Automated semantic segmentation of bridge point cloud based on local descriptor and machine learning

Automated semantic segmentation of bridge point cloud based on local descriptor and machine learning

Tian Xia[a,b], Jian Yang[a,b,*], Long Chen[c,d]

[a] *Shanghai Key Laboratory for Digital Maintenance of Buildings and Infrastructures, School of Naval Architecture, Ocean & Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, PR China*

[b] *State Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University, Shanghai 200240, PR China*

[c] *Lloyds Register Foundation/Data Centric Engineering Programme, The Alan Turing Institute, London NW1 2DB, UK*

[d] *School of Architecture, Building and Civil Engineering, Loughborough University, Loughborough LE11 3TU, UK*

Abstract

In recent years, monitoring the health condition of existing bridges has become a common requirement. By providing an information management system, Bridge Information Model (BrIM) can highly improve the efficiency of health inspection and the reliability of condition evaluation. However, the current modeling processes still largely rely on manual work, where the cost outweighs the benefits. The main barrier lies in the challenging step of semantic segmentation of point clouds. Efforts have been made to identify and segment the structural components of bridges in existing research. But these methods are either dependent on manual data preprocessing or need big training dataset, which, however, has rendered them unpractical in real-world applications. This paper presents a combined local descriptor and machine learning based method to automatically detect structural components of bridges from point clouds. Based on the geometrical features of bridges, we design a multi-scale local descriptor, which is then used to train a deep classification neural network. In the end, a result refinement algorithm is adopted to optimize the segmentation results. Experiments on real-world reinforced concrete (RC) slab and beam-slab bridges show an average precision of 97.26%, recall of 98.00%, and intersection over union (IoU) of 95.38%, which significantly outperforms PointNet. This method has provided a potential solution to semantic segmentation of infrastructures by small sample learning and will contribute to the fulfillment of the automatic BrIM generation of typical highway bridges from the point cloud in the future.

Keywords: bridge information model, semantic segmentation, local descriptor, machine learning, point cloud

1. Introduction

In most countries, the prominent status of bridges has already evolved to the maintenance stage. According to the American Society of Civil Engineers (ASCE), more than 617,000 bridges are in service in the United States, while approximately 42% are at least 50 years old. It has been estimated that 125 billion dollars will be needed to maintain existing bridges [1].

Time-consuming inspection and evaluation processes in bridge maintenance are part of the reasons behind the vast costs. Efforts have been made to automatically detect surface damages of bridges. Kim et al. [2] utilized Region with Convolutional Neural Networks (R-CNN) to identify cracks on the structure surface from the images captured by unmanned aerial vehicles (UAV). Wang et al. [3] developed a stitching algorithm to address the calculation of long cracks which could not be photographed in one picture. However, their work could only identify local damages. Global geometric context is still needed to integrate local inspection results to accomplish a comprehensive condition evaluation. At present, Bridge Management Systems (BMS) like AASHTOWare [4] are used to store and manage the structured condition information of bridges. However, BMS are primarily designed for system-wide decision-making instead of assessing the structural condition of a specific bridge component [5].

As the alternative tool, Bridge Information Model (BrIM) can manage this information on both structure and component levels. BrIM is an extension of Building Information Model (BIM), which is defined as "a digital representation of physical and functional characteristics of a facility" by the U.S. National Building Information Model Standard Project Committee [6]. BrIM is not merely a 3D geometric model but also capable of storing information on the component level for the life cycle management [7]. Tanaka et al. [8] developed an Industry Foundation Classes (IFC) based BrIM to manage acquired inspection data. DiBernardo [9] reviewed the current data management process of existing bridge assets and proposed a framework integrating BrIM and existing commercial bridge software applications to organize and analyze inspection data. In addition to managing inspection data, BrIM can also facilitate collaboration and interoperation of stakeholders through the life cycle management of bridges, including structural health monitoring, rehabilitation, behavior modeling, and prediction [7].

Motivated by the benefits, the adoption of BIM for newly-built transportation infrastructures

powerfully surged between 2012 and 2017 in Europe. However, only 52% of engineers and contractors had implemented BIM on at least half of their projects by 2017 [10]. Only a few newly-built bridges have as-designed BrIM, while most existing bridges still rely on traditional information management tools such as datasheets. Thus, to digitalize the life cycle management of existing bridges, automatic generation of as-built BrIM is pressingly needed.

In general, creating the as-built BIM for an existing structure involves four steps, i.e., 3D reconstruction, semantic modeling, geometrical modeling, and building information modeling [11]. In 3D reconstruction, reality capture devices are deployed to produce digital representations of existing structures, e.g., point cloud. It is followed by three respective modeling processes, aiming to generate BIM from the digital representation. For instance, in semantic modeling, the subsets of the 3D reconstruction are assigned with labels in a BIM taxonomy. Subsequently, in geometrical modeling, the parametric representation of the shape, the location, and the spatial relationship of each class instance will be established. Finally, in building information modeling, the semantic and geometrical parameters will be integrated and written to files according to the specified format of BIM (e.g., IFC), generating building information models.

To date, widespread reality capture technologies are available to fulfill 3D reconstruction by generating point cloud representations. Park et al. [12] utilized terrestrial laser scanning (TLS) to acquire the point cloud model of a simply supported steel beam to monitor its displacements. Brandon et al. [13] implemented photogrammetry based on 2D images to create point clouds of bridges so that surface condition and geometric information can be visualized. Roca et al. [14] captured point cloud data of buildings through a lidar equipped on UAV, aiming to generate building envelope models and perform energy analysis.

Despite the significant developments of data-acquisition methods, only a few as-built BrIM for existing bridges have been generated. Time-consuming, labor-intensive and expensive modeling processes from point cloud to BrIM is the main obstacle [15]. Many published studies automated the third step - geometrical modeling and the fourth step - information modeling. Fitting parametric geometry representations like planes and cylinders to point clusters is common to realize geometric modeling [16,17]. Also, for complex shapes which cannot be described by parametric representations, non-parametric representations like polygonal meshes were applied [18,19]. Furthermore, Lu and Brilakis [5] developed a slicing-based approach capable of directly generating

IFC entities from labeled point clusters of bridge components and yielding BrIM. Their method has managed to automate geometrical and information modeling in reinforced concrete (RC) bridges. In fact, automation of geometrical and information modeling techniques has evolved rapidly lately and numerous commercial software applications (e.g., ImageModeler, Leica CloudWorx, PolyWorks Modeler, etc.) have emerged to accomplish such tasks [20]. However, automating the second step, i.e., semantic modeling, remains a challenge. In the case of buildings, although progress has been made in detecting a few specific objects (e.g., pipes [21], structural steels [22], and walls, floors, ceilings, doorways, and windows [23]), no single method can complete the semantic modeling by identifying all required objects. In the case of bridges, the point clouds from different bridge types are usually distinct in geometries and often disturbed by outliers. This difficulty has been amplified by the insufficient training data caused by the high reality capture cost, rendering the difficulty of applying the end-to-end deep learning model. As a result, no research has yet been able to conduct accurate and robust detection of bridge components [15,24,25].

In this paper, we propose a combined local descriptor and machine learning based method to realize automatic semantic segmentation of bridge point cloud. Based on the common geometric features of bridges, we design a multi-scale local geometry descriptor. Considering the point clouds are noisy and disturbed, machine learning is employed to increase robustness in classification. In the end, utilizing the distinguishable distribution pattern of wrongly labeled points, we refine the segmentation results.

The novelty of this method lies in the fact that (1) we develop a multi-scale local descriptor customized for typical bridges which can accurately describe the unique geometries and topology relationships, a distinct improvement from the traditional descriptors often seen in computer science; and (2) we propose a combined local descriptor and machine learning based methodology to fully automatically extract structural elements in bridges on a point level, which can realize small sample learning and achieve high level of accuracy and robustness in the meantime.

We report the state-of-the-art semantic modeling methods in Section 2 and present our approach in Section 3. In Section 4, we elucidate the design of our algorithm. In Section 5, we elaborate on the research methodology and reveal experimental results. The conclusions will be followed in Section 6.

2. State of the art of semantic modeling

The semantic modeling approaches can be classified into algorithm-based, learning-based, and combined algorithm-and-learning-based methods.

2.1 Algorithm-based methods

Algorithm-based methods utilize point cloud processing algorithms and contextual information to detect components.

One way of algorithm-based detection is to perform segmentation and classification in sequence, which is also known as the bottom-up approach, where typical segmentation algorithms in computer science are applied to divide the point cloud into clusters in the first place, and human expertise is then introduced to classify the subsets. During the bottom-up segmentation process, geometric features are extracted from primitive level to higher abstraction levels successively until segmentation is accomplished. Lu et al. [15] pointed out that typical higher-level geometric features included surface normals, meshes, patches, and nonuniform B-spline surfaces. Planar surfaces are the most common geometries in structural components [26]. Algorithms to detect planes in point cloud have been widely studied in computer science (e.g., Hough Transform [27-29], Random Sample Consensus (RANSAC) [30-32], and region growing [33-35]). Primitive shapes like cylinders and spheres can also be detected [31,36]. In the Architecture, Engineering & Construction (AEC) context, numerous works on point cloud segmentation have been carried out. Tarsha-Kurdi et al. modified RANSAC algorithm by introducing a standard deviation threshold to detect roof planes from lidar data [37]. Zhang et al. extracted planar patches from point cloud by adopting spectral clustering in linear subspace and fitting planes to obtained points segments [38]. However, the methods above were incapable of detecting curved surfaces commonly seen in pipes and piers. Patil et al. [39] conducted an area-based adaptive Hough Transform to detect pipes in point cloud data. Combined with the algebraic circle fit algorithm, positions and radius of pipes were estimated. However, RANSAC and Hough Transform are too computationally expensive to detect complex as-built shapes in large datasets of the real world [15,40]. An alternative segmentation approach is region growing algorithms. Nurunnabi et al. [41] selected the point with the lowest curvature as seed points and set the point to plane orthogonal distance, the distance between candidate and seed points, and angles between two points as the criteria of region growing. Yet, noise and outliers could affect

the estimation of normal and curvature and result in over-segmentation. Walsh et al. [42] implemented a smoothness-constraint-based region growing algorithm to find smoothly connected areas in the point cloud. The seed point was extracted by segmenting planar surfaces first, followed by adding similar neighboring points to the segment until reaching the edge. Although this method yielded satisfying results in noisy point clouds, problems of occlusions remained unresolved. In computer science, hole-filling is a common approach to reconstruct occluded surfaces [43-45]. Since furniture and decorations commonly cause heavy occlusions in laser scans of buildings, the ray-tracing algorithm was implemented to detect occlusions [23,46]. Rectangular openings were then identified by training a Support Vector Machine (SVM) with derived features. However, in contrast to the repeated pattern in buildings, occlusions in bridges are scattered and irregular due to vegetation and moving vehicles. For that reason, methods designed to address the occlusion problems in buildings are not applicable for bridge settings. After segmentation, contextual information is introduced by human experts to label obtained point clusters. Pu and Vosselman [47] utilized predefined knowledge about size, position, orientation, and topology to identify façade elements from planar surfaces extracted by region growing. Nevertheless, their method could not be applied to complex structures with curved exteriors or building interiors with serious disturbance.

