Automation in Construction 165 (2024) 105572

Contents lists available at ScienceDirect

Automation in Construction

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

Domain knowledge-enhanced region growing framework for semantic segmentation of bridge point clouds

Tao Yang [a], Yang Zou [a,*], Xiaofei Yang [b], Enrique del Rey Castillo [a]

[a] *Department of Civil and Environmental Engineering, University of Auckland, Auckland 1010, New Zealand*
[b] *School of Civil and Environmental Engineering, Nanyang Technological University, 50 Nanyang Ave, S639798, Singapore*

ARTICLE INFO

Keywords:
Point cloud
Semantic segmentation
Bridge component detection
Region growing
Domain knowledge
Finite element analysis

ABSTRACT

Utilising domain knowledge (DK) to semantically segment bridge point clouds has attracted growing research interest. However, current approaches are often tailored to specific bridges, limiting their general applicability. To address this problem, this paper introduces a DK-enhanced Region Growing (DKRG) framework for point cloud semantic segmentation of reinforced concrete (RC) girder bridges. Inspired by the vertical layout characteristics of bridges, the generation of DK-based point features from Finite Element Analysis (FEA) is first proposed. Then, DKRG is employed to segment bridge components from substructures to superstructures by leveraging an “easy-to-difficult” strategy. Validation results demonstrate the effectiveness of our method, achieving the lowest mean Intersection over Union (mIoU) of 95.47% for the entire bridge and 93.44% for different component types. This study provides a new DK-based framework for semantic segmentation of RC girder bridges and sheds new light on using FEA-generated point features.

1. Introduction

Bridges are subject to ageing, dangerous deterioration and corrosion over time due to prolonged usage, environmental impact, and natural disasters. As reported, approximately 42% of the bridges in the United States have been in service for over 50 years [1]. The same trend is also commonly observed in other countries, such as the United Kingdom and Japan [2–4]. Regular inspection and condition assessment of existing bridges are necessary to ensure their safety and serviceability. However, the traditional visual inspection processes are heavily reliant on manpower, and the inspection data is often recorded in a fragmented manner and cannot be efficiently integrated with other bridge-specific data. This traditional inspection significantly hinders the ability to conduct a comprehensive safety assessment and make well-informed decisions regarding bridges. Building Information Modelling (BIM) addresses these problems by creating and managing three-dimensional (3D) geometric and semantic-rich information models throughout the lifecycle of a bridge [5]. BIM serves as a central data repository for the management and storage of information about the as-is condition of bridge structures in the 3D environment [6].

Various emerging reality capture tools such as laser scanning, photogrammetry, and Light Detection and Ranging (LiDAR) have been employed to capture the as-is bridge information for BIM-based inspection data management. These tools can generate point cloud data to describe the as-is condition and support data management of existing bridges, and point cloud can be further used for generating the as-is BIM, also known as the Scan-to-BIM process [7]. The Scan-to-BIM process generally involves two essential steps: (1) semantic segmentation, which involves the assignment of semantic labels to the whole bridge point cloud based on the types of bridge component, and (2) the reconstruction of 3D as-is BIM by fitting the Industry Foundation Classes (IFC) entities with well-labelled point clouds [8]. Accurate segmentation of bridge point clouds can provide a solid foundation for the BIM model generation. However, how to automatically and accurately segment bridge point clouds remains a significant global challenge.

Currently, there are two primary categories of methods for bridge point cloud segmentation, i.e., learning-based methods and algorithm-based methods. Learning-based methods have become increasingly popular in recent years due to advancements in computing power and neural networks such as PointNet [9]. It can provide versatile solutions to various bridge types, such as girder bridges [10,11] and masonry arch bridges [12], and perform the point cloud semantic segmentation in an end-to-end manner because it can automatically extract useful features from the training point clouds. While successful in various tasks, the

* Corresponding author.
E-mail address: yang.zou@auckland.ac.nz (Y. Zou).

https://doi.org/10.1016/j.autcon.2024.105572
Received 3 November 2023; Received in revised form 14 June 2024; Accepted 14 June 2024
Available online 20 June 2024

scarcity of training data in the civil infrastructure sector significantly hinders the wide application of learning-based methods. The model that has been trained on limited bridge point clouds cannot generalise well to another group of bridges with distinct geometric features [13].

To address the limitations of learning-based methods, algorithm-based methods have been developed for bridge point cloud segmentation, primarily the domain knowledge (DK)-based method and Region Growing (RG)-based method. Versatile DK, primarily the domain-specific information such as component spatial distribution, engineering criteria and known design parameters, can be interpreted and developed to assist bridge point cloud segmentation. For instance, the point density histogram, a new manifestation of bridge component spatial distribution, was widely adopted to segment bridge point clouds, particularly in some early efforts for complex bridges such as masonry arch bridges [14] and truss bridges [15]. The engineering knowledge (e.g., the relative position and size of bridge components) and design parameters (e.g., the design value of general transverse maximum gradient and the general longitudinal maximum gradient) were also adopted for the segmentation of reinforced concrete (RC) girder bridges [8] and steel girder bridges [16]. For RG-based segmentation methods, an early effort of using RG algorithm for bridge point cloud segmentation was developed by Walsh et al. [17]. The process started with sharp feature detection to identify the surface boundaries, and the smoothness-constraint-based RG was subsequently utilised to obtain the surface primitives based on predefined sharp features. The RG algorithm can also be combined with DK, such as spatial relationships of bridge components and contextual knowledge, to extract the individual surface of bridge components for damage identification [18].

Despite the promising results presented in the literature, two challenges remain in the utilisation of algorithm-based methods: 1) these frameworks proposed in domain knowledge-based research are often tailored for specific bridges and sensitive to the design and geometric changes of bridges, and no prior work explored to adopt other high-level implicit DK, such as structural mechanical properties, to perform semantic segmentation tasks for bridges; 2) RG algorithms in previous research can only output low-level surface or shape primitives of bridges, and no research has been found that can generate component-level growing results.

To further explore the untapped potential of DK and the use of RG algorithm, this study proposes a novel DK-enhanced RG (DKRG) framework for the semantic segmentation of RC girder bridges. The contributions of this research are primarily the following:

- Introduction of a robust “easy-to-difficult” segmentation strategy, applicable to a wide range of RC girder bridges and enabling segmentation of bridges with complex superstructure components, such as multi-box girder bridges.
- Proposal of a new method for utilising DK through Finite Element Analysis (FEA), allowing the translation of DK into accessible point features for point cloud segmentation.
- Development of a multi-feature-based RG algorithm that incorporates geometric features and DK-based point features for component-level segmentation of bridge point clouds.

