Contents lists available at ScienceDirect

Engineering Structures

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

Automated bridge component recognition using close-range images from unmanned aerial vehicles

Hyunjun Kim [a], Yasutaka Narazaki [b], Billie F. Spencer Jr. [c,*]

[a] *Department of Civil Engineering, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea*
[b] *Zhejiang University-University of Illinois at Urbana-Champaign Institute, Zhejiang University, Haining 314400, China*
[c] *Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

ARTICLE INFO

Keywords:
3D semantic segmentation
Automated structural inspection
Bridge components
Close-range images
Computer vision
Point cloud
Unmanned aerial vehicle (UAV)

ABSTRACT

Unmanned aerial vehicles (UAVs), in conjunction with computer vision techniques, have shown great potential for bridge inspections. Close-range images captured in proximity to the structural surface are generally required to detect damage and also need to be linked to the corresponding structural component to enable assessment of the health of the global structure. However, the lack of contextual information makes automated identification of bridge components in close-range images challenging. This study proposes a framework for automated bridge component recognition based on close-range images collected by UAVs. First, a 3D point cloud is generated from the UAV survey of the bridge and segmented into bridge components. The segmented point cloud is subsequently projected onto the camera coordinates to categorize each of the images into the bridge component. The proposed approach is successfully validated using a local highway bridge, pointing the way for improved inspection of full-scale bridges.

1. Introduction

Bridges, one of the important types of civil engineering structures, are exposed to a wide variety of loads during their lifetime, including self-weight, wind, seismic, and traffic forces. According to the 2021 America's Infrastructure Report Card published by the American Society of Civil Engineers (ASCE) [1], 42 % of all bridges in the United States are at least 50 years old. In addition, 7.5 % (~46,000) of the nation's bridges are structurally deficient. To ensure the safety and functionality of these bridges, the Federal Highway Administration (FHWA) mandates a detailed inspection at least every-two years [2]. Typically, these inspections are conducted visually, identifying surface damage (e.g., corrosion, cracking, and spalling); however, such manual inspections are intrinsically costly, time-consuming, dangerous, and potentially subjective/inaccurate.

Advances in unmanned aerial vehicle (UAV) technologies offer the potential for improved inspection based on their automated imaging capabilities [3–7] and the associated image processing techniques [8,9]. Additionally, the application of UAVs can provide a set of images taken of hard-to-reach areas on a full-scale bridge. However, manual review of the enormous amount of imagery generated by a UAV survey is laborious and error prone. For example, a major crack in a steel beam of the Interstate 40 bridge over the Mississippi River was captured in a 2019 UAV survey but it was not noticed by engineers [10]. A subsequent UAV survey was conducted in 2021, after which review of the collected imagery identified the crack, and the bridge was immediately closed for repair. Had a reliable approach for processing imagery automatically been available, then this crack could have been identified two years earlier.

Convolutional neural networks (CNNs) [11] have been considered as an innovative tool that can facilitate the automated detection of damage from visual imagery. In these approaches, a number of images including damage are collected from the structural surface to train the classification model, from which the obtained model can be employed to automatically identify damage in new images [12–15]. Region-based CNNs, including R-CNN [16], Fast R-CNN [17], and Faster R-CNN [18], are specifically designed to produce a set of bounding boxes that include a single object with the corresponding class; its applications have been presented for multiple types of damage in each image [19,20]. Furthermore, semantic segmentation algorithms (e.g., Fully Convolutional Network (FCN) [21] and Mask R-CNN [22]), optimized to identify a group of pixels with the associated class, also have been utilized successfully for detection of multiple types of damage [23–25]. While these studies have enabled the automated detection and classification of local

[*] Corresponding author.
E-mail address: bfs@illinois.edu (B.F. Spencer Jr.).

https://doi.org/10.1016/j.engstruct.2022.115184
Received 2 April 2022; Received in revised form 8 August 2022; Accepted 22 October 2022
Available online 2 November 2022

Table 1
Selected existing literature about bridge component recognition using 2D images.

Literature	Dataset	Bridge component type	Algorithms used
Narazaki et al. (2020) [26]	1563 images (up to 320 × 320)	Column, beam/slab	FCN [21], SegNet [33]
Saovana et al. (2020) [27]	4,200 images (512 × 384)	Road, pier	U-Net [34]
Karim et al. (2021) [28]	4440 images	Slab, pier cap, pier, bearing, etc.	Mask R-CNN [22]
Narazaki et al. (2021) [29]	8,648 images (640 × 360)	Slab, beam, column	FCN [21]
Park et al. (2021) [30]	245 images (800 × 600)	Girder, pier	DeepLabv3+ [35]
Sajedi and Liang (2021) [31]	236 images (430 × 400)	Superstructure, column, cap beam, abutment, foundation	FC-DenseNet [36]
Saovana et al. (2021) [32]	1100 images (512 × 288)	Road, pier	YOLACT [37]
	1,670 images (512 × 384)	Deck, girder, abutment	

Table 2
Selected existing literature about bridge component recognition using 3D point clouds.

Literature	Dataset	Bridge component type	Algorithms used
Riveiro et al. (2016) [38]	Point clouds[a]	Wall, pier, pathway	Geometric features
Lu and Brilakis (2019) [39]		Slab, girder, pier cap, pier	
Lu et al. (2019) [40]		Slab, girder, pier cap, pier	
Yan and Hajjar (2021) [44]		Deck, girder, pier, abutment	
Truong-Hong and Lindenbergh (2022) [45]		Superstructure and substructure	
Kim and Kim (2020) [41]	Three bridges (4,096 input points)	Slab, girder, pier, abutment	PointNet [46], PointCNN [47], DGCNN [48]
Kim et al. (2020) [42]	Seven bridges (2,048 input points)	Deck, pier	PointNet [46]
Lee et al. (2021) [43]	Six bridges (4,096 input points)	Deck, pier, abutment	PointNet [46], DGCNN [48]

[a] Training process is not required.

damage, the identification performance can be degraded by the presence of structural component damage types that were not included in the training. Therefore, knowing the component information in advance can be useful to determine appropriate classifiers for accurate damage identification. Moreover, to assess the effect of local damage to the global health of the structure, the component associated with this damage needs to be determined.

A variety of semantic segmentation algorithms have applied to a set of 2D images for the classification of bridge components [26–32]. In general, wide-angle images including multiple bridge components are preferred for this purpose, because distinctive features (e.g., surface texture, color gradient, and pattern) can be distinguished in each image. In addition, the boundaries between the bridge components can be also used as key indicators for segmentation purposes. The segmentation model constructed by utilizing the obtained features is then applied to new images, collecting a group of pixels with the associated class (e.g., slab, girder, and pier). For reference, a summary of the state-of-the-art methods for automated bridge component recognition based on 2D images is provided in Table 1. As such, the segmentation model based on 2D images has shown the potential for automated bridge component recognition; however, the classification of close-range images captured from the bridge components that have a similar surface texture and pattern (e.g., concrete) is challenging in practice, because such features are limited to 2D space.

The semantic segmentation of bridge components from 3D point clouds obtained from laser scans has been reported by a number of authors [38–45]. In contrast to 2D images, point clouds with an extra dimension provide the geometric and topological features of bridge components in the physical coordinate system. Although such features can be utilized to automatically categorize each point into the bridge component without additional training, its application is limited because of several constraints and the corresponding thresholds (e.g., voxel size, line fitting, and thickness ratio) [38–40,44,45]. Thus, semantic segmentation algorithms optimized for point clouds are considered as a promising tool for automated bridge component recognition [41–43]. Here, as the laser scanner is designed to efficiently provide a set of point clouds, such point cloud data in conjunction with the semantic segmentation network have shown a great potential. Table 2 represents a summary of state-of-the-art methods for automated bridge component recognition based on 3D point clouds. Although point clouds in the physical coordinate system are useful for damage localization and quantification [49], close-range images captured in the proximity to the structural surface are typically required for accurate damage identification.

The application of structure from motion (SfM) to UAV imagery can generate the point cloud, achieving the advantages of both 2D images and 3D points [50,51]. For example, Perry et al. (2020) [51] utilized a set of images taken by the UAV to construct a point cloud of the bridge, from which bridge components were identified in a human-in-the-loop manner. However, automated bridge component recognition based on close-range images that are adequate for accurate damage identification still remains a challenge. For example, as shown in Fig. 1, while multiple bridge components are labeled by different colors in wide-angle images, the class of the bridge component in the close-range image captured in proximity to the structural surface is difficult to identify. Note that Fig. 1 (b) is a part of the pier cap in Fig. 1(a), which can provide a high pixel density for accurate damage identification. Thus, the segmentation model that reflects the distinctive 3D features obtained from laser scans of full-scale bridges can be employed for the point cloud generated by close-range images; however, the use of two different data obtained from the laser scanner and UAV camera results in several challenges, including: (1) the performance of semantic segmentation is degraded, because the orientation of point clouds measured by the laser scanner and UAV can be very different; (2) as the segmentation model is generated based on the laser scanner, its identification accuracy for the point cloud resulting from SfM can be decreased by different measurement noise mechanisms and point cloud density; and (3) the 3D semantic segmentation results do not directly provide the information about bridge components captured in each close-range image. To realize the full potential of UAV surveys for bridge inspection, such challenges must be overcome.