Another algorithm-based approach is to perform segmentation and classification simultaneously in a top-down manner, where knowledge of semantic classes is usually utilized as the criteria for segmentation. Priori knowledge of geometric features and design regulations can be one key reference to perform the top-down semantic segmentation. Sanchez and Zakhor [48] divided cloud points into ceiling points, floor points, wall points, and stair points according to their orientations. However, taking orientation as the only criteria is not robust since noise would affect orientation estimation. The other valuable knowledge source is the as-designed information of existing structures (e.g., drawings [49] and BIM [22,50,51]). Using design drawings as supplemented information, Liu et al. [49] extracted building elements and heating, ventilation, and air-conditioning (HVAC) systems from the point cloud to construct the as-built BIM. Nonetheless, deviations in the size and location of components between as-designed documents and the as-built status might lead to inaccuracy in the reverse modeling. Turkan et al. [51] registered the as-designed 4D BIM and 3D laser scan data in the same coordinate system, succeeding in recognizing as-built objects and estimating construction progress. Although adopting as-designed information for

semantic modeling is a straightforward and reliable approach, frequent lack of as-designed documentation or large deviations between as-designed and as-is status could limit its application scope.

In bridges, features of components highly differ from those in buildings. A limited number of works specifically applied for bridges have been conducted. Riveiro et al. [52] detected components of masonry bridges based on normal orientation histograms and morphological information. However, in real-world bridges with traffics and surrounding vegetation, unavoidable data quality problems like noise, low point density, and occlusions would sabotage the outcome. Later on, based on histograms of coordinates and distinct elevation of components, Lu et al. [15] employed a slicing algorithm to identify structural components in point clouds of existing RC bridges. But this method was only applicable to straight RC bridges supported by piers with the same height, whereas numerous RC bridges do not meet this standard. Yan and Hajjar [25] adopted a similar approach to extract structural elements of steel girder bridges. Likewise, Zhao et al. [53] separated superstructures and substructures of bridges according to density projection images. Then, a fine segmentation was performed based on histograms and continuity. However, existing top-down methods in semantic modeling of bridges require manually extracting bridges from the background point cloud in advance and have substantial restrictions on the geometries of bridges.

2.2 Learning-based methods

In this paper, learning-based methods refer to models that directly operate on raw 3D data and only depend on learning. No processing related to transcendental knowledge on geometries or design standards is required. In recent years, numerous works have been carried out in end-to-end learning-based semantic segmentation of point cloud, among which deep learning has achieved remarkable results. The deep learning methods have been demonstrated to be a very promising method due to their fully automated nature and generic applicability to all objects. In general, these deep learning models can be divided into three categories, i.e., 3D convolution on voxels [54], multi-view projection onto images or planes followed by 2D convolution [55] and PointNet [56,57]. PointNet is the state-of-the-art model to deploy deep learning on points directly, which usually contains three parts, namely, a sharing multi-layer perceptron (MLP) to extract point features, a max-pooling layer to aggregate point features, and an MLP for object classification. Since PointNet

has addressed the problems associated with permutation invariance and sparsity of points, this method has been widely adopted as a popular approach to extract geometric features from point cloud for purposes like object detection [58]. PointNet++ [57] is a more sophisticated model based on the backbone of PointNet, where a hierarchical feature learning method is applied and some units proven to be redundant are removed. Specifically, PointNet is recursively applied on a nested partitioning of the point cloud, forming a hierarchical neural network. Through varying metric space distances progressively, local features with increasing contextual scales can be learned in PointNet++. Also, the transformer net (T-net) in PointNet, designed for learning an affine transformation matrix by its own mini network, is abandoned because of its indistinctive effect. The most significant improvement in PointNet++ is the capability of capturing local structures and global features in the point cloud simultaneously. Many other deep learning models have also been developed, such as PointCNN [59] and Dynamic Graph Convolutional Neural Network (DGCNN) [60]. However, most works in computer science have initially been designed for automatic drive scenarios or indoor objects. Implementing end-to-end deep learning models to detect outdoor structural components has not been well explored. Kim et al. [24] detected piers and decks of full-scale RC bridges from the point cloud using PointNet. In order to preserve the spatial resolution of point clouds under limited video card memory, points were divided into small subspaces before training, so global features were lost. Also, because of insufficient training data, only piers and decks were detected in their work. Kim and Kim [61] compared the performance of PointCNN, DGCNN, and PointNet in semantic segmentation of bridge point clouds. They trained and tested the deep learning models on the same bridges by simply dividing each bridge into two halves. The high similarity between training and testing data could make the results controversial. In summary, implementing end-to-end deep learning methods to large-scale bridges is severely restricted by insufficient training data and high computational burden.

2.3 Algorithm-and-learning-based methods

To yield satisfying results with limited training data and computational power, algorithms can be employed to extract high-level manually defined features or to generate geometry primitives. After that, learning-based models (e.g., logistic regression, Naive Bayes, decision tree, support vector machines, random forest, and neural networks) can be trained to classify the components.

Xiong et al. [23] leveraged a region growing algorithm to connect points with similar surface normals, generating planar patches from the voxelized point data. Then, according to the Euclidean distance, they identified the four nearest neighbors of each patch, based on which the predefined contextual information of each patch was calculated. Meanwhile, the local feature of each patch was defined by a combination of its orientation, shape, size, point density, and height. Finally, a stacked learning model [62] was trained to classify the planar surfaces into walls, ceilings, floors, and clutters. Although this work was robust to occlusions in the indoor environment, it only output planar surface models and could not detect structural components encompassed by curved surfaces. Koppula et al. [63] took both image-based features (e.g., color, texture, and gradient of intensities) and point cloud features (e.g., shape, orientation, and geometrical context) from the data captured by the RGB-D sensor as the input to train various learning algorithms, aiming to detect floors, walls, and other non-structural objects. However, the precision of this method turned out to be low in their tests. In bridge settings, Zhang et al. [64,76]] started the semantic modeling by extracting surface primitives(e.g., cuboids, cylinders, and sheets) from the point cloud. A decision tree was then applied using the type, parametric model, orientation, and mutual geometric relations of the fitted surface primitives as decision variables, succeeding in detecting structural components such as columns, caps, deck, beams, etc. Yet, the scope of this work was limited to bridges composed of simply shaped elements. In engineering practices, deviations are inevitable between as-designed models and as-built structures due to construction errors. Using parametric primitives to describe the geometries of as-built components could compromise the fidelity of as-built models, which is contradictory to the original purpose of creating as-is BIM to reveal the real status of existing structures. In conclusion, even though the hybrid method yielded promising results with limited training data, no work has yet been capable of performing accurate point-wise classification on the point cloud of real-world RC bridges, which, however, is necessary if we need to describe non-parametric components like curved decks and I-section beams in bridges with high fidelity.

2.4 Identified knowledge gaps

The bottom-up algorithm-based methods work well in synthetic data. However, they are highly sensitive to noise and occlusions in the real-world environment since they largely depend on accurate partitioning of surface primitives and patches during the segmentation stage. And in the

classification stage, manually defining rules to classify the obtained subsets is unrealistic, especially in complex structures. The top-down algorithm-based approaches rely on semantic information of objects, i.e., as-designed documents or engineering knowledge. However, lack of as-designed information or deviations between as-designed model and as-is structure have rendered providing accurate as-built information unpractical. And the utilization of engineering knowledge relies on clean point clouds, which often requires manual removal of background. Also, strict limitations on the geometries of bridges (e.g., piers with the same height and no curvature in decks) make this type of method unfeasible to many real bridge cases.

End-to-end learning-based models have great potential to become versatile solutions, but their applications are still highly limited due to lacking in training data or computing power. Although combining algorithm-based and learning-based methods can alleviate such needs, the current work is only restricted to parametric geometrical primitives and yet applicable to real and complex surfaces. In summary, with limited training data and computing resources, automated semantic segmentation of real-world point cloud data of RC bridges with high fidelity has yet to be solved.

3. Methodology

To overcome the challenge of insufficient training data and achieve versatile semantic segmentation, we propose an algorithm-and-learning-based approach utilizing local descriptors for geometric feature extraction and machine learning for point-wise classification. The objectives of this research include (1) designing a novel local descriptor capable of describing the geometric features of bridges, and (2) developing an optimized machine learning pipeline to automatically assign semantic labels to points.

The topological relationship and geometric features of bridges are relatively distinguishable and straightforward compared to buildings. Aiming at conducting point-wise segmentation, local descriptors can be used to describe the geometric features at each point. In this approach, we utilize the distribution pattern of neighboring points' normals as the descriptor to represent the area's geometric features surrounding the point of interest. The angle between the normal vector and vertical direction has been chosen to represent the orientation of normals.

Descriptors are prone to noise interference and are difficult to be used to classify objects

directly, so machine learning is considered to overcome such challenges in this method. Common machine learning models include Naive Bayes, logistic regression, nearest neighbors, decision trees, support vector machines, and artificial neural networks (ANN). Bridge point cloud data has the characteristics of large sample size, high dimensionality, unbalanced categories, and much noise. Although Naive Bayes has stable performance and strong interpretability, our descriptor does not meet its requirement of independent sample attributes [65]. Logistic regression is efficient and straightforward, but it performs poorly when dealing with high-dimensional data [65]. Decision trees and the nearest neighbor algorithm have high flexibility and simple theory, but it is difficult to effectively classify data with unbalanced categories [66]. ANN has been successful in unstructured data processing such as computer vision and natural language processing. It has the advantages of high classification accuracy, strong ability to fit complex nonlinear relationships, and robustness to noise [67]. The advantages of neural networks have also made them widely used in point cloud processing. For example, the most famous point cloud deep learning model PointNet[56] uses multiple ANN for feature learning and classification. Therefore, this study chooses ANN to classify the local feature descriptors of the point cloud.

Since the proposed approach requires no manual data preprocessing, disturbances (e.g., vegetations, handrails, and traffic cones) and noises from moving vehicles can affect the outcome of segmentation. Thus, segmented structural components should be further refined to eliminate such detrimental effects. Tailored for the scattered distribution pattern of incorrectly labeled points, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [68] is employed to filter the noise and scattered small clusters. DBSCAN is a query-based data clustering algorithm concerning point density. It groups together points that are closely packed and identify the isolated points as noise. DBSCAN has been chosen for two advantages, i.e., no needs to specify the number of clusters a priori and the ability to identify clusters of arbitrary shapes.

The workflow of our method contains three steps (see Fig. 1), namely, computing designed descriptors, point-wise classification through neural networks, and results refinement. The first step defines a multi-scale descriptor (i.e., descriptors with the perceptual fields of different sizes) for each point, which can be utilized as the classification criteria. The second step is to classify these points with ANN. The third step includes identifying and correcting wrongly labeled points in structural elements.