The rest of this article is organised as follows. Section 2 presents a review of related works and summarises the research gaps. The research scope and objectives are described in section 3. Section 4 presents the detailed segmentation framework. The evaluation of our proposed framework is illustrated in section 5. Section 6 further distinguishes our proposed framework from previous research. Finally, section 7 concludes the main findings of this study and discusses recommendations for future work.

2. Literature review

2.1. *Learning-based methods*

Learning-based methods, particularly deep learning, are the most influential and fastest-growing technique in point cloud semantic segmentation. A pioneering work presented by Kim et al. [10] leveraged the PointNet-based framework [9] to extract bridge components from point clouds. This method starts by partitioning the bridge point clouds into numerous overlapping subsets, and 2048 points are sampled from each subgroup to serve as the input of PointNet, where an accuracy of 95% was found in the final model prediction. Another study [19] further conducted a comparative analysis of three widely used deep learning models, including PointNet, PointCNN [20], and Dynamic Graph Convolutional Neural Network (DGCNN) [21]. The results revealed that DGCNN outperformed the other two models, exhibiting superior accuracy. Inspired by PointNet and DGCNN, a hierarchical DGCNN model was introduced by Lee et al. [22] for the semantic segmentation of railway bridges with electric poles. The results revealed a better segmentation performance for electric poles and deck, and relatively lower accuracy for detecting abutments and piers.

The methods above are mainly designed for small-scale bridge point clouds with a limited number of input points, which is inappropriate to process the large-scale bridge point clouds directly. To address this issue, some researchers presented a BridgeNet neural network [12] to segment large-scale masonry arch bridge point clouds. The result revealed that the BridgeNet with a 1 × 1 convolution layer can accurately segment point clouds and outperform other neural networks like PointNet++ and RandLA-Net [23]. In addition, the super-point method [24] was developed to reduce data input by grouping point clouds into hundreds of super-points with semantic homogeneity in the initial stage. For instance, Lin and Habib [25] leveraged the super-point graph (SPG) framework to automatically segment the point cloud of road infrastructure, and the abutment, deck, and pier were successfully detected from 17 highway bridge point clouds. Further, a weighted SPG was proposed by Yang et al. [26] to alleviate the data imbalance issue by using a novel loss function and validate its effectiveness on almost all key RC bridge component types. The result indicated that the proposed method outperformed PointNet, DGCNN, and the original SPG in the test of both synthetic and real-world datasets. Nonetheless, the segmentation accuracy of the parapet, abutment, and diaphragms is relatively lower compared with other bridge components.

To summarise, the learning-based method provides a promising solution to perform the semantic segmentation of bridge point clouds in an end-to-end manner. However, training data scarcity is the main barrier to this method. Although the synthetic dataset [12,26] and synthetic data augmentation strategies [11,27] can be used as a remedy for the data scarcity challenge, these datasets are idealised point clouds primarily converted from as-design BIM models and fail to consider the data incompleteness, different noises, and other uncertainties in real-world data collection process. Additionally, the bridges in the real world are likely to suffer from various damage, such as component deficiency and deformation, which is less considered in the synthetic data in previous research. Therefore, synthetic data still cannot 100% address the data scarcity issue and cannot guarantee the reliability of segmentation results.

2.2. *Algorithm-based methods*

2.2.1. *DK-based methods*

The DK-based methods often involve using domain-specific knowledge such as design rules, topological constraints, and the element spatial relationships for bridge point cloud segmentation. The advantages of using DK in bridge scenes are twofold: (1) Compared with building scenes, bridges display more noticeable spatial distribution characteristics of components, and the components in both longitudinal and vertical directions are highly structured and well-organised; (2) The stable bridge component types, e.g., the pier, girder, deck, and parapet, also provide a friendly environment to encode and use DK in point cloud segmentation. Thus, DK-based methods are widely used in the segmentation of different types of bridges, such as masonry arch bridges [14], RC slab-beam bridges [8,17,18,28–30], truss bridges [15], steel girder bridges [16].

The density histogram, another manifestation of bridge component spatial distribution, has been broadly adopted in previous research [8,14–16,30] because the various bridge components can be distinguished according to the projected point cloud histogram in different views. For instance, the cutting plane obtained from the projected point cloud histogram was adopted by Zhao et al. [30] to extract key components of RC bridges. An early effort on the segmentation of masonry arch bridges by using the point cloud histogram was presented by Riveiro et al. [14]. Recently, a pioneering work of using density histograms for the segmentation of truss bridges was reported by Lamas et al. [15]. The density histograms in the views of the Z and Y coordinate directions were first used to segment point clouds into several faces. The structural elements were then detected from point density histograms according to the element distribution characteristics. Researchers have also investigated the use of point density histograms to segment specific bridge component types such as girder [8], deck surface and cross-frame [16]. However, the utilisation of point density may make this method more sensitive to the factors that could impact projection quality, such as data quality problems and the horizontal curvature and vertical tilt of bridges. This may result in the inaccurate identification of the coordinate range for point cloud segmentation.

In addition to density histograms, DK can be interpreted as rules or constraints to guide the bridge point cloud segmentation. To make these DK feature more distinguishable, the slicing of bridge point clouds has also been commonly used in previous research. For example, the slicing method proposed by Lu et al. [8] aimed to simplify the segmentation of a whole RC bridge into several tasks, each focused on detecting bridge components within specific assemblies. Engineering knowledge-based DK (e.g., piers should start from the ground and higher than the deck, and the general transverse maximum gradient is defined as 5%) was interpreted as rules to guide the segmentation process. Despite the satisfactory result for RC slab and beam-slab bridges, the heavy demand for engineering parameters would result in poor applicability of the proposed segmentation framework, particularly to bridges with more complex geometric and design features. Moreover, this research focused on the substructure. The components in superstructures, such as parapets and decks, were not sufficiently segmented. A similar slicing framework was also proposed by Yan and Hajjar [16] for curved steel girder bridges. In this approach, the traffic direction estimation was performed in advance to guide the slicing process. After slicing, the geometric and topological constraints from DK were utilised to guide the segmentation of bridge components. Specifically, the longitudinal structural elements can be detected by evaluating the relative orientations of the surface normal underlying each point with respect to the estimated traffic direction. The relative orientation of the bridge component has been used to detect the lower deck surface, cross-frames, and substructure.