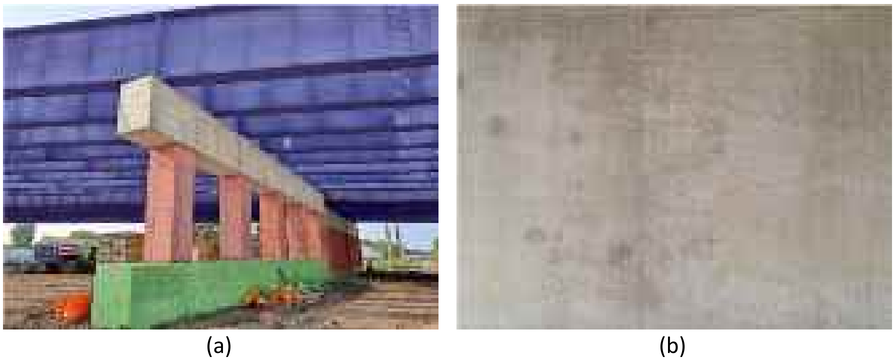


Fig. 1. Typical bridge component images: (a) wide-angle image including multiple bridge components and (b) close-range image of the pier cap.

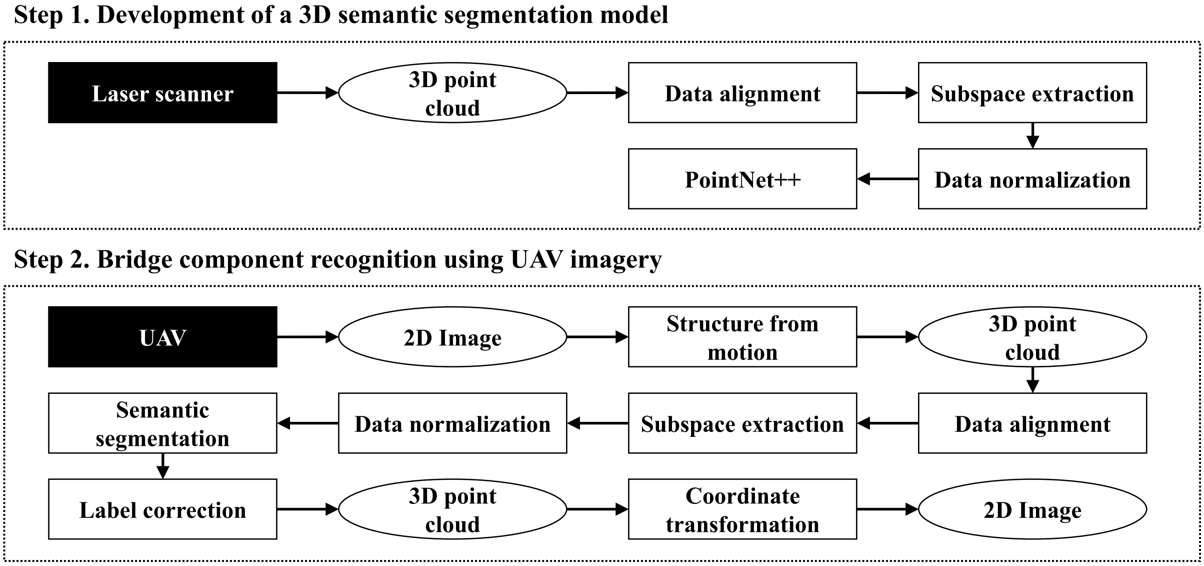


Fig. 2. Overall framework for automated bridge component recognition.

This paper presents a framework for automated bridge component recognition based on close-range images acquired by a UAV. The proposed approach consists of two main steps: (1) development of a 3D semantic segmentation model from a set of point clouds collected by a laser scanner, and (2) segmentation of a point cloud generated by UAV imagery that is projected onto each camera coordinate system to identify the bridge component in the corresponding close-range image. The main idea of the proposed approach is to build the segmentation model that reflects the distinctive 3D features derived from laser scans of full-scale bridges. The obtained model is further employed to categorize a set of close-range images taken by the UAV into the bridge components based on a coordinate transformation. As 2D close-range images have insufficient contextual information, the application of the 3D point cloud generated by UAV imagery and SfM can be useful for automated bridge component recognition. The proposed approach is then field validated using a highway bridge in Illinois, USA. The resulting information regarding the bridge components, expressed by both image and point cloud, can be utilized for the evaluation of the global status of the full-scale bridge.

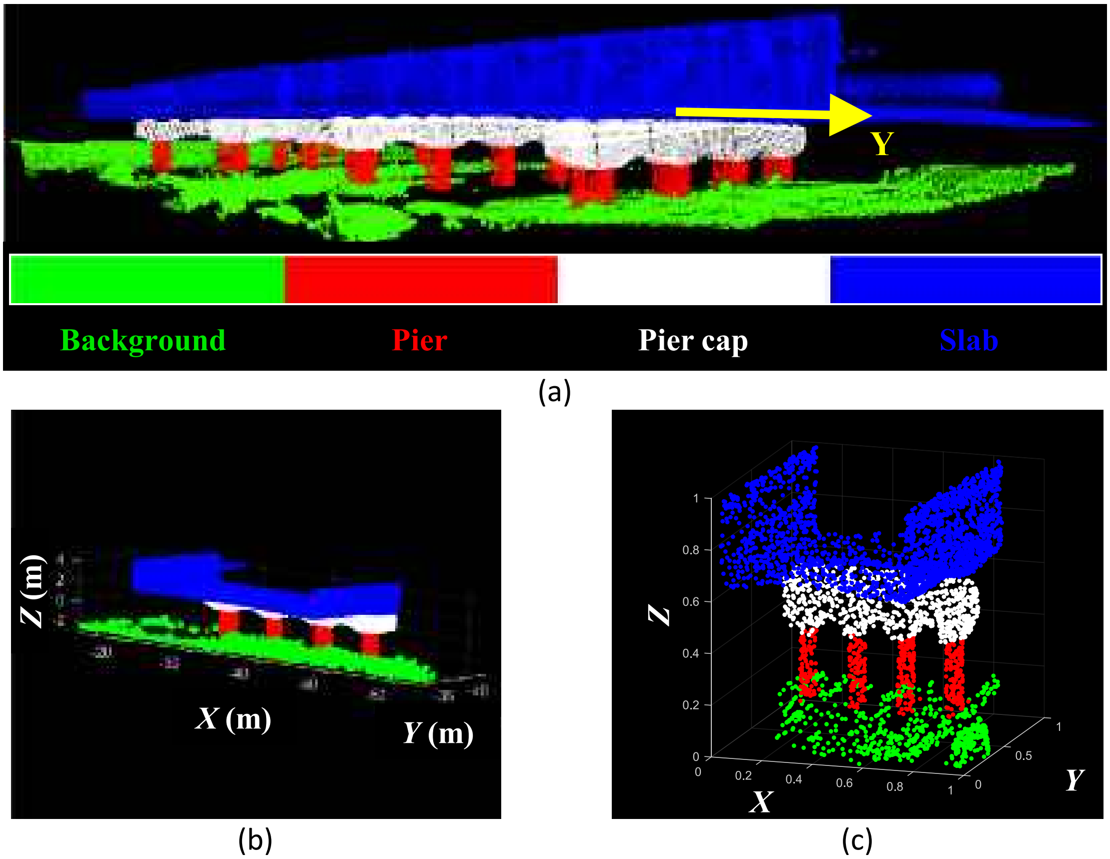
2. Automated bridge component recognition using close-range images collected by a UAV

The previous studies for automated bridge component recognition have focused on the use of wide-angle images including several bridge components [26–32]; however, the proposed approach is designed to use close-range images of a single bridge component, which are typically required for accurate damage identification. This section describes the generation of a segmentation model based on the laser scanner data; subsequently its application to point cloud data generated by UAV imagery is presented. Fig. 2 shows the entire process of the proposed framework for automated bridge component recognition based on a combination of the laser scanner data and UAV imagery.

2.1. Development of a 3D semantic segmentation model

A set of point clouds collected by a laser scanner is used to build the segmentation model that classifies each 3D point into an appropriate bridge component category. The previous studies utilized the obtained model for classifying point clouds measured by the laser scanner [38–45], whereas the proposed approach is focused on the application to data generated by UAV imagery. However, the segmentation model trained using the laser scanner data cannot be applied directly to the point cloud obtained from UAV imagery, because such point cloud data is expressed in very different coordinate systems, especially in terms of the rotation. Thus, all the point clouds in this study are aligned to minimize the effect of similarity transformations. The entire process for generating the segmentation model can be summarized in the following three steps: (1) data alignment, (2) subspace extraction, and (3) data normalization.