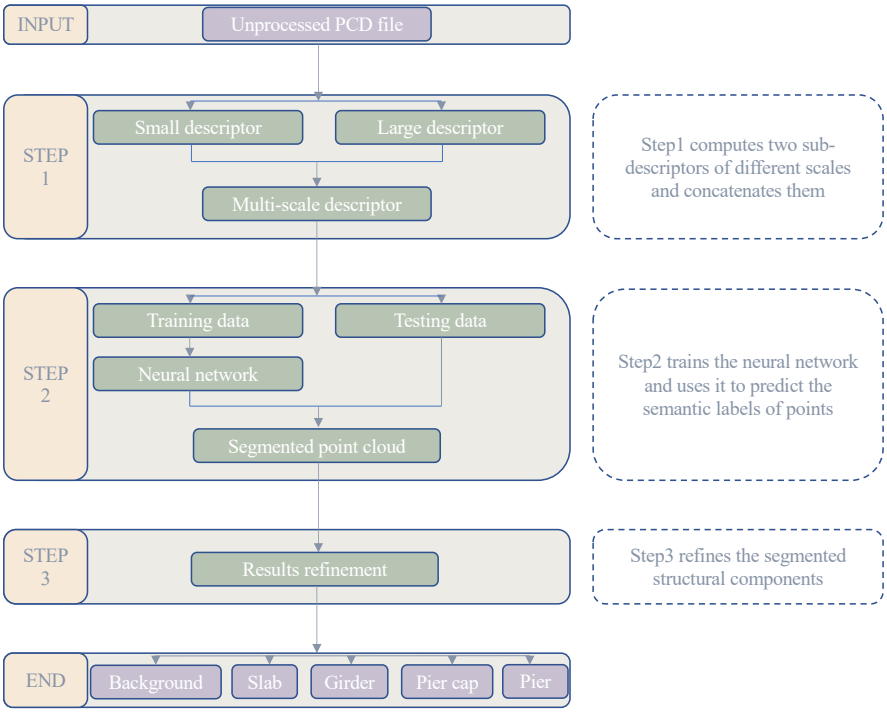


Fig. 1. Workflow of semantic segmentation system.

This paper mainly focuses on optimizing the semantic segmentation of RC slab and beam-slab bridges because those are the most common bridge types in highways, city expressways, and other overpasses. We will identify typical structural components, including slabs, girders, pier caps, and piers. Apart from structural components, all other objects (e.g., vegetation, vehicles, pedestrians, handrails, noise, and other irrelevant items) in the point cloud are classified as background.

4. Algorithm design

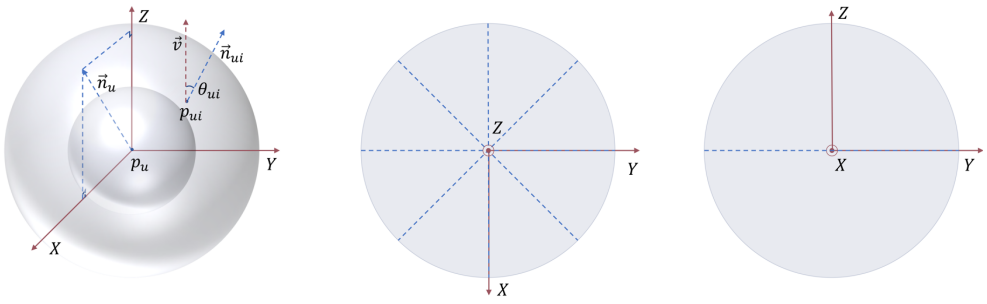
4.1 Step1 - computing designed descriptors

To structure the defined features, we implement the encoding method of the Signature of Histograms of OrienTations (SHOT) [69], which is usually used for the registration of point clouds. Tailoring for the semantic segmentation task in bridge settings, we define the content of our descriptor in the following paragraph.

$\boldsymbol{p_u}$ is the $\boldsymbol{u_{th}}$ target point in the point cloud, and $\boldsymbol{p_{ui}}$ is the $\boldsymbol{i_{th}}$ neighboring point of the target point $\boldsymbol{p_u}$. We choose the angle, θ_{ui}, between the normal at each neighboring point, $\boldsymbol{n_{ui}}$, within the target point's perceptual field, R_u, and the vertical direction, $\boldsymbol{v}$, to represent the direction of normal and describe the geometric feature (Fig. 2 (a)). For computational efficiency, $\cos\theta_{ui}$ is chosen to

reveal the value of θ_{ui}, since $\cos\theta_{ui} = \boldsymbol{n_{ui}}.\boldsymbol{v}$. Considering site scans of bridges are usually noisy, histogram is selected as the feature encoder in this research because of its excellent performance in filtering noise effects [70]. Each histogram has 11 bins, and neighboring point counts are accumulated to these bins according to $\cos\theta_{ui}$.

To reveal the spatial relationships of surfaces and components around the target point, we introduce the local reference system (LRF) and spherical grid to store the geometric features corresponding to neighboring points' locations. When setting the LRF, two factors are considered, namely, (1) the local reference system should be rotational invariant in the horizontal plane since horizontal orientations of bridges are not fixed; (2) the primary topological relationships between surfaces and components are vertically stacked or horizontally sided. Thus, the X-axis of LRF is along with the horizontal component of the normal at the target point, n_u, and the Z-axis is parallel to the vertical direction. Naturally, the Y-axis is the cross product of the Z-axis and X-axis (Fig. 2 (a)). In addition, to clearly distinguish different locations with limited computing resources, the sphere grids encompass 2 radius divisions (Fig. 2 (a)), 8 azimuth divisions (Fig. 2 (b)), and 2 elevation divisions (Fig. 2 (c)), which results in 32 volumes in total.



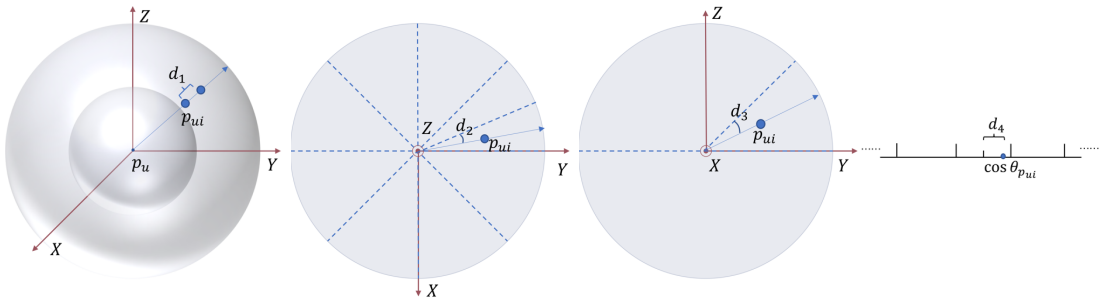
(a) (b) (c)

Fig. 2. Illustrative diagrams of descriptors (a) perspective view of descriptor; (b) top view of descriptor; (c) front view of descriptor.

When using histograms, the boundary effect could affect the result. We conduct a quadrilinear interpolation for each point count to eliminate the boundary effect. Specifically, we multiply a weight factor $1 - d$ along every dimension, including radius, azimuth, elevation, and histogram.

$$n = \prod_{i=1}^{4}(1 - d_i) \quad (1)$$

In radius (Fig. 3 (a)), azimuth (Fig. 3 (b)) and elevation (Fig. 3 (c)) dimension, d is the angular or Euclidean distance between the neighboring point and the volume's center. In histogram dimension (Fig. 3 (d)), d is the difference between the neighboring point's cosine value and the bin's central value. Note that d in every dimension has been normalized by the size of the volume or the bin.



(a) (b) (c) (d)

Fig. 3. Linear interpolation on (a) radius; (b) azimuth; (c) elevation; (d) histogram.

The sub-descriptors will be normalized by L2 norm (i.e., the square root of the sum of the squared vector values) to address the non-uniform point density issue.

Following the above-mentioned steps, two sub-descriptors of different perceptual fields will be calculated. The first sub-descriptor will describe the distribution of close neighbors' orientations (Fig. 4 (a)). In this way, points on the boundaries are easy to separate. However, lacking global information will make points in the interior of components hard to distinguish since they are most likely on similar planes. Thus, the second sub-descriptor will reserve geometric features and topological relationships in a larger perceptual field (Fig. 4 (b)).

(a) (b)

Fig. 4. Multi-scale sub-descriptors (a) small scale sub-descriptor; (b) large scale sub-descriptor.

4.2 Step2 - point-wise classification through the neural network

ANN is composed of neurons aligned in the same layer and connected to neighboring layers. Neuron is the primary function unit of ANN, which usually contains a linear function and a nonlinear

function (activation function). The linear function is the summation of the weighted inputs and bias, mathematically denoted as in Eq. (2):

$$S_i = \Sigma w_{ij} x_j + b_i \tag{2}$$

For the i_{th} neuron in the current layer, S_i is the result of the linear function, w_{ij} is the weight of the j_{th} input, x_j is the value of the j_{th} input, b_i is the bias. The sum is then passed through activation function to produce the output. We choose the Rectified Linear Unit (ReLU) as the activation function in this research, same as PointNet [56], since ReLU converges faster and avoids gradient vanishing [71]. It has the following form (3):

$$O_i = f(S_i) = \max(0, S_i) \tag{3}$$

O_i is the output of this neuron, f is the activation function (ReLU). The outputs of neurons will be passed through connections and become the input of neurons in the next layer. The form of the n_{th} neuron in the next layer is illustrated in Eq. (4), where O_n is the output of the neuron, w_{ni} is the weight of the i_{th} input, b_n is the bias:

$$O_n = f(\Sigma w_{ni} f\big(\Sigma w_{ij} x_j + b_i\big) + b_n) \tag{4}$$

For the output layer, we choose Softmax as the nonlinear function. Through Softmax, we can realize that the values of final outputs are between 0 and 1and the sum of final outputs is 1. When we set the number of neurons in the output layer equal to the number of categories, we can acquire each label's probability. It is shown in Eq. (5):

$$Softmax(t_i) = \frac{e^{t_i}}{\Sigma e^{t_i}} \tag{5}$$

Here, t_i is the result of the linear function of the i_{th} neuron in the output layer.

The structure of the neural network in this research is presented in Fig. 5. The numbers of hidden layers and their neurons are decided based on two factors, i.e., the Universal Approximation Theorem [72] states that a neural network with one hidden layer containing a sufficient but finite number of neurons can approximate any continuous function to a reasonable accuracy; the required computational resources for training the neural network increase with the number of hidden layers and neurons. Thus, we set the structure of hidden layers as Fig. 5, which has 6 hidden layers containing a decreasing number of neurons from 1024 to 32. In addition, the number of inputs is the dimension of the descriptor, and the number of outputs corresponds to the number of categories.

Furthermore, to avoid over-fitting, every liner function in hidden layers is followed by batch normalization (BN). Since the neural network is quite deep, 50% dropout [73] is conducted in the last hidden layer to prevent over-fitting, taking care of the problems associated with noise and high correlation in the feature space. Such dropout technique is conducted by randomly dropping neurons as well as their connections during training.

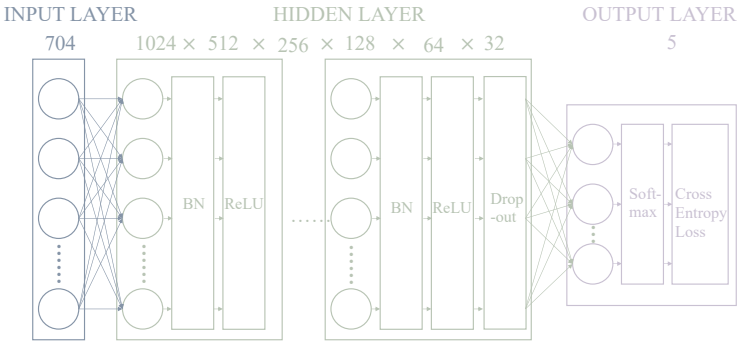


Fig. 5. Structure of the neural network.