2.2.2. Region growing-based methods

RG is a classical algorithm in point cloud segmentation [31]. It was developed initially for measuring the 2D pixel of images and was subsequently adopted to process 3D point clouds. The main idea behind the RG is to assess the feature similarity between seed points/patches and neighbour points/patches and group the units with similar features into the same clusters.

A common RG approach comprises several important steps, including seed point selection, growth unit setting, growing criterion definition, threshold setting, and further allocation of un-clustered points [32,33]. Among them, the growing criterion definition refers to the rules for grouping two points into the same segments from the comparison of feature similarity, which is the core step that can significantly affect the final growing results. The most widely used features in point cloud segmentation include the point normals, curvature, and Euclidean distance. The adaptation of different features could result in different segmentation results. For example, Poux et al. [32] and Vo et al. [34] used normal-based RG to cluster point clouds of building scenes into plane segments and achieved satisfactory results. Several studies introduced high-level features in the RG algorithm, such as roughness [35] and smoothness [36] of point clouds, where they can output curved point segments for the Mechanical, Electrical, and Plumbing (MEP) scenes.

Few researchers have adopted the RG algorithm in bridge scenes due to the complex geometric features of bridge components [17,18,37]. A two-stage method of using the RG algorithm for bridge point cloud segmentation was developed by Walsh et al. [17]. The process commenced with sharp feature detection to identify the surface boundaries. Subsequently, the smoothness-constraint-based RG was utilised to obtain the surface primitives based on predefined sharp features. However, in their experiment, this method failed to detect the border between the pier cap and pier in a small portion of a bridge point cloud. In another study reported by Truong-Hong and Lindenbergh [18], the RG can combine with the spatial relationships of bridge components and contextual knowledge to detect the surface of each bridge component from point clouds, which has been proved to successfully extract the surface of each component for box-girder and slab-beam bridges. The key assumption behind this approach is that the orientation of the bottom surface of the girder differs from its adjacent surfaces of the substructure, allowing for distinguishing the superstructure and substructure. This assumption may result in its limited applicability and make this approach fail to distinguish the superstructure and substructure for bridges with more complex girders, such as multiple-box girder bridges. Overall, the segmentation results using RG-based methods are primarily the surface primitive of bridge components, and no existing RG approach can output the component-level segmentation results.

2.3. Research gaps

Based on the literature review, two research gaps are identified in current algorithm-based approaches for bridge point cloud segmentation:

Gap 1. DK-based segmentation methods present competent capability in the point cloud segmentation of various bridge types. However, the existing efforts are based on predefined procedures and are usually tailored for certain groups of bridges, which are not generalisable to other bridges with different designs and geometric features. Thus, there is a need to explore a more robust and generalisable DK-based framework for bridge point cloud segmentation.

Gap 2. The RG method holds the capability of leveraging sophisticated features to achieve the expected growth results, e.g., planes and curved surfaces. However, a significant challenge of the RG method in previous studies is that the output of the RG algorithm is primarily over-segmented planar or curved-surface primitives, and no research has been found that can generate component-level growing results, which significantly limits its application in bridge scenes.

3. Research scope and objectives

3.1. Scope

This research focuses on typical RC bridges, including slab bridges, beam-slab bridges and single or multi-box girder bridges. These bridges have been reported to account for 80% of existing RC bridges and 89% of the planned future bridges in the United Kingdom [38] and constitute the largest proportion of bridge deficiencies [39]. The RC bridges generally consist of superstructures and substructures, where the substructure often includes piers and pier caps, and the superstructure primarily includes the girders, decks and parapets above the bearings. This study considers five key components of RC bridges in our segmentation framework, including piers, pier caps, girders, decks and parapets.

3.2. Research objectives

Bridges often exhibit the noticeable vertical distribution characteristics of bridge components since components are often systematically interconnected in the vertical direction. For instance, piers often serve as connectors between the girder or pier cap and the ground, while the deck and parapet are supported by the girder. According to the absence of bearing and pier caps, these typical RC bridges can be divided into four classical vertical layouts [6], as shown in Fig. 1.

Based on the bridge's vertical layouts, a novel DKRG framework for semantic segmentation of RC bridge point clouds is proposed to narrow these gaps identified in subsection 2.3. The framework combines the use of DK and RG to achieve the following two research objectives (ROs):

RO.1 Improving the generalisability of DK-based methods for bridge point cloud semantic segmentation.

RO.2 Empowering RG-based methods to be capable of component-level segmentation of bridge point clouds.

To improve the generalisability of DK-based methods (RO.1), this paper proposes a new “easy-to-difficult” segmentation strategy and a novel method of using DK for point cloud segmentation.

Firstly, we provide an “easy-to-difficult” segmentation strategy by considering the complexity of bridge components. As shown in Fig. 1, the geometric features of superstructure components, such as girders and parapets, are often more complex and changeable than those in substructure, such as piers and pier caps. The removal of relatively simpler bridge components can reduce the difficulty in further segmenting more complex components. Thus, our method follows the workflow to extract bridge components from substructures to superstructures, which can display better robustness for segmenting more complex superstructure components, such as girders and parapets.

Secondly, we propose a novel method of translating the bridge vertical layout characteristic into the accessible point-level feature through FEA for point cloud segmentation, which is radically different from directly interpreting DK as rules or constraints. According to structural DK, the bridge component with a relatively lower position tends to resist more gravity under dead load. The sharply changed cross-section along the boundary between bridge components can also result in the noticeable variation gradient of structural response. The response feature of bridge components can remain similar in the same components and can be distinguished among different components. Thus, the structural response feature can be regarded as a component-level feature that partially captures the vertical layout characteristics of the bridges.

To enable RG-based methods to be capable of generating component-level segmentation results (RO.2), a DK-enhanced RG algorithm is introduced in this study.

The DK-based point feature is first introduced as a component-level feature to account for the differences among various bridge components. A multi-feature-based RG segmentation framework that incorporates geometric features and DK-based point features is subsequently designed. Finally, with the guide of structural knowledge, the RG algorithm is employed with the DK-based point feature and geometry feature to perform the component-level segmentation from substructure to superstructures.

4. Overall framework

The overall DKRG framework is illustrated in Fig. 2. It consists of four phases: (1) data pre-processing, (2) substructure segmentation, (3) superstructure segmentation, and (4) original point cloud segmentation.