The point clouds that include the essential bridge components (i.e., pier, pier cap, and slab) are aligned to generate a robust segmentation



(a)

(b)

(c)

Fig. 3. Data processing of point cloud taken by laser scans: (a) data alignment, (b) subspace extraction, and (c) data normalization.

model. Here, because the foundations in the point clouds collected by a laser scanner are covered by background (see Fig. 3(a)), such foundation under the bridge piers is defined as background in this study. Based on the coordinate system of the laser scanner, a span of the bridge is aligned with the *y*-axis in this study. The entire point cloud is then divided into multiple subspaces with a fixed size along the bridge span, from which the subspaces including bridge piers are retained only for the subsequent process, as shown in Fig. 3(b). Note that as the pier is an essential part of the bridge, the subspaces with the corresponding points are employed for constructing the segmentation model. Based on the size of the input for the semantic segmentation algorithm, a constant number of points is randomly sampled in each subspace. Those points are normalized to minimize the effect of scale, as shown in Fig. 3(c). The resulting point clouds are utilized to train the segmentation model.

PointNet++ [52], an extension of PointNet [46], is selected in this study to perform the semantic segmentation of point clouds. PointNet takes 3D point clouds as the input, and then applies a shared multi-layer perceptron repeatedly with different layer sizes. This process yields a 1024-dimensional feature vector for every point. A global feature vector is subsequently obtained by applying a max-pooling layer to the stack of point feature vectors, which is employed for classification tasks. Furthermore, the global feature is concatenated with 64-dimensional point feature vectors to perform semantic segmentation. Although PointNet has shown the potential for improved performance in a few benchmarks [46], a single max pooling operation can be insufficient to obtain the local features at different scales. Thus, PointNet++ is developed to implement a hierarchical grouping of points, in which the local features at multiple scales can be effectively extracted to train the model. As the segmentation and classification performance of PointNet++ outperforms that of PointNet in validations using various datasets [52], PointNet++ is utilized in this study for the segmentation of point clouds into the bridge components.

2.2. Bridge component recognition using UAV imagery

This section describes the process of classifying UAV imagery into the bridge components using the segmentation model developed in Section 2.1. For this purpose, the segmentation model is applied to the point cloud data generated by UAV imagery, and the resulting labels are utilized to identify the class of the bridge component captured in each close-range image. To minimize the effect of different orientations of point clouds, a set of points is aligned with the *y*-axis defined in Section 2.1. The classified labels resulting from the segmentation model are updated based on the label correction to increase the identification accuracy; subsequently, the entire point cloud is projected onto each camera coordinate system to recognize the bridge components. The overall process for automated bride component recognition can be summarized as the follows: (1) data alignment, (2) subspace extraction, (3) data normalization, (4) label correction, and (5) coordinate transformation.

After obtaining the point cloud by SfM, all the bridge spans are made parallel to the *y*-axis for the successful segmentation of 3D points with minimizing the effect of the rotation, following the procedure in Section 2.1. To efficiently align the point cloud, an image captured in the bridge's longitudinal direction is used as a reference, and the transformation between different coordinate systems (i.e., UAV camera coordinate system and laser scanner coordinate system) is obtained to match the orientations of point clouds. Let $W = [X\ Y\ Z]^{\mathrm{T}}$ be the coordinates of a 3D point resulting from SfM in the world coordinate system, and $C = [x\ y\ z]^{\mathrm{T}}$ be the coordinates of the projection of W onto the camera coordinate system. If the first image taken by the UAV is selected as a reference, the projection of the point cloud onto the first camera coordinate system can be expressed as follows:

$$C_1 = R_1 W + T_1 \tag{1}$$

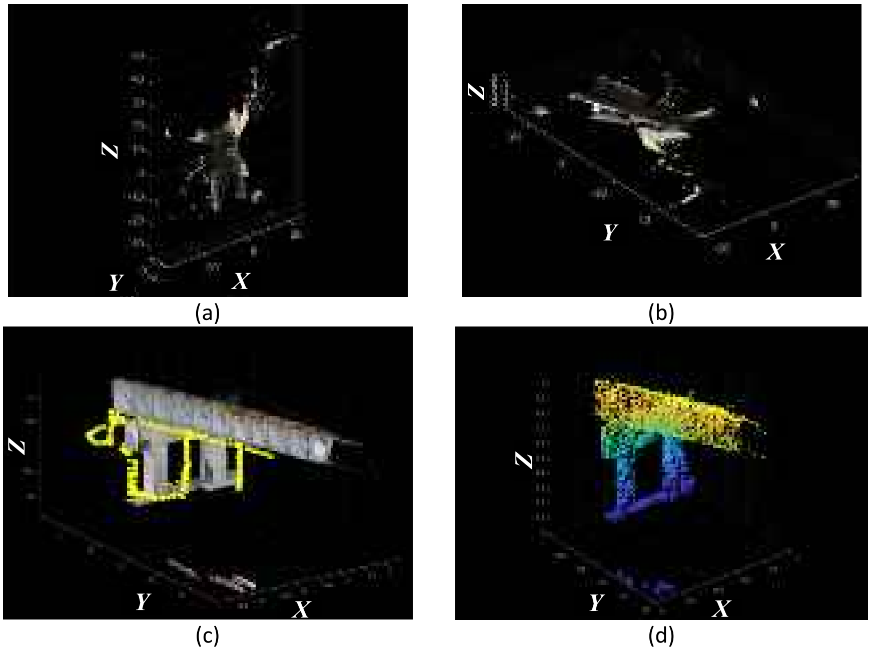


Fig. 4. Data processing of point cloud taken by UAV survey: (a) original point cloud, (b) data alignment, (c) subspace extraction, and (d) data normalization.

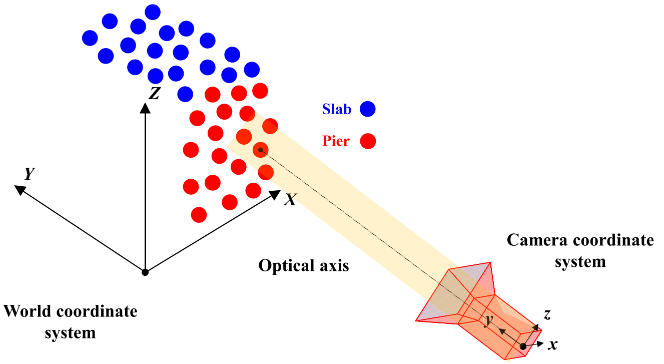


Fig. 5. Coordinate transformation for identifying the bridge component in close-range image.

where R_1 and T_1 are the rotation matrix and translation vector of the first camera pose, respectively, and C_1 is the camera coordinate system of the first image taken by the UAV. The rotation that aligns the point cloud generated by UAV imagery with the orientation defined during the training stage is represented by a transformation matrixR_s:

$$W^{'} = R_s W \tag{2}$$

where W' is the transformed coordinates in which a bridge span in the point cloud is parallel to the y-axis. Thus, the coordinates of the projection of W' onto the first camera coordinate system can be represented as follows:

$$C_1 = R_1 R_s^{-1} W^{'} + T_1 \tag{3}$$

Because the first image is assumed to be captured along the bridge span, the rotation terms in Eq. (3) can be used to align the camera with the y-axis in the training stage.

$$R_1 R_s^{-1} \cong \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} = R_x \tag{4}$$

where R_x is the rotation matrix that rotates the vector by 90° about x-axis to match the orientations of point clouds obtained by the laser scanner and UAV. Now the unknown,R_s, can be calculated in Eq. (4), and each camera pose is expressed as follows:

$$C_i = R_i R_s^{-1} W^{'} + T_i \tag{5}$$

where R_i and T_i are the rotation matrix and translation vector of the i-th camera pose, respectively, and C_i is the i-th camera coordinate system. Then, the optical center O_i for i-th camera in the world coordinate system can be derived from $C_i = [0\,0\,0]^{\mathrm{T}}$ in Eq. (5).

$$O_i = -R_s R_i^{-1} T_i \tag{6}$$

Fig. 4(a) and (b) show the original point cloud and the aligned point cloud, respectively. After data alignment, a subspace that is the smallest cuboid containing a group of adjacent 3D points is extracted from the entire point cloud based on the locations of optical centers, as shown in yellow cameras in Fig. 4(c); subsequently, a constant number of points is randomly sampled and then normalized for the input of semantic segmentation (see Fig. 4(d)). For data normalization, the minimum value in each axis is subtracted from the corresponding coordinate of 3D points, which is further divided by the range. As such, the segmentation model obtained by the laser scanner can be applied to a point cloud generated by UAV imagery with minimizing the effect of transformation of point clouds.

