & Mu Lee, K. (2013). Dense 3d reconstruction from severely blurred images using a single moving camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 273-280).
- [91] Sinha, S. K., & Fieguth, P. W. (2006). Automated detection of cracks in buried concrete pipe images. *Automation in construction*, 15(1), 58-72.
- [92] Smith, P., & Ricanek, K. (2020). Mitigating algorithmic bias: Evolving an augmentation policy that is non-biasing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops* (pp. 90-97).
- [93] Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Wang, T., ... & Kanazawa, A. (2023, July). Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings* (pp. 1-12).

- [94] Tao, F., Xiao, B., Qi, Q., Cheng, J., & Ji, P. (2022). Digital twin modeling. *Journal of Manufacturing Systems*, 64, 372-389.
- [95] Teng, S., Chen, X., Chen, G., & Cheng, L. (2023). Structural damage detection based on transfer learning strategy using digital twins of bridges. *Mechanical Systems and Signal Processing*, 191, 110160.
- [96] Tennyson, R. C., Mufti, A. A., Rizkalla, S., Tadros, G., & Benmokrane, B. (2001). Structural health monitoring of innovative bridges in Canada with fiber optic sensors. *Smart materials and Structures*, 10(3), 560.
- [97] Torok, M. M., Fard, M. G., & Kochersberger, K. B. (2012). Post-disaster robotic building assessment: Automated 3D crack detection from image-based reconstructions. In *Computing in Civil Engineering (2012)* (pp. 397-404).
- [98] Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J. T., & Srinivasan, P. (2022, June). Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5481-5490). IEEE.
- [99] Wang, G., Li, W., Aertsen, M., Deprest, J., Ourselin, S., & Vercauteren, T. (2019). Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338, 34-45.
- [100] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4), 600-612.

- [101] Xiang, C., Gan, V. J., Guo, J., & Deng, L. (2023). Semi-supervised learning framework for crack segmentation based on contrastive learning and cross pseudo supervision. *Measurement*, 217, 113091.
- [102] Xu, J., Shu, X., Qiao, P., Li, S., & Xu, J. (2023). Developing a digital twin model for monitoring building structural health by combining a building information model and a real-scene 3D model. *Measurement*, 217, 112955.
- [103] Xu, Y., Bao, Y., Chen, J., Zuo, W., & Li, H. (2019). Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images. *Structural Health Monitoring*, 18(3), 653-674.
- [104] Xue, Y., Shi, P., Jia, F., & Huang, H. (2022). 3D reconstruction and automatic leakage defect quantification of metro tunnel based on SfM-Deep learning method. *Underground Space*, 7(3), 311-323.
- [105] Yamaguchi, T., & Hashimoto, S. (2010). Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Machine Vision and Applications*, 21, 797-809.
- [106] Yang, M. D., Chao, C. F., Huang, K. S., Lu, L. Y., & Chen, Y. P. (2013). Image-based 3D scene reconstruction and exploration in augmented reality. *Automation in Construction*, 33, 48-60.
- [107] Ye, C., Butler, L., Calka, B., Iangurazov, M., Lu, Q., Gregory, A., ... & Middleton, C. (2019). A digital twin of bridges for structural health monitoring.
- [108] Ye, X. W., Jin, T., & Yun, C. B. (2019). A review on deep learning-based structural health monitoring of civil infrastructures. *Smart Struct. Syst*, 24(5), 567-585.

- [109] Yeum, C. M., Choi, J., & Dyke, S. J. (2017). Autonomous image localization for visual inspection of civil infrastructure. *Smart Materials and Structures*, 26(3), 035051.
- [110] Yu, S. N., Jang, J. H., & Han, C. S. (2007). Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. *Automation in construction*, 16(3), 255-261.
- [111] Zakeri, H., Nejad, F. M., & Fahimifar, A. (2017). Image based techniques for crack detection, classification and quantification in asphalt pavement: a review. *Archives of Computational Methods in Engineering*, 24, 935-977.
- [112] Zhang, G. Q., Xu, Y. L., Dan, D. H., Jiang, S. J., & Zhu, Q. (2024). Simulation and prediction of vortex-induced vibration of a long suspension bridge using SHM-based digital twin technology. *Journal of Wind Engineering and Industrial Aerodynamics*, 247, 105705.
- [113] Zhang, K., Riegler, G., Snavely, N., & Koltun, V. (2020). Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*.
- [114] Zhao, S., Kang, F., & Li, J. (2022). Concrete dam damage detection and localisation based on YOLOv5s-HSC and photogrammetric 3D reconstruction. *Automation in Construction*, 143, 104555.
- [115] Zhao, S., Kang, F., Li, J., & Ma, C. (2021). Structural health monitoring and inspection of dams based on UAV photogrammetry with image 3D reconstruction. *Automation in Construction*, 130, 103832.
- [116] Zhou, H. F., Zheng, J. F., Xie, Z. L., Lu, L. J., Ni, Y. Q., & Ko, J. M. (2017). Temperature effects on vision measurement system in long-term continuous monitoring of displacement. *Renewable energy*, 114, 968-983.

- [117] Zollmann, S., Hoppe, C., Kluckner, S., Poglitsch, C., Bischof, H., & Reitmayr, G.
(2014). Augmented reality for construction site monitoring and documentation.
Proceedings of the IEEE, 102(2), 137-154.