After the neural network is constructed, we train the neural network with labeled data. We use gradient descent as the optimization method since it converges fast in large datasets [74]. In general, it iteratively optimizes weights and biases through the partial derivatives of loss function. A common training technique, namely, the backpropagation algorithm[75] is implemented in this research. The loss function in this paper is the Cross Entropy Loss, which is usually used to describe the loss of classification result. The formulations are defined in Eq. (6) and Eq. (7):

$$l(x,y) = \Sigma_{n=1}^{N} \frac{1}{\Sigma_{n=1}^{N} w_{y_n}} l_n \tag{6}$$

$$l_n = -\omega_{y_n} log x_n y_n \tag{7}$$

Here, x is the input, y is the target, ω is the weight, N is the batch size. Weights are usually set to deal with unbalanced training datasets.

4.3 Step3 – results refinement

As shown in Fig. 6 (a), the incorrectly described points are scattered and sparse. To amend the wrongly labeled points, we start with a point clustering using DBSCAN and then filter the noise and scattered small clusters.

The process of DBSCAN is illustrated in Fig. 7. DBSCAN begins with a query on an arbitrary point. If the point's ε-neighborhood contains less than $minPts$ points (ε(eps) and $minPts$ are the

predefined radius and the minimal number of neighboring points, respectively), it will be identified as noise. Query on this point will be terminated and passed to another arbitrary unvisited point. In contrast, if the queried point has sufficient neighbors, it will be decided as core point and start a cluster. Points in its ε-neighborhood will be considered part of the cluster and queried. If a neighboring point is also identified as core point, the clustering process will be extended to it, and a new query cycle will start. On the contrary, it will be identified as border point, and this thread will end. This cluster will continue to grow until all newly added points are border points. Then, another unvisited point will be selected, and the above process will be conducted again. The clustering process will finish when all points have been visited.

Through DBSCAN, every point is either assigned to a cluster or identified as noise (Fig. 6 (b)). Since incorrectly described points are scattered, we can resume that noise points and clusters with points less than a threshold were wrongly labeled and should be labeled as background (Fig. 6 (c)).

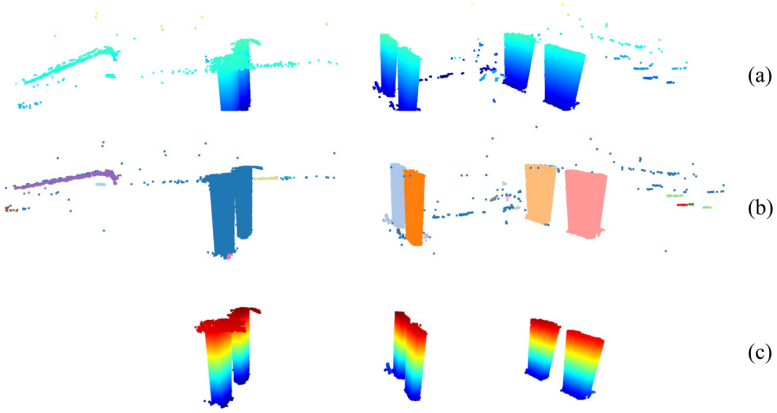


Fig. 6. Results refinement (a) piers from neural network; (b) piers after DBSCAN; (c) piers after filtering.

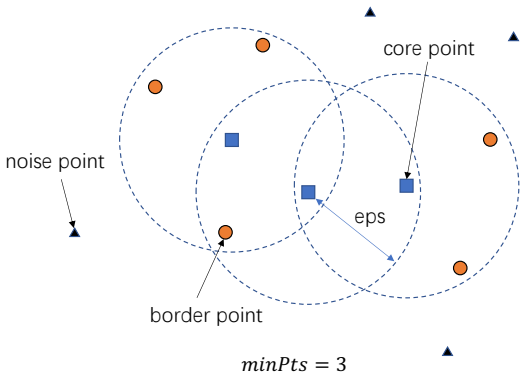


Fig. 7. Process of DBSCAN.

5. Validation

5.1 Data preparation

We validated our algorithm on two datasets. The first dataset is the open-source data provided by Lu et al. [15]. The raw point cloud data can be downloaded on the openly accessible website (https://zenodo.org/record/1240534). The dataset contains ten highway bridges around Cambridge shire, UK. Since three bridges in the dataset each belongs to a different bridge type and thus cannot be used for training and testing at the same time, they are excluded in this experiment. Thus, a total number of 7 bridges are included in the first dataset. We collected the second dataset using Leica P40 Terrestrial Laser Scanner (ranging error 3mm at 50m, angular accuracy 8”), including four highway bridges located in Shanghai, China. The point density was set to 3.1mm over 10m. Note that all point cloud data have been horizontally leveled. And due to memory limitation, we have conducted a voxel down-sampling with a size of 4cm.

Since the structural components of different bridges have various geometric shapes, the similarity between training data and validation data must be considered in data splitting. Thus, the splitting of dataset 1 and dataset 2 is presented in Fig. 8 and Fig. 9.

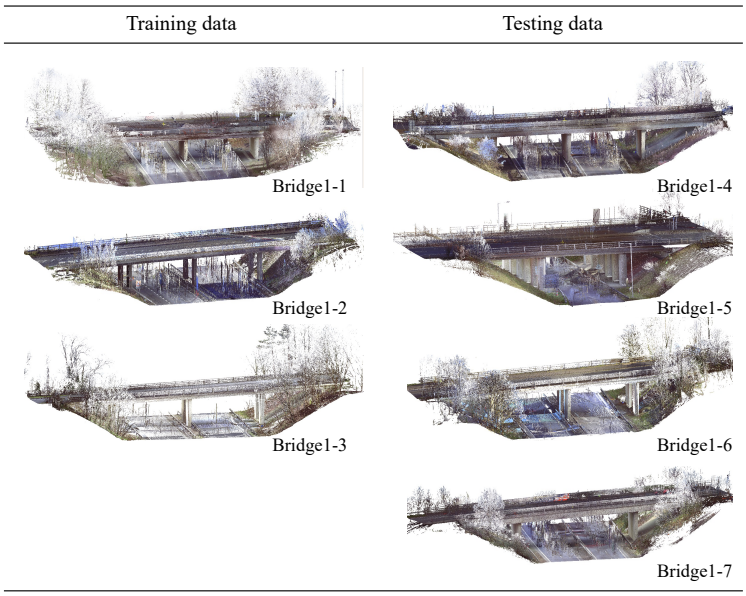


Fig. 8. Splitting of dataset 1.

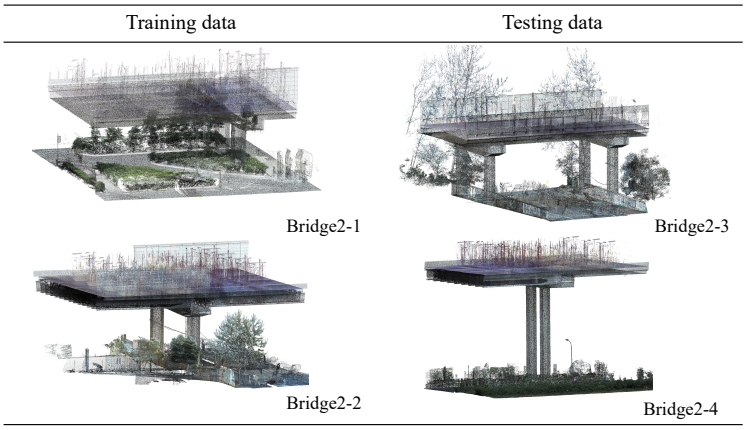


Fig. 9. Splitting of dataset 2.

5.2 Implementation

The semantic segmentation system was implemented in C++ and Python based on the proposed framework. The computing environment is Intel Core i9-9900k processor clocked at 3.60 GHz, 16 GB of RAM, and a NVIDIA GeForce GTX 1080ti. Parameters in the experiment are presented in Table 1.

Table 1
Parameter settings in the experiment.

Output layer dimension	Weights	Batch size	ε	$minPts$	Minimal points number
5	(1,10,40,100,40)	262144	0.4m	10	10000

The output layer dimension is equal to the number of semantic categories, i.e., 5. The number of points in each category in the dataset is highly unbalanced. The ratio of points in the background, slabs, girders, pier caps, and piers is approximately 40:10:2:1:2. Considering that recognition of structural components is more important, the weights of their labels in training are set to (1, 10, 40, 100, 40). Since the data sequence has been shuffled before training, only the training speed and video memory should be considered when setting the batch size. When using DBSCAN to optimize the results, ε and $minPts$ depend on the point density. The minimum number of points in a point cluster depends on both the point density and the size of target components. The point clusters failing to meet the threshold will be classified as background.

5.3 Results

To quantitatively evaluate the performance of our algorithm, we use three criteria, i.e.,

Precision (Pr), Recall (R), and Intersection over Union (IoU).

$$Pr = \frac{TP}{TP+FP} \tag{8}$$

$$R = \frac{TP}{TP+FN} \tag{9}$$

$$IoU = \frac{TP}{TP+FP+FN} \tag{10}$$

Where TP(True Positive) is the number of correctly labeled points, FP(False Positive) is the number of wrongly labeled points, FN(False Negative) is the number of miss-labeled points.

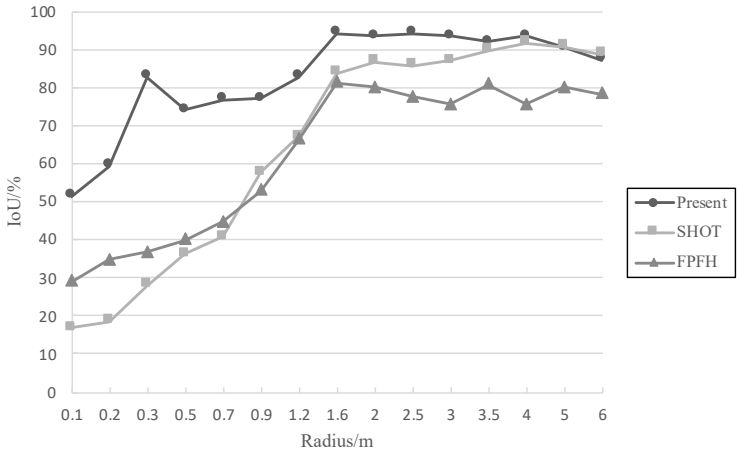


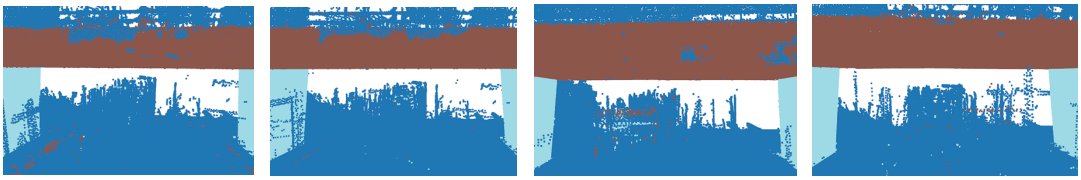
Fig. 10. IoU of semantic segmentation of descriptors with different scales.