The first phase involves data pre-processing and feature generation. Two down-sampling processes are introduced and performed in Section 4.1.1. The raw point cloud from laser-scanning is firstly sampled by a

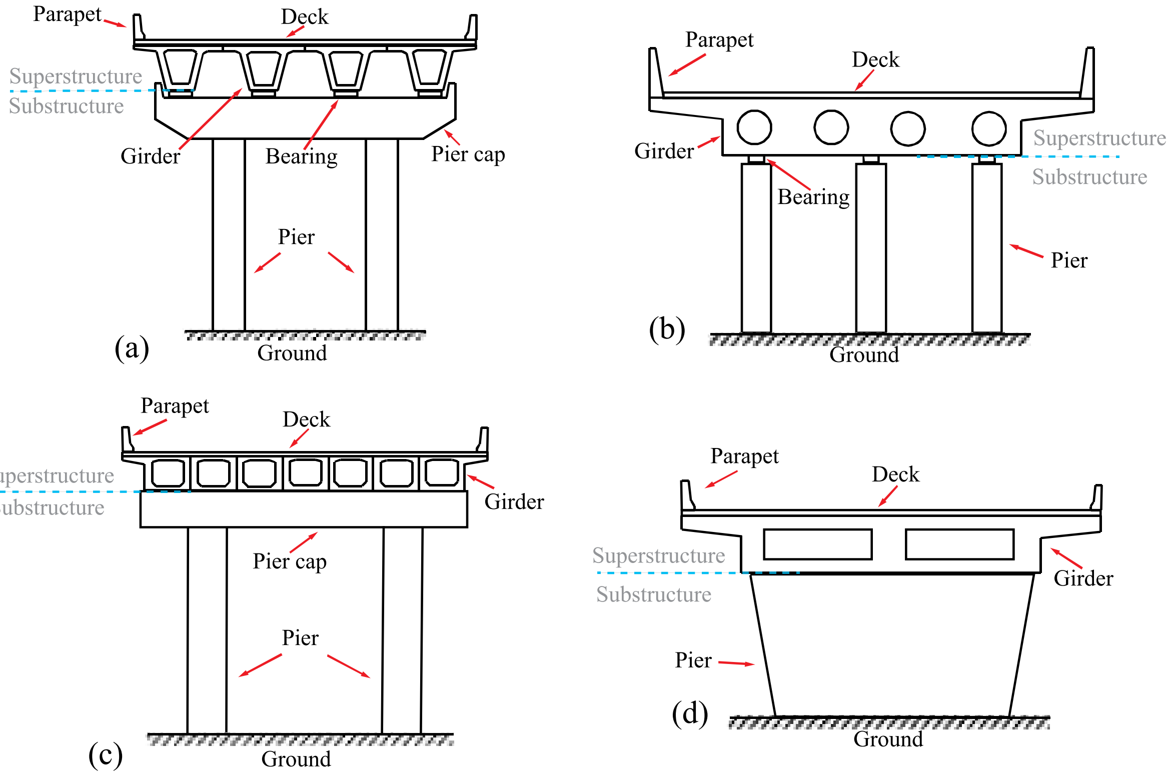

Fig. 1. Typical vertical layouts of bridges: (a) bridges with both bearings and pier caps, (b) bridges with bearings but no pier caps, (c) bridges with pier caps but no bearings, and (d) bridges without both pier caps and bearings.

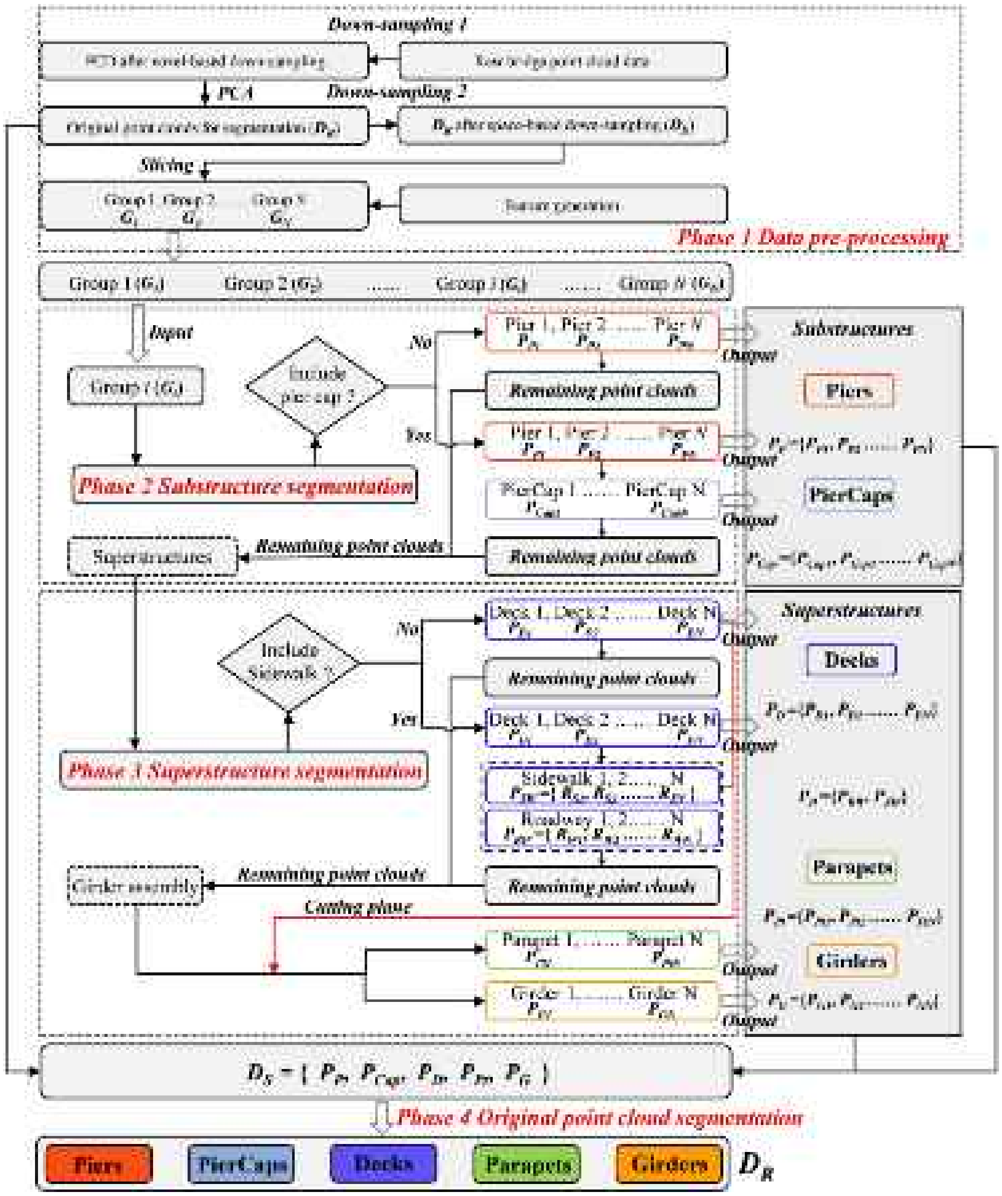
Down-sampling 1
PCD after voxel-based down-sampling
Raw bridge point cloud data
PCA
Down-sampling 2
Original point clouds for segmentation (D_R)
D_R after space-based down-sampling (D_S)
Slicing
Group 1 Group 2 Group N
G_1 G_2 G_N
Feature generation
Phase 1 Data pre-processing
Group 1 (G_1) Group 2 (G_2) ...... Group i (G_i) ...... Group N (G_N)
Input
Group i (G_i)
Include pier cap ?
No
Yes
Pier 1, Pier 2 ...... Pier N
P_{P1} P_{P2} P_{PN}
Remaining point clouds
PierCap 1 ...... PierCap N
P_{Cap1} P_{CapN}
Phase 2 Substructure segmentation
Superstructures
Remaining point clouds
Output
Substructures
Piers
$P_P=\{P_{P1}, P_{P2} ...... P_{PN}\}$
PierCaps
$P_{Cap}=\{P_{Cap1}, P_{Cap2} ...... P_{CapN}\}$
Include Sidewalk ?
No
Yes
Deck 1, Deck 2 ...... Deck N
P_{D1} P_{D2} P_{DN}
Sidewalk 1, 2......N
$P_{SW}=\{R_{S1}, R_{S2} ...... R_{SN}\}$
Roadway 1, 2......N
$P_{RW}=\{R_{R1}, R_{R2} ...... R_{RN}\}$
Phase 3 Superstructure segmentation
Girder assembly
Remaining point clouds
Cutting plane
Parapet 1...... Parapet N
P_{Pt1} P_{PtN}
Girder 1...... Girder N
P_{G1} P_{GN}
Superstructures
Decks
$P_D=\{P_{D1}, P_{D2} ...... P_{DN}\}$
$P_D=\{P_{RW}, P_{SW}\}$
Parapets
$P_{Pt}=\{P_{Pt1}, P_{Pt2} ...... P_{PtN}\}$
Girders
$P_G=\{P_{G1}, P_{G2} ...... P_{GN}\}$
$D_S=\{P_P, P_{Cap}, P_D, P_{Pt}, P_G\}$
Phase 4 Original point cloud segmentation
Piers
PierCaps
Decks
Parapets
Girders
D_R