The point cloud data obtained by SfM is different from those collected by 3D laser scanners in terms of measurement noise and point cloud density. Thus, to improve the classification performance of each point, a predetermined number of nearest points is searched for each point after applying the segmentation model. Subsequently, the most frequent class is selected to update the original label (label correction step). Note that the effect of the number of nearest points in the label correction is investigated in Section 3.1, determining the optimal value.

After the label correction, the class of bridge components in each close-range image is identified by transforming the entire point cloud with the associated labels into the camera coordinate system. Let $[x_i(j)\ y_i(j)\ z_i(j)]^{\mathrm{T}}$ be the j-th point expressed in the i-th camera coordinate system. Then,

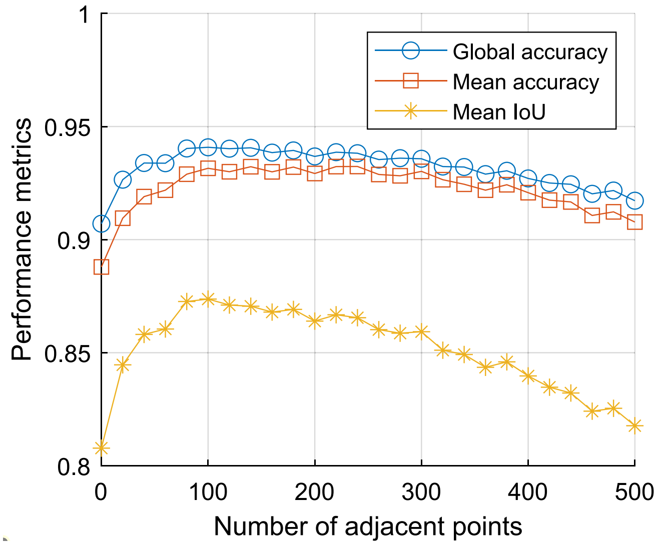


Fig. 6. Performance metrics with respect to the number of adjacent points in label correction.

$$\begin{bmatrix} x_i(j) \\ y_i(j) \\ z_i(j) \end{bmatrix} = C_i(j) = R_i R_s^{-1} W'(j) + T_i \tag{7}$$

where $W'(j)$ and $C_i(j)$ are the j-th point expressed in the world coordinate system and the i-th camera coordinate system, respectively. To categorize each close-range image into the bridge component, the distance between the optical axis and the projected point cloud is used as an indicator as follows:

$$D_i(j) = \left\| \begin{bmatrix} x_i(j) \\ y_i(j) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\|_2 \tag{8}$$

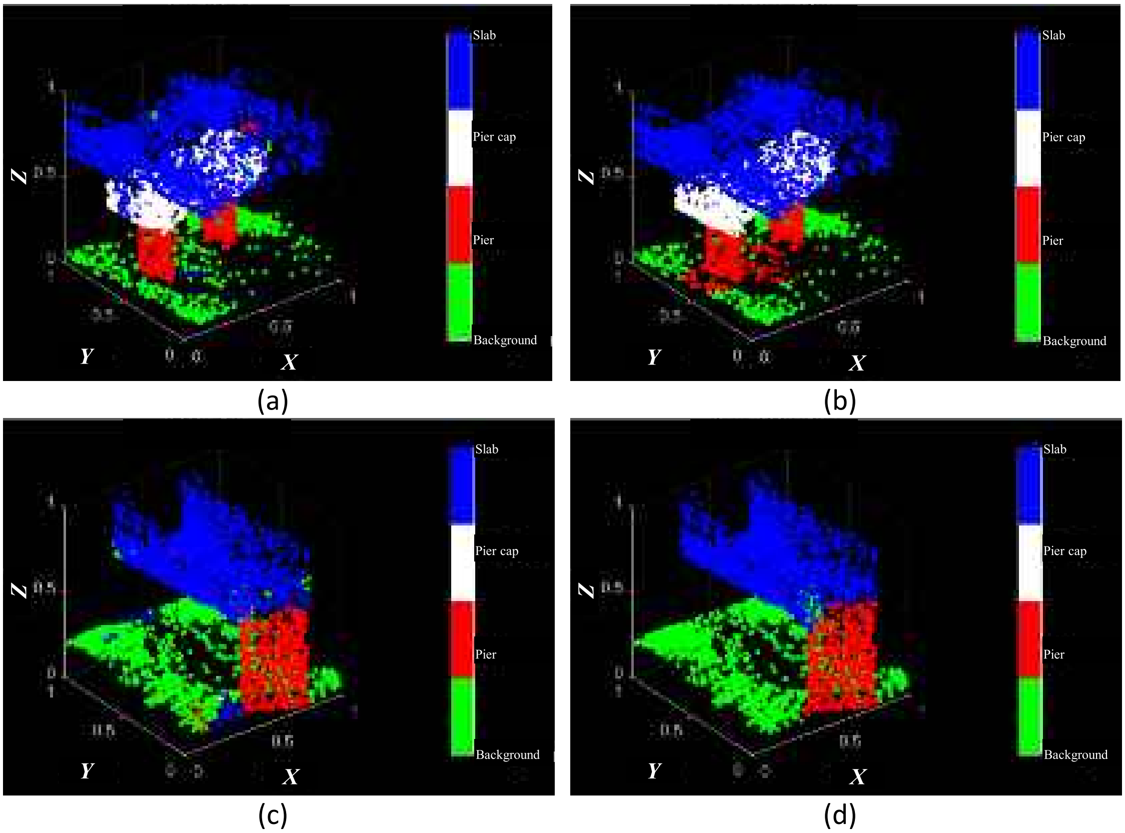
where $D_i(j)$ is the distance between the i-th optical axis and the j-th point in the i-th camera coordinate system, and $||\cdot||_2$ is the L2 norm. Fig. 5 shows the coordinate transformation between the world and camera coordinates, recognizing the category of the bridge component in the close-range image. For example, when the point cloud is projected onto the camera coordinate system, the most frequent class of points that are close to the optical axis is the bridge pier. Thus, the corresponding close-range image can be classified as “Pier”.

3. Experimental validation

This section investigates the performance of proposed automated bridge component recognition approach. First, the segmentation model is trained using a set of point clouds derived from laser scans of full-scale bridges. Subsequently, the application of the proposed framework to a point cloud reconstructed from UAV imagery is field validated using a highway bridge in Urbana, Illinois, USA.

3.1. Semantic segmentation network

The point clouds derived from the laser scans are used to train the segmentation model. A total of seven different full-scale bridges in South Korea are utilized [42], in which the total lengths range from 54 to 272 m, with span lengths ranging between 17 and 38 m. As described in Section 2.1, all the point clouds are manually classified into four classes: background, pier, pier cap, and slab using an open-source system MeshLab [53]. Subsequently, the entire point cloud is divided into multiple subspaces along the bridge spans (y-axis). Here, the subspace length and its overlap are set to 10 m and 80 %, respectively, following



(a)

(b)

(c)

(d)

Fig. 7. Typical semantic segmentation results: (a) prediction, (b) label correction for (a), (c) prediction, and (d) label correction for (c).

Fig. 8. Experimental validation: (a) UAV and (b) highway bridge.