Under different descriptor radii, the IoU of bridge semantic segmentation is shown in Fig. 10. Compared to SHOT and Fast Point Feature Histograms (FPFH), our descriptors have better performance in the semantic segmentation of bridges. This proves that our descriptor has succeeded in accurately describing the geometric features of bridges. Especially when the descriptor radius is small, the IoU of our descriptor is substantially larger than the other two. This indicates our descriptor has superior local geometric feature abstracting capability.

As the descriptor radius increases, IoU shows an ascending and then descending trend. The reason for the increase in IoU is that as the radius of the descriptor increases, topological relationships on a larger scale are revealed in the descriptor, making the segmentation results more accurate. The reason for the decrease of IoU is that after the descriptor radius exceeds the maximum thickness of the component, continuing to increase the descriptor size has no more positive influence on the expression of geometric features but will make the descriptors less distinguishable. Besides, the interference from the environmental point cloud will also increase, worsening the semantic segmentation results.

When the descriptor radius is 0.3 meters, IoU reaches the local maximum. At this point, the descriptor provides a balance between accuracy and robustness. Due to the lack of topological relationship, the interior points of the component cannot be accurately classified though (Fig. 11 (a)). When the descriptor radius is less than 0.3 meters, noise, measurement errors, and density changes will affect the calculation of the descriptor; when the descriptor radius is slightly greater than 0.3 meters, the descriptor will be interfered with by adjacent component points and unable to describe the local geometry accurately. When the radius reaches 1.6 meters, IoU starts to reach the maximum value, which indicates that the descriptor can include the global geometric shape of the individual components and the topological relationship between different components. The interior points can be correctly labeled. Nevertheless, the segmentation of the boundary (Fig. 11 (b)) is not as accurate as of that of the small-scale descriptor (Fig. 11 (a)). When the radius increases to 3.5m, the points on the boundaries begin to be massively misclassified because of the increasing interference effect of the environmental point cloud. (Fig. 11 (c)).

Therefore, to fully represent the geometric features and improve the segmentation accuracy, this study combines the descriptors with radii of 0.3 m and 1.6 m into multi-scale descriptors. We can see that both the boundary and interior points of components can be accurately classified (Fig. 11 (d)). And the IoU increases from 94.31% to 96.69%.



(a) (b) (c) (d)

Fig. 11. Segmentation results of descriptor radius (a) 0.3m; (b) 1.6m; (c) 3.5m; (d) 0.3m and 1.6m.

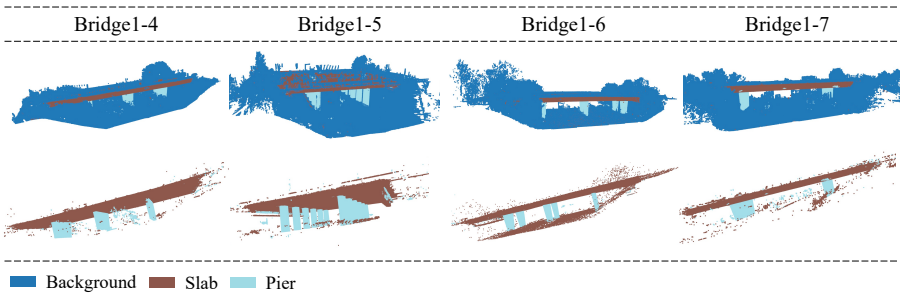


Fig. 12. Semantic segmentation results from the neural network of dataset 1.

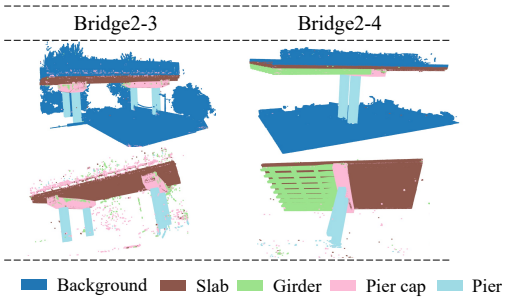


Fig. 13. Semantic segmentation results from the neural network of dataset 2.

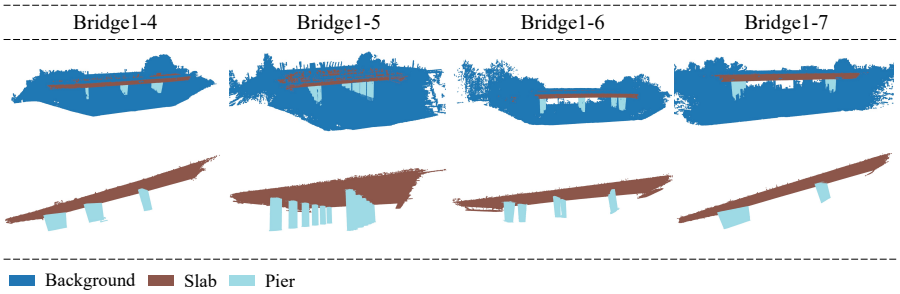


Fig. 14. Semantic segmentation results after refinement of dataset 1.

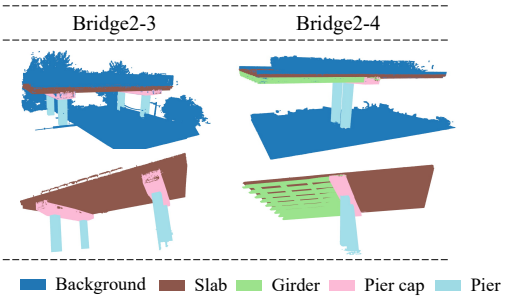


Fig. 15. Semantic segmentation results after refinement of dataset 2.

The semantic segmentation results from the neural network of dataset 1 and dataset 2 are shown in Fig. 12 and Fig. 13, respectively. And the results after refinement by DBSCAN are shown in Fig. 14 and Fig. 15. From Fig. 12 and Fig. 13, we can see that the neural network has already performed an accurate classification for most points, while a few wrongly labeled points are distributed in a scattered and sparse pattern. Even though they seem to be dense from the present angle of view in some bridges (e.g., Bridge1-6), they are actually quite sparse that they are almost invisible in the overall point cloud. This can also be verified by the statistics in the quantitative analysis below. Comparing the above figures before and after the refinement, we can find that the wrongly labeled points have been effectively removed by Step 3.

It should be noted that Bridge1-1 and Bridge1-4, Bridge1-2 and Bridge1-5, Bridge1-3 and Bridge1-6, Bridge 2-1 and Bridge2-3, Bridge2-2 and Bridge2-4 are similar in geometries in pairs

514 respectively, one for training and the other for testing. As shown in the refined results (Fig. 14 and
515 Fig. 15), even when only one bridge of the same type is used for training, vegetation, railings, ground
516 road surfaces, scanning noise, and other interference objects are accurately identified as background,
517 and bridge components can be precisely detected and segmented. Therefore, the algorithm has
518 achieved satisfying results by small sample learning. It is interesting to note that Bridge1-7 has no
519 bridge of the same type in the training set, but only the same type of components, and it is still
520 accurately identified. Therefore, this method learns geometric features on component level, and the
521 training results can be extended to bridges with different combinations of trained components. This
522 has demonstrated that the present method has good versatility. Another strength of this method is
523 that it can accurately classify the points on the boundaries of different components. Compared with
524 the deep learning method that is difficult to identify the boundary points on an individual basis, this
525 method can significantly improve the level of accuracy in geometrical modeling. It thus can greatly
526 improve the usability of the semantic point cloud model.

527 **Table 2**

528 Quantitative analysis of semantic segmentation results from the neural network of dataset 1.

	Precision			Recall			IoU		
	Slab	Pier	Avg.	Slab	Pier	Avg.	Slab	Pier	Avg.
Bridge1-4	97.09%	97.98%	97.24%	99.55%	98.73%	99.41%	96.67%	96.76%	96.69%
Bridge1-5	93.36%	90.12%	92.66%	94.48%	98.52%	95.31%	88.54%	88.92%	88.62%
Bridge1-6	92.05%	97.22%	92.86%	98.42%	98.97%	98.51%	90.71%	96.25%	91.58%
Bridge1-7	98.12%	95.48%	97.74%	97.87%	99.80%	98.13%	96.06%	95.30%	95.96%
Avg.	95.16%	95.20%	95.13%	97.58%	99.01%	97.84%	93.00%	94.31%	93.21%

529 **Table 3**

530 Quantitative analysis of semantic segmentation results after refinement of dataset 1.

	Precision			Recall			IoU		
	Slab	Pier	Avg.	Slab	Pier	Avg.	Slab	Pier	Avg.
Bridge1-4	97.32%	99.16%	97.62%	99.55%	98.73%	99.41%	96.89%	97.90%	97.06%
Bridge1-5	95.36%	98.72%	96.05%	94.45%	98.25%	95.29%	90.30%	97.28%	91.69%
Bridge1-6	93.86%	98.28%	94.56%	98.42%	98.97%	98.51%	92.47%	97.28%	93.22%
Bridge1-7	98.82%	98.17%	98.73%	97.87%	99.79%	98.13%	96.74%	97.98%	96.91%
Avg.	96.34%	98.58%	96.74%	97.57%	98.94%	97.84%	94.10%	97.61%	94.72%

531 The quantitative evaluation of segmentation results from the neural network is shown in Table
532 2 and Table A.1(Appendix A). And the results after refinement are shown in Table 3 and Table
533 A.2(Appendix A). Comparing the results, Pr and IoU of semantic segmentation are slightly
534 improved through Step 3, while R remains basically unchanged. The reason is that the refinement

algorithm only reduces FP by re-labeling identified noise and small clusters from the structural element to background. As illustrated in Eqs. (8)(9)(10), decreasing FP only has a positive effect on improving Pr and IoU. Finally, the mean precision, recall, and IoU of all testing bridges in dataset 1 and dataset 2 are calculated as 97.26%, 98.00%, and 95.38%, respectively, indicating that the proposed method can realize high-precision automatic semantic segmentation for RC slab and beam-slab bridges.

Our approach also has advantages on applicability as compared with other existing methods in Table 4. Although the algorithm-based method proposed by Lu et al.[15] was quite accurate, it needed manual extraction of main structures from environmental point cloud before applying the segmentation algorithm. It also could not deal with bridges with piers of varying heights, which, however, are common in RC bridges. The learning-based method proposed by Kim et al. [24] also realized automatic semantic segmentation. However, even their training datasets (4 training bridges and 3 testing bridges) were larger than ours, they only coarsely segmented the bridges into slabs and piers. Zhang's [76] approach successfully recognized common components in RC bridges, but it only output simple geometric primitives. From the above comparison, it can be seen that our approach has fulfilled both the accuracy and feasibility requirement.

Table 4

Comparison of state-of-the-art methods for bridge semantic modeling.