Fig. 2. Workflow of proposed DKRG framework.

voxel-based filter and then further aligned with the X-axis by principal component analysis (PCA), where the output data is leveraged as the original point cloud for semantic segmentation and denoted as D_R.The data D_R after space-based sampling is denoted as D_S. Notably, the proposed method is to perform the segmentation of data D_S at first, and the segmentation result will be adopted as a reference for the segmentation of the original point cloud D_R. Then, the bridge point cloud slicing process is performed to slice the D_S into a series of groups G_1, G_2, …, G_N (Section 4.1.2), The point in each group could be assigned a geometric feature and a DK-based point feature from the feature generation process (Section 4.1.3).

After data pre-processing, the point cloud with features in each group will be input in the segmentation workflow, including Phase 2-substructure segmentation (Section 4.2) and Phase 3-superstructure segmentation (Section 4.3). The bridge components are extracted in D_S following the order from piers (denoted P_P), pier caps (denoted P_{Cap}), decks (denoted P_D) to parapets (denoted P_{Pt}) and girders (denoted P_G). Specifically, in substructure segmentation, each group of point clouds starts by determining the presence or absence of pier caps; if present (Yes), it will output the point clouds of piers first and then output pier caps; if absent (No), it will output the point cloud of piers. Then, the remaining point cloud would be regarded as superstructures and input in superstructure segmentation. This process starts by determining the presence or absence of the sidewalk; if present (Yes), it will output the point clouds of decks first and then output the roadways and sidewalks simultaneously; if absent (No), it will output the point cloud of decks directly. The decks or sidewalks will be further used for cutting plane fitting and, subsequently, for chopping the remaining point clouds (girder assembly), which will output the point clouds of parapets and girders. Finally, the segmentation result of D_S could be used as the reference for the segmentation of D_R in Phase 4 (Section 4.4). It should be noted that pier caps and sidewalks may not exist in some bridge point clouds. If sidewalks exist in the bridge point clouds, the deck would be composed of the roadway (denoted P_{RW}) and the sidewalk (denoted P_{SW}).

4.1. Phase 1: data pre-processing

4.1.1. Down-sampling and alignment

Down-sampling is a crucial pre-processing step in point cloud segmentation, providing benefits such as computational efficiency, noise reduction, and simplification of complex scenes. This study involves two data down-sampling steps. The initial down-sampling process aims to reduce data redundancy by decreasing the data size while retaining important feature information. For this purpose, the widely used Voxel Grid Filter algorithm is adopted here to enhance the effectiveness and efficiency of the segmentation. The output data is further aligned with the X-axis by PCA, and the resulting point cloud is regarded as the original point cloud D_R for semantic segmentation and future BIM reconstruction.

Then, the space-based filter algorithm is employed for the down-sampling of D_R, primarily for two purposes: (1) to further reduce the number of points input in FEA software, enabling it to handle the heavy point clouds, and (2) to obtain uniformly distributed points, facilitating the further establishment of point connection and structural analysis. Thus, in this study, the space-based filter method is introduced to control the distance between any two adjacent points no more than the defined distance threshold (d). Finally, the output data D_S with a relatively small size and uniformly distributed points is used as input for the feature generation and further segmentation.

4.1.2. Bridge point cloud slicing

Bridge point cloud slicing is a commonly used procedure in the DK-based method, as demonstrated by Lu et al. [8] and Yan and Hajjar [16]. It can make bridge components' geometric and design features more easily distinguished from point clouds. Thus, a bridge-slicing method is proposed to chop the bridge point clouds D_S into a series of single-pier bridge groups ($G_1, G_2, \ldots, G_N$) to accentuate the characteristics of the bridge component vertical layout. As shown in Fig. 3, the proposed slicing method involves three steps: pier area detection (step 1), pier area marching (step 2), and cutting plane generation and bridge slicing (step 3).