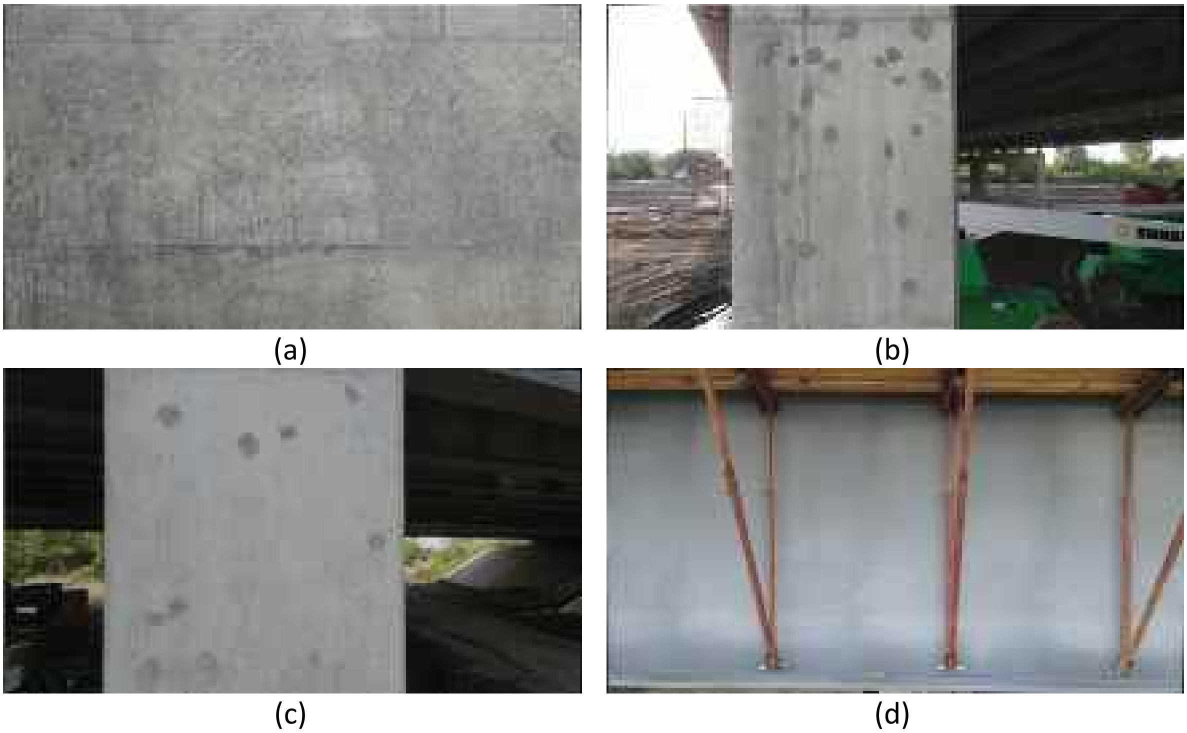


Fig. 9. Typical close-range images: (a) background, (b) pier, (c) pier cap, and (d) slab.

Table 3
Experimental cases.

Case	Algorithms used[a]		
	PointNet++	Data alignment	Label correction
1	O	X	X
2	O	O	X
3	O	O	O

[a] Subspace extraction and data normalization are included.

the previous study [42]. In each subspace, 4,096 points which include the bridge pier are randomly selected until all the points therein are used; this procedure results in a total of 1,461 point clouds, each containing 4,096 points, for the seven bridges. After normalizing, 60 %, 20 %, and 20 % of the 1,461 point-cloud datasets are extracted for training, validation, and testing stages, respectively. For the training and validation stages, PointNet++ is run on a PC with an Intel Core i7 10750H 2.6 GHz processor, an NVIDIA GeForce GTX 1660Ti 6 GB graphics card, and 16 GB RAM. Note that the batch size is 8 with the initial learning rate of 0.0001; network training is stopped after 50 epochs. Here, such hyperparameters are defined empirically based on the literature [41–43].

During the testing stage, the segmentation performance of the obtained model is evaluated using a set of point clouds that are not employed in the training and validation stages. Because the label correction is presented in this study to update the classified label in each point based on its adjacent points, the identification accuracy is further investigated with respect to the number of nearest points. Here, global accuracy, a ratio of correctly identified points to the total number of points, is utilized as an indicator to compare the results. In addition, mean accuracy and mean intersection over union (IoU) are introduced as follows:

$$\text{Mean accuracy} = \frac{1}{N}\sum_{k=1}^{N}\{TP_k/(TP_k + FN_k)\} \tag{9}$$

$$\text{Mean IoU} = \frac{1}{N}\sum_{k=1}^{N}\{TP_k/(TP_k + FP_k + FN_k)\} \tag{10}$$

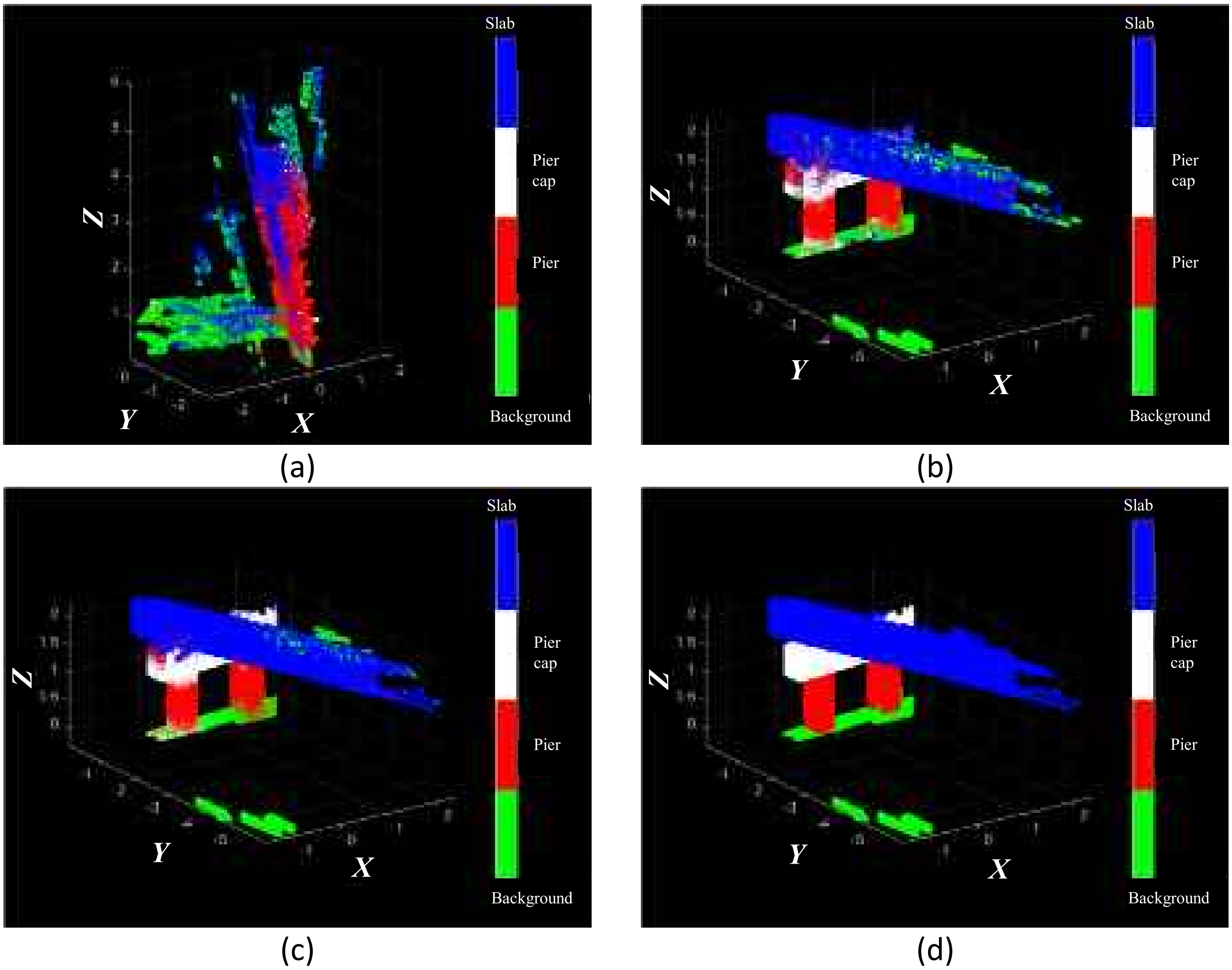


Fig. 10. Semantic segmentation results of points: (a) Case 1, (b) Case 2, (c) Case 3, and (d) ground truth.

where TP_k,FP_k, and FN_k represent true positives, false positives, and false negatives of the k-th class, respectively, and N is the number of classes. Fig. 6 shows the changes of performance metrics with respect to the number of adjacent points in the label correction. If the original point label is replaced by the most frequent class among 100 nearest points, the highest accuracy can be achieved, in which the global accuracy increases from 0.91 to 0.94. Representative examples to show the improved performance in such case are presented in Fig. 7. While several incorrect detections are found in the results of the raw PointNet++ output (see Fig. 7(a) and (c)), most of them are corrected by the proposed approach, as shown in Fig. 7(b) and (d).

3.2. *Bridge component recognition*

The proposed framework for automated bridge component recognition is validated using a highway bridge located in Urbana, Illinois, USA. Note that shape of this bridge is quite different with the dataset collected from the South Korea; additionally, traffic was temporarily blocked on this bridge due to the reconstruction process, as shown in Fig. 8. A commercial drone (DJI Mavic Air 2) is utilized to capture the imagery. Here, the UAV is flying in proximity to the target surface, in which the video is recorded for around 10 min by the installed camera with an image resolution of 3840 × 2160 and a frame rate of 30 fps. A sequence of images with at least 80 % overlap is extracted from the video data by sampling at regular intervals, generating a point cloud of the structure. A total of 222 images are finally collected and input to openMVG (https://github.com/openMVG/openMVG), so as to construct a sparse point cloud with the associated camera poses; subsequently, openMVS (https://github.com/cdcseacave/openMVS) is employed to build a dense point cloud based on a patch-match approach [54]. Here, the generation of a dense point cloud takes about 6 h. Fig. 9 shows the typical close-range images used in this study, including background, pier, pier cap, and slab. Note that the foundation under the bridge piers (see Fig. 9(a)) is also defined as background in this study, because such foundation in the training stage is covered by background, as shown in Fig. 7. Although these close-range images provide detailed information of the structural surface at a high pixel density, the lack of contextual information makes correctly recognizing the class of bridge components using images challenging, even with human intervention.