References	Methodology	Fully Automatic	Recognition Capability					Point-wise
			Slab	Girder	Pier cap	Pier	Background	
[15]	Algorithm-based	No	**Yes**	**Yes**	**Yes**	**Yes**	No	**Yes**
[24]	Learning-based	**Yes**	**Yes**	No	No	**Yes**	**Yes**	**Yes**
[76]	Algorithm-and-learning-based	No	**Yes**	**Yes**	**Yes**	**Yes**	No	No
Ours	Algorithm-and-learning-based	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

To compare the performance of the proposed method with end-to-end learning-based models, we implement PointNet on dataset 1. The parameters are set according to the work of Kim et al. [24]. The results are presented in Fig. 16 and Table 5. We can see that PointNet performs notably poorly. For instance, the boundaries between objects are not clearly segmented, which has the chain effect in the subsequent geometric modeling process. The quantitative analysis in Table 3 and Table 5 confirms that our method can accomplish more accurate semantic segmentation with small

learning sample compared to deep learning model. It can also be seen that Bridge1-7 is substantially poorly segmented by PointNet, which confirms the claim that our approach is more versatile to bridges with varying combinations of components. Furthermore, unlike the local descriptor, PointNet will fail to recognize the same object if it rotates. Data augmentation is required to solve this problem, which will consume considerable computing resources. Besides, for large-scale scenes, we need to divide the point cloud into partitions before training the PointNet. Since the maximum number of points in each partition is limited by video memory, the spatial resolution of the point cloud will be compromised to reserve the global context. The proposed combined local descriptor and machine learning approach only concerns local regions and thus has the evident advantage when high precision semantic models are demanded.

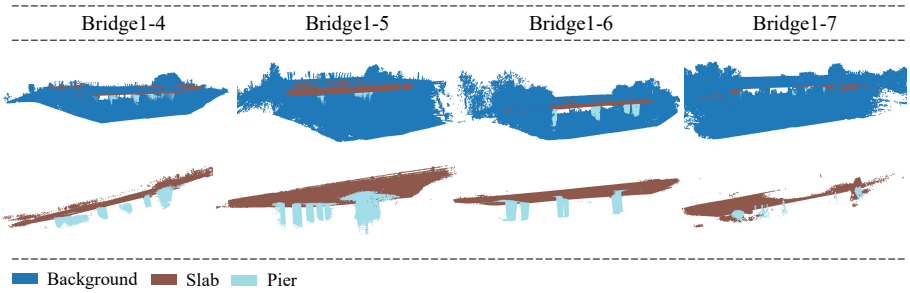


Fig. 16. Semantic segmentation results of dataset 1 by PointNet.

Table 5

Quantitative analysis of semantic segmentation results of dataset 1 by PointNet.

	Precision			Recall			IoU		
	Slab	Pier	Avg.	Slab	Pier	Avg.	Slab	Pier	Avg.
Bridge1-4	84.73%	58.23%	80.89%	41.42%	23.49%	38.37%	38.55%	20.10%	35.18%
Bridge1-5	96.86%	53.57%	86.50%	59.24%	40.09%	55.32%	58.12%	29.75%	50.92%
Bridge1-6	96.72%	90.29%	95.54%	87.66%	93.77%	88.67%	85.14%	85.18%	85.14%
Bridge1-7	74.42%	76.17%	74.47%	14.62%	2.68%	12.97%	13.92%	2.66%	12.42%
Avg.	88.18%	69.57%	84.35%	50.74%	40.01%	48.83%	48.93%	34.42%	45.92%

5.4 Discussion

Since the proposed combined local descriptor and machine learning based method employs surface normals of points to represent geometric features, the level of precision on estimating the surface normals can heavily affect the performance of this method. This brings about a strict request to the quality of point cloud data, including spatial resolution, precision, and the level of completeness. The impact of these three factors will be quantitatively analyzed in the following paragraphs.

From Table 3, we can see that the segmentation accuracy of Bridge 1-5 is poor. The primary reason is that the normal vector and descriptor estimation of points in the interior of slabs are inaccurate due to low point density. To examine the spatial resolution required for normal vector and descriptor calculations, we use different voxel down-sampling distances to simulate point clouds with different densities. The segmentation results are shown in Fig. 17. When the minimum distance between the points reaches 4 cm, the IoU reaches the maximum value. If the spatial resolution is inadequate, the calculation of the normal vector and the descriptor tends to be inaccurate. Therefore, a spatial resolution of 4 cm is chosen to meet the requirements of descriptor-based algorithms in this approach. It is worth noting that although our approach is rather robust despite the point density variation in point cloud data, it may present problems in the event of very sparse points. This is related to the scale we set for the determination of a point's neighborhood. When the spatial density decreases dramatically at some points, the number of points enclosed might not be adequate to conduct eigenvalue decomposition and estimate the surface normal. The low point density in certain local regions, however, frequently happens in the point cloud of bridges since long scanning distances or large angles of reflection are often demanded to capture some unreachable locations (e.g., girders and road surfaces on the bridge). In the future, we will attempt to develop interpolating or up-sampling algorithms to rectify the partial spatial resolution deficiency problem.

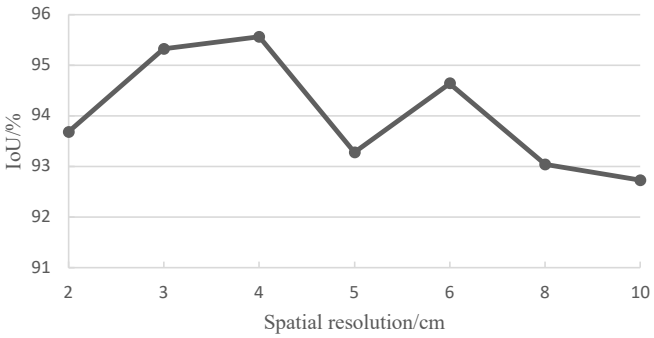


Fig. 17. IoU of semantic segmentation with different spatial resolutions.

So far, our method has only been validated on high-precision laser scanning point clouds, on which the estimation of normal vectors is satisfactory. However, point clouds obtained by photogrammetry, depth cameras, and lidars are not that accurate. To investigate the impact of point accuracy on the segmentation results, we add Gaussian Noise to the coordinates of points to simulate

scanning errors. The theoretical basis is that scanning errors usually follow the Gaussian distribution with zero mean. Thus, we can adjust the value of standard deviation σ to generate point clouds with different levels of scanning errors. The IoU of semantic segmentation with different standard deviations is presented in Fig. 18. We can see that scanning errors have a negative influence on semantic segmentation. However, when the standard deviation is less than 4cm, the benefits of a higher scanning accuracy are not substantial. Therefore, a level of scanning error around 4cm will be adequate for this method, and it is not unreachable for other data acquisition means. But the usability of our method on point clouds acquired through other devices still needs further experimental investigation.

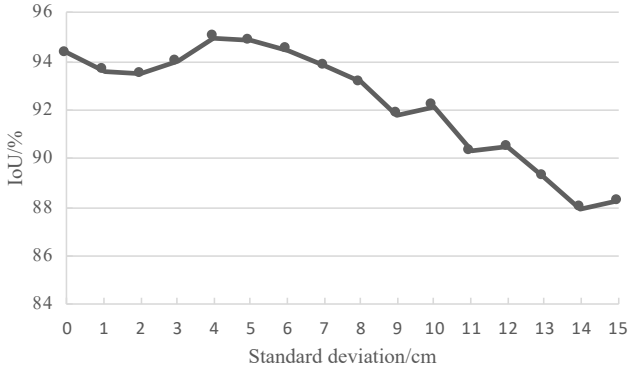


Fig. 18. IoU of semantic segmentation with different scanning accuracies.

In the results of dataset 2 (Table A.2), the segmentation result of pier caps is less satisfactory than that of other elements. This is due to two factors, namely, (1) the similarity of pier caps' geometries in the training and testing data is rather low; (2) the pier caps in the training data are heavily occluded by vegetation. To address the similarity issue, using more training data will be one of the solving options. For the occlusion problem, we have conducted quantitative analysis on the relationship between the level of completeness and the segmentation results (Fig. 19). The occlusions are simulated by randomly dropping point groups with $total\ number\ of\ points/500$ points from original training data. It can be concluded that the segmentation of slabs, girders, and piers is only slightly affected by occlusions. However, lack of completeness will heavily impact the identification of pier caps. It is because pier caps are usually located in the intersection areas of all structural components, which have complex geometries and topological relationships. Consequently, occlusions will significantly disturb the performance of descriptors in pier caps. To conclude, during laser scanning, the level of completeness should be considered to outweighing spatial resolution and

scanning accuracy.

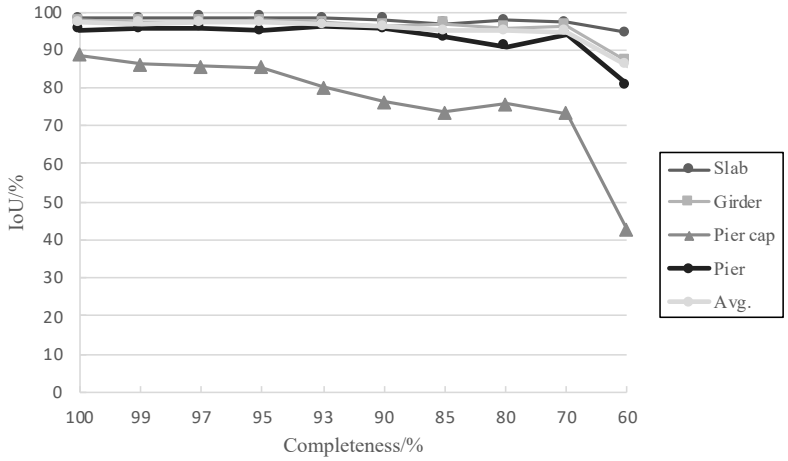


Fig. 19. IoU of semantic segmentation with different levels of completeness.

From the above analysis, it can be seen that our method is efficient in detecting the key components in RC slab and beam-slab bridges. Also, the method can be used in other structures such as buildings and tunnels whose structural components can be equally represented by the similar descriptors. Although there is more room to improve in the future, the present method has provided a very promising means to automate semantic segmentation with limited training data. Such future work includes enriching the descriptors for structures of more complex geometric shapes and structural components.

6. Conclusions

This paper proposes a combined local descriptor and machine learning based method to detect primary structural components in bridges. After comparing various descriptors' performance in different scales, we choose a customized multi-scale local descriptor for each point in the point cloud. ANN is then applied to classify points according to the calculated descriptors, and results are further refined by DBSCAN. The validation results prove that our method has a high degree of segmentation accuracy on RC slab and beam-slab bridges. Since it requires no manual data preprocessing, it offers an approach to facilitate the automatic semantic segmentation of RC bridges.

The novel contributions of this research are as follows:

1. The present method can automatically distinguish environmental point clouds such as vegetation, ground road surfaces, and scanning noise. Thus, the semantic segmentation requires no manual data

preprocessing and operates in a fully automated manner.

2. This method designs a multi-scale local descriptor customized for standard geometric features in typical structural settings. Therefore, the performance of our descriptor in recognition of structural components substantially overweighs other traditional descriptors created for generic objects in computer science.
3. The proposed method depends on identifying the geometric features of various objects by machine-learning the distribution patterns of normal vectors on a high abstraction level. Thus, in contrast to algorithm-based approaches, it requires no specific assumption on geometries and accomplishes higher versatility.
4. The proposed combined local descriptor and machine learning based method for semantic segmentation only requires small learning sample. Experiments in this study have demonstrated that this method can improve the mean IoU from 45.92% to 94.72% as compared to PointNet. This has well addressed the problem of gaining a large amount of training data encountered in practice.
5. From the computational cost perspective, this method can reduce computational cost since descriptors are only attached to local regions. So large-scale bridges with high-density point clouds can also be easily dealt with.