Step 1: Pier area detection.

To determine the location of the pier area, the D_S is firstly voxelised on the X-Y plane, with a voxel size of 0.05 m. The result is a set of voxels denoted as $V = \{V_1, V_2, \ldots, V_n\}$, as shown in Fig. 3(b). The points in D_S is denoted as $D_S = \{P_1, P_2, \ldots, P_m\}$. Each voxel contains a different number of points. For example, the points in voxel i can be denoted as $V_i = \{P_{i1}, P_{i2}, \ldots, P_{ik}\}$. The number of points in the voxel V_i is denoted as $\text{num}(V_i)$ ($\text{num}(V_i) = k$ for voxel V_i). The Z-coordinate of points in voxel i is denoted as $Z_{V_i} = \{Z_{i1}, Z_{i2}, \ldots, Z_{ik}\}$. Then, the points in the bridge pier area can be clustered by collecting the voxels that meet the following two requirements:

- The voxel V_x with the number of points larger than the mean number of points of all voxels, as defined in **Eq. (1)**.

$$M = \left\{ V_x \,\middle|\, \text{num}(V_x) > \frac{1}{n}\sum_{i=1}^{n}\text{num}(V_i) \right\} \tag{1}$$

- The minimum Z-coordinate of the points in the voxel V_x is smaller than the height threshold Z_θ (m), as defined in **Eq. (2) and (3)**.

$$N = \{ V_x \mid min(Z_{V_x}) < Z_\theta \} \tag{2}$$

$$Z_\theta = \text{mean}\{Z_i \mid D_S\} - 4 \tag{3}$$

where the $\text{num}(V_x)$ is the number of points in the voxel V_i, and n is the total number of voxels. The $min(Z_{V_x})$ is the minimum Z-coordinate of the

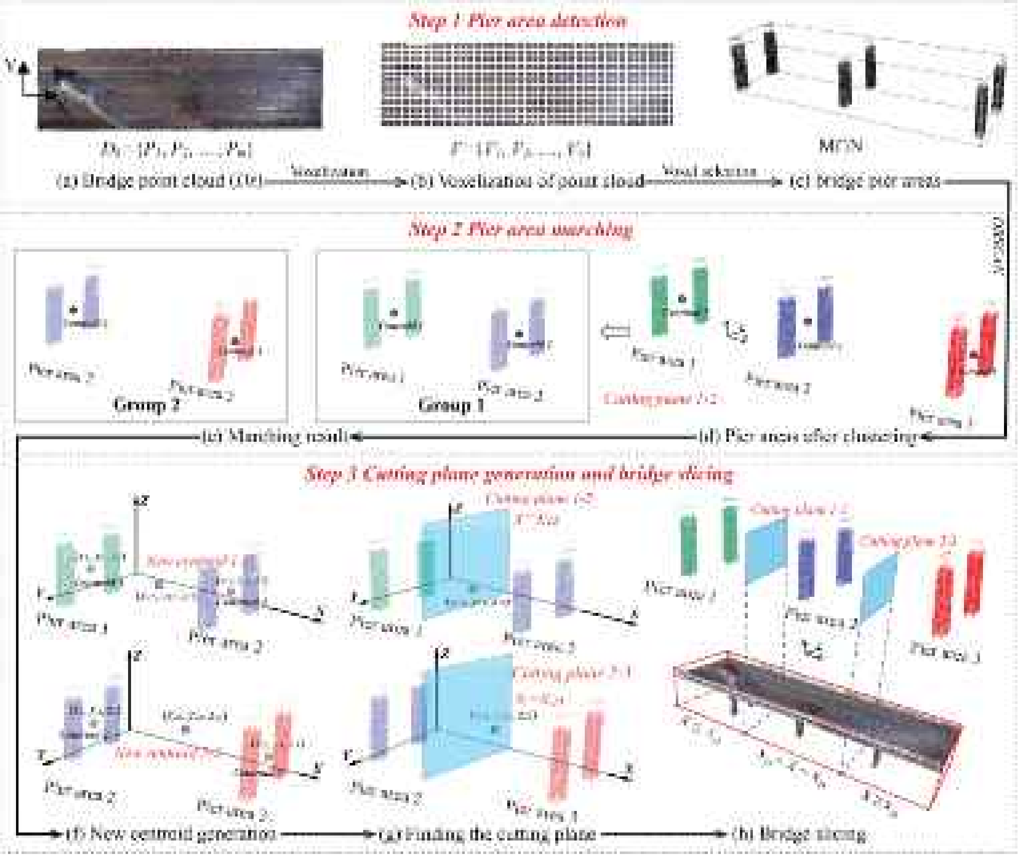

Fig. 3. Workflow for bridge slicing.

points in the voxel V_x. The Z_i is the Z-coordinate for point i in D_S.

The height threshold Z_θ is defined as 4 m lower than the mean height of all the points in D_S. This is because the voxels containing points belonging to piers exhibit a higher density from a top-down view and a relatively smaller minimum Z-coordinate. Finally, the points from pier areas can be calculated by the intersection of set M and set N, as shown in Fig. 3(c). Notably, for the determination of Z_θ, the mean height of the bridge point cloud should commonly be lower than the deck height and higher than the height of the bottom surface of girders. To make sure that only the voxel containing the points from the piers would be collected, the Z_θ should be lower than the maximum height of the pier. To achieve this, we have collected 20 real-world RC girder bridges and calculated the distance between the top surface of bridge decks and the maximum height of piers. The result indicated that this distance ranges from 1 m to 2 m for bridges without pier caps and 2 m to 4 m for bridges with pier caps. Thus, the Z_θ is defined as 4 m lower than the mean height of all the points in D_S.

Step 2: Pier area marching.

After the pier areas are determined, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [40] is adopted to group the bridge piers located in the same region of bridges along the longitudinal direction, as shown in Fig. 3(d). It should be highlighted that DBSCAN is primarily adopted here for distance-based clustering. Thus, for the threshold setting of the predefined radius (ε_D) and the minimal number of neighbour points (minPts), the ε_D should be larger than the distance between two adjacent piers in the bridge crossing direction and smaller than the distance between two adjacent pier groups along the bridge direction. The minPts can be set as a relatively larger value to remove potential noise points. In this study, the ε_D and minPts are set as 10 m and 200 points here, which can effectively cluster different pier areas. This is because the span of most RC girder bridge with pier is often longer than 10 m, and the distance between two adjacent piers in the same region is generally smaller than 10 m. Notably, the point clouds are typically separated after pier area detection, and different clustering algorithms can also be effectively applied to achieve the same results.