The performance of the proposed framework for automated bridge component recognition is evaluated in terms of the SfM point cloud with the associated camera poses. As described in Section 2.2, the segmentation model obtained from laser scans of full-scale bridges is applied to the aligned point cloud data, and the raw semantic segmentation results are updated based on their adjacent points. To validate the effects of data alignment and label correction on the classification performance, the identification accuracy of the proposed approach is compared with those of the PointNet++ algorithm alone and the PointNet++ algorithm with only the data alignment. For example, Table 3 represents a total of three cases considered in this study. Note that all the cases include both the subspace extraction and data normalization; in which the most frequent class among 100 adjacent points is employed for the label correction, because of its highest accuracy (see Fig. 6).

Fig. 10 shows the results of applying three different cases to a point

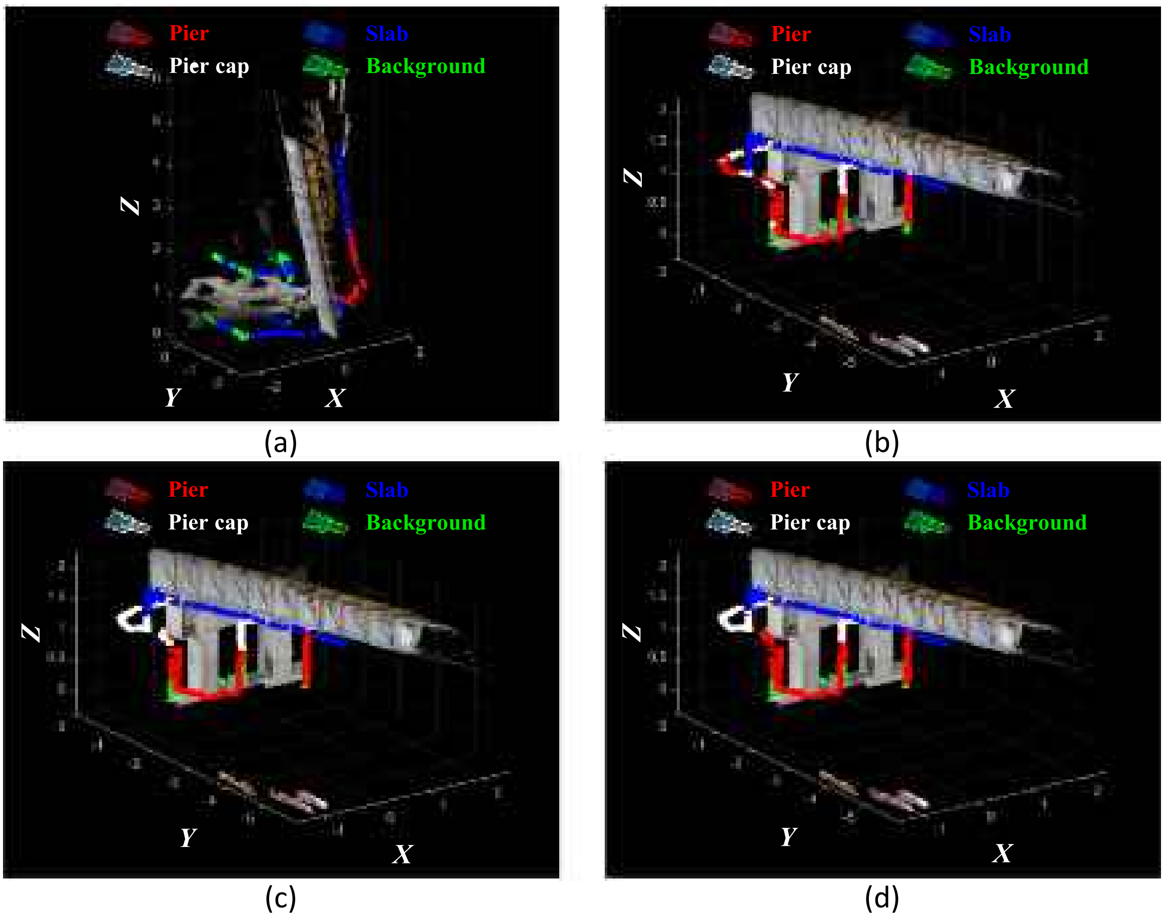


Fig. 11. Classification results of images: (a) Case 1, (b) Case 2, (c) Case 3, and (d) ground truth.

Table 4
Quantitative results of 3D point cloud and 2D imagery.

	3D point cloud (172,032 points)			2D imagery (222 images)		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
Global accuracy	0.31	0.78	0.87	0.17	0.88	0.94
Mean accuracy	0.30	0.74	0.83	0.36	0.90	0.96
Mean IoU	0.15	0.58	0.71	0.09	0.80	0.91

cloud generated by UAV imagery. By performing both data alignment and label correction, most of the points are successfully categorized into appropriate bridge components. To further identity the bridge component class in each close-range image, the entire point cloud is projected onto each camera coordinate system. Similar to the label correction, 100 points that are close to the optical axis are selected to determine the dominant bridge component class associated with the image. All the camera poses are visualized with the color corresponding to the predicted bridge component categories in Fig. 11. These results demonstrate that the proposed approach can classify the bridge component in each image successfully, based on their global-scale 3D geometric information.

Quantitative performance evaluation is conducted using ground truth labels from a total of 172,032 points which have been assigned manually (background, pier, pier cap, and slab). The bridge component class for each of the 222 images is then recognized based on the coordinate transformation between the world and camera coordinate systems. Table 4 shows the performance metrics for the 3D point cloud and 2D imagery. For the 3D point cloud, overall accuracies of the proposed approach are higher than using PointNet++ alone, showing the effectiveness of both data alignment and label correction steps. Furthermore, the most frequent class among multiple points near the optical axis is effective for the classification of close-range images. As such, considering global-level 3D geometric information is useful in achieving accurate bridge component recognition from close-range images.

Fig. 12 shows the efficacy of the proposed approach in using a series of close-range images associated with the specific bridge component. As the entire image is used to generate a point cloud of the bridge, a proper number of features needs to be selected due to the computational burden. Thus, a point cloud of the obtained bridge pier may have a low density, as shown in Modal A of Fig. 12(a). In contrast, because 38 images associated with a bridge pier are determined automatically using the proposed approach, a high number of features can be efficiently utilized for such images to construct a dense point cloud (see Model B in Fig. 12(b)). Table 5 shows the comparison between two models in terms of the number of points and the computational time. Consequently, the proposed approach can be effective for the inspection of each bridge component.

To further explore the classification performance of the proposed approach, the projection of the entire point cloud onto the image coordinate system is checked. Here, the intrinsic matrix including the focal length and focal point is also employed in Eq. (7) to map the points in the camera coordinate system onto the image coordinate system. Fig. 13 shows the projected points in each image coordinate system corresponding to Fig. 9. As can be seen here, the proposed approach can perform the successful classification of close-range images based on their 3D geometric information. Note that the points under the center of the image in Fig. 13(a) are not used here, because a set of 3D points are extracted from the entire point cloud utilizing the locations of optical

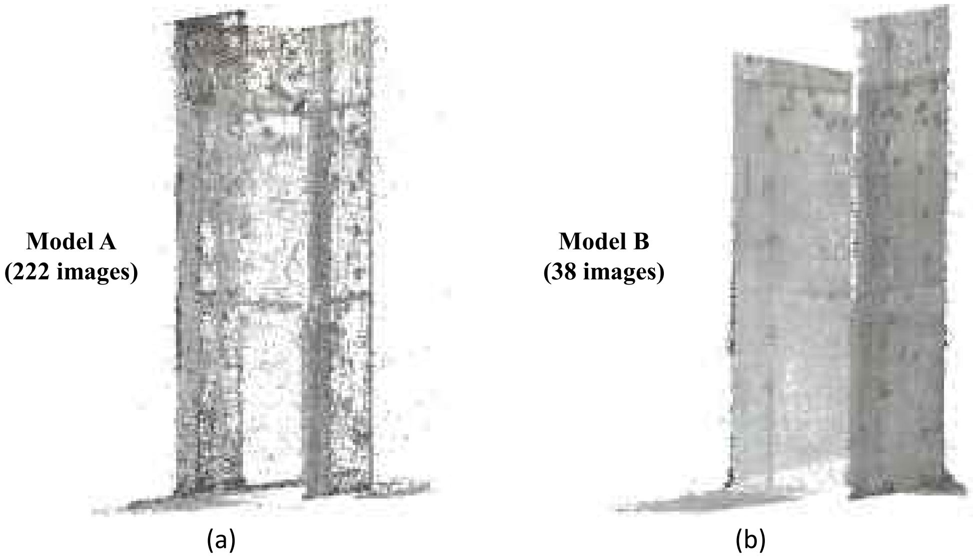


Fig. 12. Point clouds: (a) Model A (222 images) and (b) Model B (38 images associated with a bridge pier).