However, this method still has some obvious limitations as follows:

1. The performance of descriptors heavily depends on the calculation of normal vectors at each point. It has been suggested that a minimum spatial resolution of 4cm and a maximum scanning error of 4cm should be met in order to achieve satisfactory semantic segmentation. No experimental evidence has confirmed that the present method may be directly applicable to the less accurate reality capture solutions (e.g., photogrammetry, depth camera, and lidar), which requires future exploration.
2. Although the method is not sensitive to the variation of spatial resolution in point clouds, very dramatic drop in point density may cause problems in calculating descriptors correctly. Density smoothening techniques (e.g., up-sampling or local densification) can be considered to solve this issue.
3. The level of completeness of the point cloud has a large impact on the recognition of some components such as pier caps. Due measures (e.g., changing scanning angles or distances) should be taken when scanning the locations where occlusion may occur.
4. It is noted that the local descriptor is designed and proven to work well in RC slab and beam-slab

bridges. For other types of bridges with complex geometry (e.g., truss bridges) or other infrastructure, the proposed methodology may as well have the potential but needs further experimental verification.

In conclusion, the proposed method has realized automatic semantic segmentation of typical bridges, which has fulfilled one key step in carrying out automatic BrIM. Future works will concentrate on (1) collecting point clouds with different devices to investigate the applicability of our method; (2) validating the proposed method on other types of infrastructure; and (3) developing IFC entity fitting algorithms to realize automatic BrIM.

Acknowledgements

This work was supported by the Scientific Research Project of Shanghai Science and Technology Commission (No.18DZ1205603, 20DZ1201300, 21DZ1204704).

.

Appendix A

Table A.1

Quantitative analysis of semantic segmentation results from the neural network of dataset 2.

	Precision					Recall					IoU				
	Slab	Girder	Pier cap	Pier	Avg.	Slab	Girder	Pier cap	Pier	Avg.	Slab	Girder	Pier cap	Pier	Avg.
Bridge2-3	97.77%	None	91.44%	95.90%	96.84%	99.66%	None	84.40%	99.35%	97.67%	97.45%	None	78.22%	95.30%	94.65%
Bridge2-4	98.68%	98.70%	95.33%	95.68%	98.22%	99.60%	99.05%	92.77%	99.36%	98.99%	98.29%	97.78%	88.73%	95.09%	97.25%
Avg.	98.23%	98.70%	93.39%	95.79%	97.53%	99.63%	99.05%	88.59%	99.36%	98.33%	97.87%	97.78%	83.48%	95.20%	95.95%

Table A.2

Quantitative analysis of semantic segmentation results after refinement of dataset 2.

	Precision					Recall					IoU				
	Slab	Girder	Pier cap	Pier	Avg.	Slab	Girder	Pier cap	Pier	Avg.	Slab	Girder	Pier cap	Pier	Avg.
Bridge2-3	98.30%	None	97.97%	96.58%	98.09%	99.66%	None	84.40%	99.35%	97.67%	97.97%	None	82.95%	95.97%	95.85%
Bridge2-4	98.78%	98.99%	97.37%	95.98%	98.51%	99.60%	99.05%	92.77%	99.36%	98.99%	98.39%	98.06%	90.50%	95.40%	97.54%
Avg.	98.54%	98.99%	97.67%	96.28%	98.30%	99.63%	99.05%	88.59%	99.36%	98.33%	98.18%	98.06%	86.73%	95.69%	96.70%

References

[1] ASCE, America's Infrastructure Report Card 2021, Available at https://infrastructurereportcard.org/wp-content/uploads/2020/12/National_IRC_2021-report.pdf, (2021), Accessed date: 23 August 2021.

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

[3] H.-F. Wang, L. Zhai, H. Huang, L.-M. Guan, K.-N. Mu, G.-p. Wang, Measurement for cracks at the bottom of bridges based on tethered creeping unmanned aerial vehicle, Automation in Construction 119 (2020) 103330. https://doi.org/10.1016/j.autcon.2020.103330.

[4] AASHTOWare, AASHTOWare Pavement ME Design Build 2.6 Release Notes (PDF), Available at https://me-design.com/MEDesign/data/AASHTOWare%20Pavement%20ME%20Design%20Build%202.6%20Release%20Notes.pdf, (2020), Accessed date: 23 August 2021.

[5] R.D. Lu, I. Brilakis, Digital twinning of existing reinforced concrete bridges from labelled point clusters, Automation in Construction 105 (2019) 16. https://doi.org/10.1016/j.autcon.2019.102837.

[6] NIBS - National BIM Standard Organization, National BIM standard–United States, V3, Available at https://www.nationalbimstandard.org/files/NBIMS-US_V3_4.2_COBie.pdf, (2017), Accessed date: 23 August 2021.

[7] A. Costin, A. Adibfar, H. Hu, S.S. Chen, Building Information Modeling (BIM) for transportation infrastructure – Literature review, applications, challenges, and recommendations, Automation in Construction 94 (2018) 257-281. https://doi.org/10.1016/j.autcon.2018.07.001.

[8] F. Tanaka, M. Hori, M. Onosato, H. Date, S. Kanai, Bridge information model based on IFC standards and web content providing system for supporting an inspection process, in: Proceedings of 16th International Conference on Computing in Civil and Building Engineering, Osaka, Japan, 2016, pp. 1140-1147. Available at http://www.see.eng.osaka-u.ac.jp/seeit/icccbe2016/Proceedings/Full_Papers/144-301.pdf, Accessed date: 23 August 2021.

[9] S. DiBernardo, Integrated modeling systems for bridge asset management—Case study, in: Structures Congress, ASCE, Chicago, United States, 2012, pp. 483-493. https://doi.org/10.1061/9780784412367.043.

[10] Dodge Data & Analytics, The business value of BIM for infrastructure 2017, Available at https://www2.deloitte.com/content/dam/Deloitte/us/Documents/finance/us-fas-bim-infrastructure.pdf, (2017), Accessed date: 23 August 2021.

[11] T. Czerniawski, F. Leite, Automated digital modeling of existing buildings: A review of visual object recognition methods, Automation in Construction 113 (2020) 19. https://doi.org/10.1016/j.autcon.2020.103131.

[12] H.S. Park, H. Lee, H. Adeli, I. Lee, A new approach for health monitoring of structures: terrestrial laser scanning, Computer-Aided Civil and Infrastructure Engineering 22 (1) (2007) 19-30. https://doi.org/10.1111/j.1467-8667.2006.00466.x.

[13] B.J. Perry, Y.L. Guo, R. Atadero, J.W. van de Lindt, Streamlined bridge inspection system utilizing unmanned aerial vehicles (UAVs) and machine learning, Measurement 164 (2020) 14. https://doi.org/10.1016/j.measurement.2020.108048.

[14] D. Roca, J. Armesto, S. Lagüela, L. Díaz-Vilariño, LIDAR-equipped UAV for building information modelling, in: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 40, International Society for Photogrammetry and Remote Sensing, Riva del Garda, Italy, 2014, pp. 523-527. http://dx.doi.org/10.5194/isprsarchives-XL-5-523-2014.

[15] R.D. Lu, I. Brilakis, C.R. Middleton, Detection of Structural Components in Point Clouds of Existing RC Bridges, Computer-Aided Civil and Infrastructure Engineering 34 (3) (2019) 191-212. https://doi.org/10.1111/mice.12407.

[16] R. Qiu, Q.-Y. Zhou, U. Neumann, Pipe-run extraction and reconstruction from point clouds, in: European Conference on Computer Vision, Springer, 2014, pp. 17-30. https://doi.org/10.1007/978-3-319-10578-9_2.

[17] A.R. Dick, P.H. Torr, R. Cipolla, Modelling and interpretation of architecture from several images, International Journal of Computer Vision 60 (2) (2004) 111-134. https://doi.org/10.1023/b:visi.0000029665.07652.61.

[18] F. Lafarge, R. Keriven, M. Brédif, Insertion of 3-D-primitives in mesh-based representations: towards compact models preserving the details, IEEE Transactions on Image Processing 19 (7) (2010) 1683-1694. https://doi.org/10.1109/tip.2010.2045695.

[19] F. Lafarge, C. Mallet, Building large urban environments from unstructured point data, in: 2011 International Conference on Computer Vision, IEEE, Barcelona, Spain 2011, pp. 1068-1075. https://doi.org/10.1109/iccv.2011.6126353.

[20] Q.C. Lu, S. Lee, Image-Based Technologies for Constructing As-Is Building Information Models for Existing Buildings, Journal of Computing in Civil Engineering 31 (4) (2017) 14. https://doi.org/10.1061/(asce)cp.1943-5487.0000652.

[21] B. Wang, C. Yin, H. Luo, J.C.P. Cheng, Q. Wang, Fully automated generation of parametric BIM for MEP scenes based on terrestrial laser scanning data, Automation in Construction 125 (2021) 103615. https://doi.org/10.1016/j.autcon.2021.103615.

[22] F. Bosché, Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction, Advanced Engineering Informatics 24 (1) (2010) 107-118. https://doi.org/10.1016/j.aei.2009.08.006.

[23] X. Xiong, A. Adan, B. Akinci, D. Huber, Automatic creation of semantically rich 3D building models from laser scanner data, Automation in Construction 31 (2013) 325-337. https://doi.org/10.1016/j.autcon.2012.10.006.

[24] H. Kim, J. Yoon, S.H. Sim, Automated bridge component recognition from point clouds using deep learning, Structural Control & Health Monitoring 27 (9) (2020) 13. https://doi.org/10.1002/stc.2591.

[25] Y. Yan, J.F. Hajjar, Automated extraction of structural elements in steel girder bridges from laser point clouds, Automation in Construction 125 (2021) 103582. https://doi.org/10.1016/j.autcon.2021.103582.

[26] Q. Wang, M.K. Kim, Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018, Advanced Engineering Informatics 39 (2019) 306-319. https://doi.org/10.1016/j.aei.2019.02.007.

[27] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, The 3d hough transform for plane detection in point clouds: A review and a new accumulator design, 3D Research 2 (2) (2011) 3. https://doi.org/10.1007/3dres.02(2011)3.

[28] D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition 13 (2) (1981) 111-122. https://doi.org/10.1016/0031-3203(81)90009-1.

[29] G. Vosselman, B.G. Gorte, G. Sithole, T. Rabbani, Recognising structure in laser scanner point clouds, International archives of photogrammetry, remote sensing and spatial information sciences 46 (8) (2004) 33-38. Available at https://www.isprs.org/proceedings/xxxvi/8-w2/VOSSELMAN.pdf , Accessed date: 23 August 2021.

[30] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381-395. https://doi.org/10.1145/358669.358692.

[31] R. Schnabel, R. Wahl, R. Klein, Efficient RANSAC for point-cloud shape detection, in: Computer graphics forum, Vol. 26, Wiley, 2007, pp. 214-226. https://doi.org/10.1111/j.1467-8659.2007.01016.x.