Since the cutting plane in Step 3 is calculated from two adjacent pier areas, we need to find all the adjacent pier areas. Thus, the centroid of each cluster is used to represent the pier area location, and the nearest neighbour of each pier area is determined by calculating the centroid distance of the targeted pier area with other pier areas. As shown in Fig. 3(e), there are two groups after the pier area marching: Group 1 - pier areas 1 and 2; Group 2 - pier areas 2 and 3.

Step 3: Cutting plane generation and bridge slicing.

For each group, the spatial location of the pier area is similarly represented by the centroid of point clouds. The generation of the cutting plane involves two steps: (1) new centroid generation and (2) finding the cutting plane, as shown in Figs. 3(f) and 3(g). For each marching group (e.g., Group 1), the new centroid point (x_{12}, y_{12}, z_{12}) is adopted to determine the location of the cutting plane, which is calculated by the mean coordinates of centroid 1 (x_1, y_1, z_1) and centroid 2 (x_2, y_2, z_2). Then, the cutting plane 1–2 can be determined by $x = x_{12}$ since the bridge point cloud has been aligned with the X-axis in advance through PCA. All the cutting planes will be generated by repeating the same procedure for all marching groups. Finally, the cutting planes are used to slice the whole bridge point clouds into a series of single-pier groups by different ranges of the X-coordinate, as shown in Fig. 3(h).

4.1.3. Feature generation

4.1.3.1. Normal estimation. The normal vector of points is a widely used feature in previous research, which can be used to distinguish different point segments based on the planarity of point clouds, especially in building scenes and other plane-dominated scenes [34–36]. In this study, the normal of points is estimated by the covariance matrix C in PCA, as defined by **Eqs.** (4) and (5). PCA involves performing an Eigenanalysis of the neighbourhood surrounding each point, from which the Eigenvector corresponding to the smallest Eigenvalue is considered as the normal of points.

$$\mathrm{C} = \sum\nolimits_{i=1}^{k} (\mathrm{P_i} - \overline{\mathrm{P}})^{\mathrm{T}} (\mathrm{P_i} - \overline{\mathrm{P}}) \tag{4}$$

$$\overline{\mathrm{P}} = \frac{1}{\mathrm{k}} \sum\nolimits_{i=1}^{k} \mathrm{P_i} \tag{5}$$

where k represents the number of points around the targeted point in a radius of ε. Since the point cloud may be polluted by noise, the radius for neighbour searching of each point in PCA is an important parameter that needs to be determined before normal estimation. In this study, the ratio between the second-largest Eigenvalue λ_2 and the smallest Eigenvalue λ_1 is adopted to assess the planarity of the neighbour points. The criterion $\lambda_2/\lambda_1 \geq 3$ is employed to automatically compute the smallest neighbour point searching radius (ε) for each point, which has been proved to have reliable segmentation performance in previous research [32].

4.1.3.2. DK-based point feature generation. This section will introduce the method of generating the DK-based point feature for substructure segmentation. Due to the fact that the point clouds obtained from current reality capture technologies can only cover the external surface of bridges, establishing an effective FEA model for response feature generation is a challenging problem. In this study, with the assumption that the points of a point cloud model can be treated as nodes in FEA models, we propose to connect all the points directly on the surface to form a frame structure for FEA instead of converting the point cloud into a solid model. This is primarily due to the following two reasons: (1) although the solid model can better reflect the mechanical characteristics of bridges, obtaining the solid model from point clouds remains a problem. The lower accuracy of generating a solid model may directly affect the final segmentation results, particularly for some irregular bridge components; (2) the focus of this study is to capture the differences in structural response among different bridge components, not the specific response value or the local response of any components. The beam element, therefore, is selected to simplify the modelling process and provide a better successful rate and computing efficiency.

However, the adoption of beam elements also involves some problems, such as (1) whether the structural response of the element can account for the characteristic of the bridge vertical layout and (2) whether the result of FEA would be affected by the absence of internal part of the bridge. Thus, we establish our analysis model under an important hypothesis: with the consideration of the absence of the internal part of bridges, the structural response of the element in the established frame structure can account for the characteristic of the bridge vertical layout to some extent, and the generated response feature can be effectively used for bridge point cloud segmentation.

The following features are observed from structural DK based on the vertical layout of bridge components. Inspired by **Features one** and **two**. This sub-section will present the process of generating DK-based point features from FEA, which primarily consists of three steps: (1) generation of structural analysis model, (2) FEA, and (3) feature translation. Fig. 4 provides a visual illustration of each step.

- **Feature one**: bridge components with a relatively lower position usually support the component with a relatively higher position against more gravity.
- **Feature two**: the boundary between different bridge components usually displays sharply changed cross-sections, such as the boundary between piers and pier caps as well as between pier caps and girders, which can result in a sharply changed structural response along the boundaries.

Step 1: Generation of structural analysis model.