Table 5
Quantitative comparison of point clouds.

Point cloud	Model A	Model B
Total images	222	38
Total points	142,259	332,082
Computational time	6 h	1.3 h

centers, as described in Section 2.2. However, as shown in Fig. 14, the bridge pier can be misclassified as the pier cap, because of the relatively low accuracy near the boundary between different bridge components. Thus, advances in semantic segmentation algorithms that are optimized for points on the boundary may enhance the performance of the proposed approach.

4. Conclusion

This study presented a framework for automated bridge component recognition based on close-range images taken by the UAV. The proposed approach was designed to process a set of close-range images typically required for identifying structural damage, enabling the automated classification of the associated bridge component for each image. In contrast to 2D images, 3D point clouds derived from laser

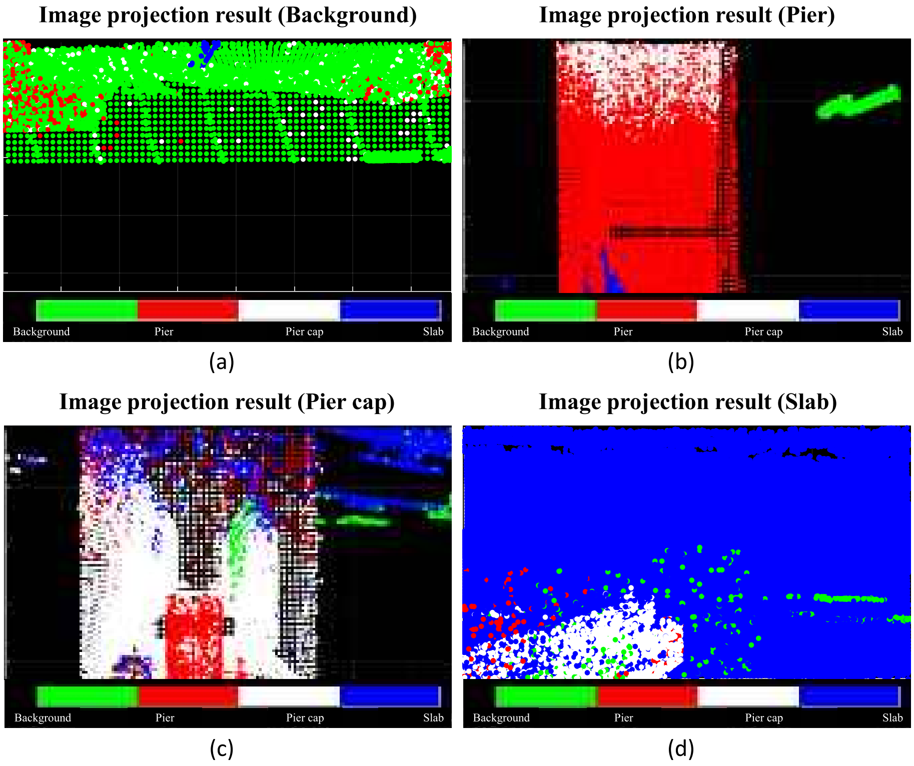


Image projection result (Background)
Background
Pier
Pier cap
Slab
(a)
Image projection result (Pier)
Background
Pier
Pier cap
Slab
(b)
Image projection result (Pier cap)
Background
Pier
Pier cap
Slab
(c)
Image projection result (Slab)
Background
Pier
Pier cap
Slab
(d)

Fig. 13. Typical projection results: (a) background, (b) pier, (c) pier cap, and (d) slab.

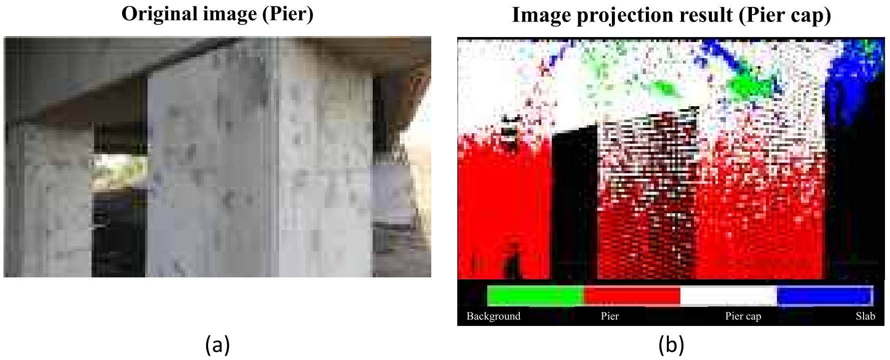


Fig. 14. False detection: (a) close-range image for pier and (b) projection result.

scans of full-scale bridges can effectively provide the geometric and topological properties of bridge components. To realize the full potential of such 3D information for the successful classification of close-range images, a combination of point cloud data and UAV imagery was considered in this study. In detail, a set of point clouds were collected from the full-scale bridges utilizing the laser scanner, from which the segmentation model was trained from their distinctive 3D features. A point cloud generated by UAV imagery using SfM was rearranged and then classified into the bridge components based on the label correction; subsequently, the class of bridge components in each image was determined from the coordinate transformation between the world and camera coordinate systems. The classification performance of the proposed approach was field validated using a highway bridge in Urbana, Illinois, USA, from which the following conclusions can be drawn:

(1) The proposed framework can enable automated bridge component recognition based on close-range images acquired by a UAV, in which the 3D point cloud and the associated 2D imagery were successfully classified into four classes: background, pier, pier cap, and slab. According to the quantitative results, the global accuracies of the proposed framework were 0.87 and 0.94 for the classified point cloud and close-range image, respectively. In contrast, the global accuracies of the point cloud and image were recorded as 0.31 and 0.17, respectively, when using PointNet++ alone.
(2) The processes of both data alignment and label correction steps were shown to improve the classification accuracy. According to the parametric analysis, the highest accuracy can be achieved if the most frequent class of the 100 adjacent points was utilized to update the classified label for each point. Here, the global accuracy was 0.94.
(3) When the classified labels in the SfM point cloud were projected onto the image coordinate system using the intrinsic matrix, most of the misclassified points were detected near the boundary between different bridge components. For example, some images captured from the top of the bridge pier were incorrectly classified into the pier cap. Thus, implementing an advanced 3D semantic segmentation network can better deal with boundary regions and may improve the identification accuracy.

Consequently, the proposed approach can enable the successful classification of close-range images into the bridge components, facilitating more practical evaluation of the status of full-scale bridges.

CRediT authorship contribution statement

Hyunjun Kim: Conceptualization, Methodology, Writing – original draft. **Yasutaka Narazaki:** Methodology, Writing – review & editing. **Billie F. Spencer Jr.:** Supervision, Writing – review & editing.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

Acknowledgement

This study was financially supported by Seoul National University of Science & Technology.

References

[1] American Society of Civil Engineers (ASCE). *2021 Infrastructure Report Card – Bridges*. https://infrastructurereportcard.org/wp-content/uploads/2020/12/Bridges-2021.pdf. 2021; (Accessed data: March 2022).
[2] Federal Highway Administration (FHWA). *National bridge inspection standards*. Federal Register 2004;69(239):74419–74439.
[3] Morgenthal G, Hallermann N. Quality assessment of unmanned aerial vehicle (UAV) based visual inspection of structures. Adv Struct Eng 2014;17(3):289–302. https://doi.org/10.1260/1369-4332.17.3.289.
[4] Kim H, Lee J, Ahn E, Cho S, Shin M, Sim S-H. Concrete crack identification using a UAV incorporating hybrid image processing. Sensors 2017;17(9):2052. https://doi.org/10.3390/s17092052.
[5] Valença J, Santos B, Araújo A, Júlio E. Aerial crack view: crack monitoring in concrete bridges through image processing acquired by UAV. In: Proceedings of the 2019 International Association for Bridge and Structural Engineering, New York, NY, Sep 2019, pp. 2458–2465.
[6] Erdenebat D, Waldmann D. Application of the DAD method for damage localisation on an existing bridge structure using close-range UAV photogrammetry. Eng Struct 2020;218:110727. https://doi.org/10.1016/j.engstruct.2020.110727.
[7] Liu Y-F, Nie X, Fan J-S, Liu X-G. Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction. Comput-Aided Civil Infrastruct Eng 2020;35(5):511–29. https://doi.org/10.1111/mice.12501.
[8] Kim H, Ahn E, Cho S, Shin M, Sim S-H. Comparative analysis of image binarization methods for crack identification in concrete structures. Cem Concr Res 2017;99:53–61. https://doi.org/10.1016/j.cemconres.2017.04.018.
[9] Valença J, Puente I, Júlio E, González-Jorge H, Arias-Sánchez P. Assessment of cracks on concrete bridges using image processing supported by laser scanning survey. Constr Build Mater 2017;146:668–78. https://doi.org/10.1016/j.conbuildmat.2017.04.096.
[10] Sainz A. NBC News – Inspector who failed to catch Mississippi River bridge crack is fired. https://www.nbcnews.com/news/us-news/inspector-who-failed-catch-mississippi-river-bridge-crack-fired-n1267723. 2021; (Accessed data: March 2022).
[11] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Proceedings of the 2012 International Conference on Neural Information Processing Systems, Lake Tahoe, NV, Dec 2012, pp. 1097–1105.
[12] Cha Y-J, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Comput-Aided Civil Infrastruct Eng 2017;32(5):361–78. https://doi.org/10.1111/mice.12263.
[13] Jang K, Kim N, An Y-K. Deep learning–based autonomous concrete crack evaluation through hybrid image scanning. Struct Health Monit 2019;18(5–6):1722–37. https://doi.org/10.1177/1475921718821719.