[32] T.M. Awwad, Q. Zhu, Z. Du, Y. Zhang, An improved segmentation approach for planar surfaces from unstructured 3D point clouds, The Photogrammetric Record 25 (129) (2010) 5-23. https://doi.org/10.1111/j.1477-9730.2009.00564.x.

[33] T. Rabbani, F. Van Den Heuvel, G. Vosselmann, Segmentation of point clouds using smoothness constraint, in: 2006 ISPRS Commission V Symposium on Image Engineering and Vision Metrology, Vol. 36, International Society for Photogrammetry and Remote Sensing, Dresden, Germany, 2006, pp. 248-253. Available at https://www.isprs.org/proceedings/XXXVI/part5/paper/RABB_639.pdf, Accessed date: 23 August 2021.

[34] P.J. Besl, R.C. Jain, Segmentation through variable-order surface fitting, Ieee Transactions on Pattern Analysis and Machine Intelligence 10 (2) (1988) 167-192. https://doi.org/10.1109/34.3881.

[35] J.-E. Deschaud, F. Goulette, A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing, in: International Symposium on 3D Data Processing, Visualization and Transmission, Paris, France, 2010, pp. 44-54. Available at https://hal-mines-paristech.archives-ouvertes.fr/hal-01097361/document, Accessed date: 23 August 2021.

[36] T. Rabbani, F. Van Den Heuvel, Efficient Hough transform for automatic detection of cylinders in point clouds, in: 2005 ISPRS Workshop Laser Scanning 2005, Vol. 36, International Society for Photogrammetry and Remote Sensing, Enschede, Netherlands, 2005, pp. 60-65. Available at https://www.isprs.org/proceedings/xxxvi/3-w19/papers/060.pdf, Accessed date: 23 August 2021.

[37] F. Tarsha-Kurdi, T. Landes, P. Grussenmeyer, Extended RANSAC algorithm for automatic detection of building roof planes from LiDAR data, The Photogrammetric Journal of Finland 21 (2008) 97–109. Available at https://halshs.archives-ouvertes.fr/halshs-00264843/document , Accessed date: 23 August 2021.

[38] G. Zhang, P.A. Vela, P. Karasev, I. Brilakis, A sparsity-inducing optimization-based algorithm for planar patches extraction from noisy point-cloud data, Computer-Aided Civil and Infrastructure Engineering 30 (2) (2015) 85-102. https://doi.org/10.1111/mice.12063.

[39] A.K. Patil, P. Holi, S.K. Lee, Y.H. Chai, An adaptive approach for the reconstruction and modeling of as-built 3D pipelines from point clouds, Automation in Construction 75 (2017) 65-78. https://doi.org/10.1016/j.autcon.2016.12.002.

[40] A.-V. Vo, L. Truong-Hong, D.F. Laefer, M. Bertolotto, Octree-based region growing for point cloud segmentation, ISPRS Journal of Photogrammetry and Remote Sensing 104 (2015) 88-100. https://doi.org/10.1016/j.isprsjprs.2015.01.011.

[41] A. Nurunnabi, D. Belton, G. West, Robust segmentation in laser scanning 3D point cloud data, in: 2012 International Conference on Digital Image Computing Techniques and Applications (DICTA), IEEE, Fremantle, Australia, 2012, pp. 1-8. https://doi.org/10.1109/dicta.2012.6411672.

[42] S.B. Walsh, D.J. Borello, B. Guldur, J.F. Hajjar, Data processing of point clouds for object detection for structural engineering applications, Computer-Aided Civil and Infrastructure Engineering 28 (7) (2013) 495-508. https://doi.org/10.1111/mice.12016.

[43] F. Dell'Acqua, R. Fisher, Reconstruction of planar surfaces behind occlusions in range images, Ieee Transactions on Pattern Analysis and Machine Intelligence 24 (4) (2002) 569-575. https://doi.org/10.1109/34.993564.

[44] M. Pauly, N.J. Mitra, J. Giesen, M.H. Gross, L.J. Guibas, Example-based 3D scan completion, in: Symposium on Geometry Processing, Vienna, Austria, 2005, pp. 23-32. Available at http://graphics.stanford.edu/~niloy/research/shape_completion/paper_docs/shape_completion_sgp_05.pdf, Accessed date: 23 August 2021.

[45] C. Frueh, S. Jain, A. Zakhor, Data processing algorithms for generating textured 3D building facade meshes from laser scans and camera images, International Journal of Computer Vision 61 (2) (2005) 159-184. https://doi.org/10.1023/b:visi.0000043756.03810.dd.

[46] A. Adan, D. Huber, 3D reconstruction of interior wall surfaces under occlusion and clutter, in: 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, IEEE, Hangzhou, China, 2011, pp. 275-281. https://doi.org/10.1109/3DIMPVT.2011.42.

[47] S. Pu, G. Vosselman, Knowledge based reconstruction of building models from terrestrial laser scanning data, ISPRS Journal of Photogrammetry and Remote Sensing 64 (6) (2009) 575-584. https://doi.org/10.1016/j.isprsjprs.2009.04.001.

[48] V. Sanchez, A. Zakhor, Planar 3D modeling of building interiors from point cloud data, in: IEEE International Conference on Image Processing, IEEE, Orlando, USA, 2012, pp. 1777-1780. https://doi.org/10.1109/icip.2012.6467225.

[49] X. Liu, M. Eybpoosh, B. Akinci, Developing as-built building information model using construction process history captured by a laser scanner and a camera, in: Construction Research Congress 2012: Construction Challenges in a Flat World, ASCE, West Lafayette, United States, 2012, pp. 1232-1241. https://doi.org/10.1061/9780784412329.124.

[50] F. Bosche, C.T. Haas, Automated retrieval of 3D CAD model objects in construction range images, Automation in Construction 17 (4) (2008) 499-512. https://doi.org/10.1016/j.autcon.2007.09.001.

[51] Y. Turkan, F. Bosche, C.T. Haas, R. Haas, Automated progress tracking using 4D schedule and 3D sensing technologies, Automation in Construction 22 (2012) 414-421. https://doi.org/10.1016/j.autcon.2011.10.003.

[52] B. Riveiro, M. DeJong, B. Conde, Automated processing of large point clouds for structural health monitoring of masonry arch bridges, Automation in Construction 72 (2016) 258-268. https://doi.org/10.1016/j.autcon.2016.02.009.

[53] Y.-P. Zhao, H. Wu, P.A. Vela, Top-Down Partitioning of Reinforced Concrete Bridge

Components, in: ASCE International Conference on Computing in Civil Engineering ASCE, Atlanta, Georgia, 2019, pp. 275-283. https://doi.org/10.1061/9780784482445.035.

[54] D. Maturana, S. Scherer, Voxnet: A 3d convolutional neural network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Hamburg, Germany, 2015, pp. 922-928. https://doi.org/10.1109/iros.2015.7353481.

[55] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3d shape recognition, in: Proceedings of the IEEE international conference on computer vision, IEEE, Santiago, Chile, 2015, pp. 945-953. https://doi.org/10.1109/iccv.2015.114.

[56] C.R. Qi, H. Su, K. Mo, L.J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, Honolulu, Hawaii, 2017, pp. 652-660. https://doi.org/10.1109/cvpr.2017.16.

[57] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: Advances in neural information processing systems, MIT Press, Long Beach, United States, 2017, pp. 5099-5108. https://arxiv.org/abs/1706.02413.

[58] S. Shi, X. Wang, H. Li, Pointrcnn: 3d object proposal generation and detection from point cloud, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Long Beach, USA, 2019, pp. 770-779. https://doi.org/10.1109/cvpr.2019.00086.

[59] Y.Y. Li, R. Bu, M.C. Sun, W. Wu, X.H. Di, B.Q. Chen, PointCNN: Convolution On X - Transformed Points, in: Advances in neural information processing systems, Vol. 31, MIT Press, Montréal, Canada, 2018, pp. 828–838. https://arxiv.org/abs/1801.07791.

[60] Y. Wang, Y.B. Sun, Z.W. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic Graph CNN for Learning on Point Clouds, Acm Transactions on Graphics 38 (5) (2019) 12. https://doi.org/10.1145/3326362.

[61] H. Kim, C. Kim, Deep-Learning-Based Classification of Point Clouds for Bridge Inspection, Remote Sensing 12 (22) (2020) 3757. https://doi.org/10.3390/rs12223757.

[62] D.H. Wolpert, Stacked generalization, Neural networks 5 (2) (1992) 241-259. https://doi.org/10.1016/S0893-6080(05)80023-1.

[63] H.S. Koppula, A. Anand, T. Joachims, A. Saxena, Semantic labeling of 3D point clouds for indoor scenes, in: Advances in neural information processing systems, Curran Associates, Granada, Spain, 2011, pp. 244-252. Available at https://proceedings.neurips.cc/paper/2011/hash/9872ed9fc22fc182d371c3e9ed316094-Abstract.html, Accessed date: 23 August 2021.

[64] G. Zhang, P.A. Vela, I. Brilakis, Detecting, Fitting, and Classifying Surface Primitives for Infrastructure Point Cloud Data, in: ASCE International Workshop on Computing in Civil Engineering, ASCE, Los Angeles, United States, 2013, pp. 589-596. https://doi.org/10.1061/9780784413029.074.

[65] P. Tsangaratos, I. Ilia, Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size, CATENA 145 (2016) 164-179. https://doi.org/10.1016/j.catena.2016.06.004.

[66] D.A. Cieslak, N.V. Chawla, Learning decision trees for unbalanced data, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Antwerp, Belgium, 2008, pp. 241-256. https://doi.org/10.1007/978-3-540-87479-9_34.

[67] B. Yegnanarayana, Artificial neural networks, 1st ed., Prentice-Hall of India Pvt.Ltd, New Delhi,

2005. ISBN: 8120312538.

[68] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: The Second International Conference on Knowledge Discovery and Data Mining, Vol. 96, AAAI, Portland, Oregon, 1996, pp. 226-231. Available at https://aaai.org/Papers/KDD/1996/KDD96-037.pdf, Accessed date: 23 August 2021.

[69] F. Tombari, S. Salti, L. Di Stefano, Unique signatures of histograms for local surface description, in: European conference on computer vision, Springer, Crete, Greece, 2010, pp. 356-369. https://doi.org/10.1007/978-3-642-15558-1_26.

[70] A.E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scenes, Ieee Transactions on Pattern Analysis and Machine Intelligence 21 (5) (1999) 433-449. https://doi.org/10.1109/34.765655.

[71] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012) 1097-1105. https://doi.org/10.1145/3065386.

[72] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural networks 2 (5) (1989) 359-366. https://doi.org/10.1016/0893-6080(89)90020-8.

[73] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (2014) 1929-1958. https://doi.org/10.5555/2627435.2670313.

[74] S. Ruder, An overview of gradient descent optimization algorithms, ArXiv arXiv:1609.04747 (2016). https://arxiv.org/abs/1609.04747.

[75] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533-536. https://doi.org/10.1038/323533a0.

[76] G. Zhang, P. Vela, I. Brilakis, Automatic generation of as-built geometric civil infrastructure models from point cloud data, in: Proceedings of the 2014 International Conference on Computing in Civil and Building Engineering, ASCE, Orlando, Florida, 2014, pp. 406-413. https://doi.org/10.1061/9780784413616.051.