[14] Alipour M, Harris DK. Increasing the robustness of material-specific deep learning models for crack detection across different materials. Eng Struct 2020;206:110157. https://doi.org/10.1016/j.engstruct.2019.110157.

[15] Kim H, Sim S-H, Spencer BF. Automated concrete crack evaluation using stereo vision with two different focal lengths. Autom Constr 2022;135:104136. https://doi.org/10.1016/j.autcon.2022.104136.

[16] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, June 2014, pp. 580–587.

[17] Girshick R. Fast R-CNN. In: Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, December 2015, pp. 1440–1448.

[18] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 2017;39(6): 1137–49. https://doi.org/10.1109/TPAMI.2016.2577031.

[19] Cha Y-J, Choi W, Suh G, Mahmoudkhani S, Büyüköztürk O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Comput-Aided Civil Infrastruct Eng 2018;33(9):731–47. https://doi.org/10.1111/mice.12334.

[20] Mondal TG, Jahanshahi MR, Wu R-T, Wu ZY. Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance. Struct Control Health Monit 2020;27(4):e2507.

[21] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, June 2015, pp. 3431–3440.

[22] He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In: Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, October 2017, pp. 2980–2988.

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

[24] Hoskere V, Narazaki Y, Hoang TA, Spencer Jr BF. MaDnet: multi-task semantic segmentation of multiple types of structural materials and damage in images of civil infrastructure. J Civ Struct Health Monit 2020;10:757–73. https://doi.org/10.1007/s13349-020-00409-0.

[25] Kim B, Cho S. Automated multiple concrete damage detection using instance segmentation deep learning model. Appl Sci-Basel 2020;10(22):8008. https://doi.org/10.3390/app10228008.

[26] Narazaki Y, Hoskere V, Hoang TA, Fujino Y, Sakurai A, Spencer Jr BF. Vision-based automated bridge component recognition with high-level scene consistency. Comput-Aided Civil Infrastruct Eng 2020;35(5):465–82. https://doi.org/10.1111/mice.12505.

[27] Saovana N, Yabuki N, Fukuda T. Development of an unwanted-feature removal system for structure from motion of repetitive infrastructure piers using deep learning. Adv Eng Inform 2020;46:101169. https://doi.org/10.1016/j.aei.2020.101169.

[28] Karim MM, Qin R, Chen G, Yin Z. A semi-supervised self-training method to develop assistive intelligence for segmenting multiclass bridge elements from inspection videos. Struct Health Monit 2021:1–18. https://doi.org/10.1177/14759217211010422.

[29] Narazaki Y, Hoskere V, Yoshida K, Spencer BF, Fujino Y. Synthetic environments for vision-based structural condition assessment of Japanese high-speed railway viaducts. Mech Syst Signal Proc 2021;160:107850. https://doi.org/10.1016/j.ymssp.2021.107850.

[30] Park G, Lee JH, Yoon H. Semantic structure from motion for railroad bridges using deep learning. Appl Sci-Basel 2021;11(10):4332. https://doi.org/10.3390/app11104332.

[31] Sajedi SO, Liang X. Uncertainty-assisted deep vision structural health monitoring. Comput-Aided Civil Infrastruct Eng 2021;36(2):126–42. https://doi.org/10.1111/mice.12580.

[32] Saovana N, Yabuki N, Fukuda T. Automated point cloud classification using an image-based instance segmentation for structure from motion. Autom Constr 2021; 129:103804. https://doi.org/10.1016/j.autcon.2021.103804.

[33] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 2017;39(12):2481–95. https://doi.org/10.1109/TPAMI.2016.2644615.

[34] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: Proceedings of the 2015 International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, October 2015, pp. 234–241.

[35] Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the 2018 European Conference on Computer Vision, Munich, Germany, September 2018, pp. 833–851.

[36] Jégou S, Drozdzal M, Vazquez D, Romero A, Bengio Y. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Honolulu, HI, July 2017, pp. 1175–1183.

[37] Bolya D, Zhou C, Xiao F, Lee YJ. Yolact: real-time instance segmentation. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, Seoul, South Korea, October 2019, pp. 9156–9165.

[38] Riveiro B, DeJong MJ, Conde B. Automated processing of large point clouds for structural health monitoring of masonry arch bridges. Autom Constr 2016;72(3): 258–68. https://doi.org/10.1016/j.autcon.2016.02.009.

[39] Lu R, Brilakis I. Digital twinning of existing reinforced concrete bridges from labelled point clusters. Autom Constr 2019;105:102837. https://doi.org/10.1016/j.autcon.2019.102837.

[40] Lu R, Brilakis I, Middleton CR. Detection of structural components in point clouds of existing RC bridges. Comput-Aided Civil Infrastruct Eng 2019;34(3):191–212. https://doi.org/10.1111/mice.12407.

[41] Kim H, Kim C. Deep-learning-based classification of point clouds for bridge inspection. Remote Sens 2020;12(22):3757. https://doi.org/10.3390/rs12223757.

[42] Kim H, Yoon J, Sim S-H. Automated bridge component recognition from point clouds using deep learning. Struct Control Health Monit 2020;27(9):e2591.

[43] Lee JS, Park J, Ryu Y-M. Semantic segmentation of bridge components based on hierarchical point cloud model. Autom Constr 2021;130:103847. https://doi.org/10.1016/j.autcon.2021.103847.

[44] Yan Y, Hajjar JF. Automated extraction of structural elements in steel girder bridges from laser point clouds. Autom Constr 2021;125:103582. https://doi.org/10.1016/j.autcon.2021.103582.

[45] Truong-Hong L, Lindenbergh R. Automatically extracting surfaces of reinforced concrete bridges from terrestrial laser scanning point clouds. Autom Constr 2022; 135:104127. https://doi.org/10.1016/j.autcon.2021.104127.

[46] Qi CR, Su H, Mo K, Guibas LJ. Pointnet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, July 2017, pp. 77–85.

[47] Hua B-S, Tran M-K, Yeung S-K. Pointwise convolutional neural networks. In: Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, June 2018, pp. 984–993.

[48] Wang Y, Sun Y, Liu Z, Sarma SE, Bronstein MM, Solomon JM. Dynamic graph cnn for learning on point clouds. ACM Trans Graph 2019;38(5):1–12. https://doi.org/10.1145/3326362.

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

[50] Pan Y, Dong Y, Wang D, Chen A, Ye Z. Three-dimensional reconstruction of structural surface model of heritage bridges using UAV-based photogrammetric point clouds. Remote Sens 2019;11(10):1204. https://doi.org/10.3390/rs11101204.

[51] Perry BJ, Guo Y, Atadero R, van de Lindt JW. Streamlined bridge inspection system utilizing unmanned aerial vehicles (UAVs) and machine learning. Measurement 2020;164:108048. https://doi.org/10.1016/j.measurement.2020.108048.

[52] Qi CR, Yi L, Su H, Guibas LJ. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Proceedings of the 2017 International Conference on Neural Information Processing Systems, Long beach, CA, December 2017, pp. 1–10.

[53] Cignoni P, Callieri M, Corsini M, Dellepiane M, Ganovelli F, Ranzuglia G. Meshlab: an open-source mesh processing tool. In: Proceedings of the 2008 Eurographics Italian Chapter Conference, Salerno, Italy, July 2008, pp. 129–136.

[54] Barnes C, Shechtman E, Finkelstein A, Goldman DB. PatchMatch: a randomized correspondence algorithm for structural image editing. ACM Trans Graph 2009;28 (3):1–11. https://doi.org/10.1145/1531326.1531330.