The first step is to generate the connection information for all groups

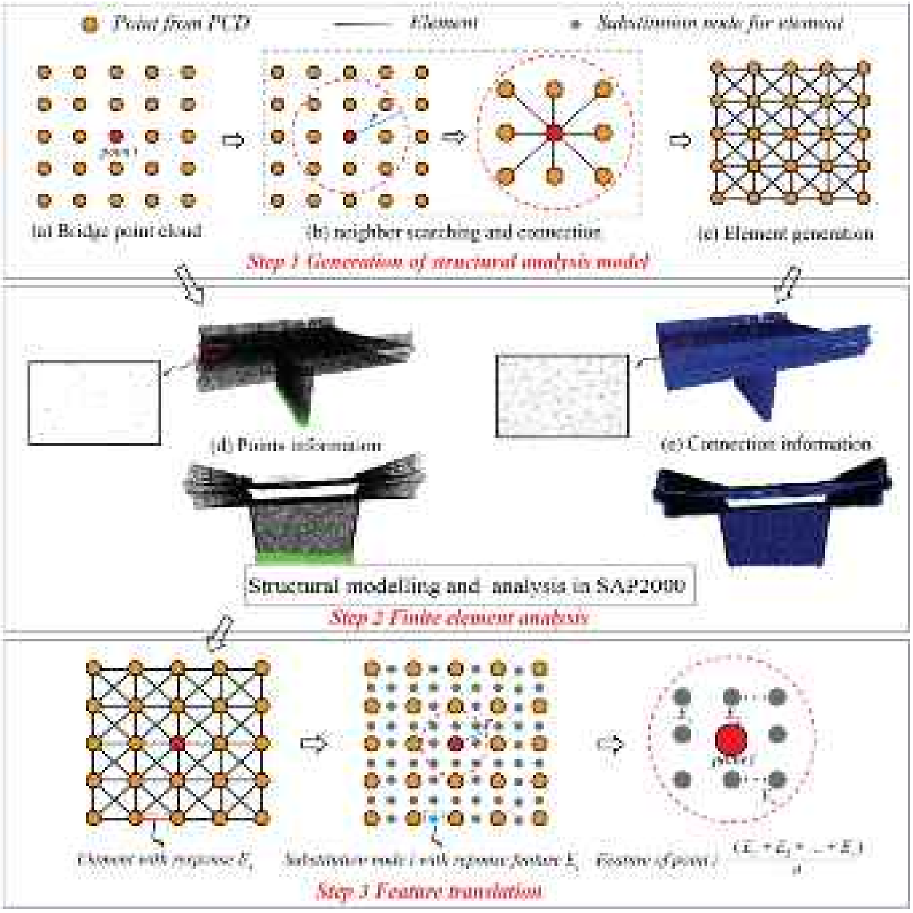
Point from PCD
Element
Substitution node for element
point i
(a) Bridge point cloud
(b) neighbor searching and connection
(c) Element generation
Step 1 Generation of structural analysis model
(d) Points information
(e) Connection information
Structural modelling and analysis in SAP2000
Step 2 Finite element analysis
Element with response E_i
Substitution node i with response feature E_i
Feature of point i
$\frac{(E_i + E_j + ... + E_n)}{n}$
point i
Step 3 Feature translation

Fig. 4. Structure modelling and generation of DK-based point feature.

($G_1, G_2, \ldots, G_N$). Here, we treat the points in point cloud data as nodes in FEA and assume that elements can be created by creating the connections between nodes. Specifically, for any point *i*, other points that lie within a predefined distance threshold (r) from point *i* are considered as neighbour points of point *i*. The connection is then generated between point *i* and each of its neighbour points, as shown in Figs. 4**(b)** and **(c)**. Applying the same procedure to all points will yield all connection information for the entire point cloud.

Step 2: FEA.

Once the connection information is determined, the points information (node information in FEA) and connection information (element information in FEA), as shown in Fig. 4**(d)** and **(e)**, can be input into the FEA software SAP2000 for structural modelling and analysis. The other settings in the model, under the unit system of (KN, m, C), are as follows: (a) each point is assigned a unit mass of 1; (b) the element material used is Q335 steel; (c) the element section is circular with a diameter of 0.1 m; (d) for boundary conditions, since the pier areas have been obtained in the bridge slicing process (Section 4.1.2), points within a distance of 0.1 m from the bottom of each pier are fixed in six degrees of freedom. The CSi Application Programming Interface (API) of SAP2000 is adopted to automate the structural modelling process. Finally, the structural model is analysed by only considering the self-weight effects, and the axial force of each element (E_i) will be output as a structural response feature for each element.

Notably, this paper focuses on the proof-of-concept of capturing structural response features from FEA. The commercial FEA software SAP2000 was selected for feature generation in this study because there is no tailored analysis software currently. The usage of different materials and cross-section information will affect the specific response value and not show much difference in the response regularity (e.g., Feature one and Feature two). As a result, the common element materials, element sections and element diameters are selected from the library in SAP2000. The elements with different diameters (0.1 m, 0.5 m, 1 m) have also been tested in our study, which can successfully construct a structural model for FEA. For the convergence of FEA, the likelihood of encountering convergence issues is low, especially when utilising beam elements in modelling. However, to address potential problems in more complex conditions, three suggestions are presented: (1) experimenting with smaller loads, such as assigning each point a mass of 0.1 instead of a unit mass of 1; (2) considering materials with higher yield stress; or (3) using elements with larger cross-sectional diameters to maintain the analysis within the elastic stage.

Step 3: Feature translation*.*

After identifying the response features of each structural element, we proceed to calculate the response feature of each point based on its surrounding elements. This approach enables the reflection of response features from elements in different components onto point features. The response for structural elements used in this research is axial force, as it effectively captures the two mentioned features. Moreover, since the feature difference originates from the vertical layout characteristic of the bridge in DK, we refer to the calculated point feature derived from FEA as the “DK-based point feature”.

The detailed process for translating the element response feature into an accessible DK-based point feature is presented in Fig. 4. Firstly, element (*i*) with the response feature (E_i) is replaced by a substitution node *i* located at the centre of the element. For any point *i* in the bridge point cloud, the response feature (f_i) is calculated as the mean response feature of the substitution nodes ($E_1, E_2, \dots, E_n$) located around it. The same distance (r) used in the element generation step is adopted here for neighbour searching. The calculation of the final structural response feature for point *i* (f_i) is defined by **Eq. (6)**.

$$f_i = \frac{1}{n}\sum_{j=1}^{n} E_j \tag{6}$$

where *n* is the number of substitution nodes around point *i*, and E_j is the response feature of substitution node *j*. In this way, the structural response feature can be assigned to all the points in bridge point clouds and directly utilised as the DK-based point feature in the RG algorithm.

Fig. 5 displays the example results of the generated DK-based point feature (f_i) and the Range (Δ_f) of the point feature for each point. For each point, the Δ_f is calculated by the difference between the maximum and minimum DK-based point feature in the neighbour points (further explanation can be found in Section 4.2.1) and can be used to measure the variation of the feature around the point itself. The overall distribution of the DK-based point feature and the Δ_f for each point in bridge point clouds are shown in **Fig. 5(a)** and **(b)**, respectively.

It can be found that the components in the substructure will resist more gravity than components in the superstructure, and the pier has the most remarkable response feature among these components, which aligns with **Feature one**. Meanwhile, the points with relatively larger Δ_f are primarily distributed in the boundary between piers and pier caps, which aligns with **Feature two**. Similarly, after removing the pier in point clouds, the pier cap tends to against more gravity than other components, and these points with relatively larger Δ_f are located around the boundary between the pier cap and the girder, as shown in Figs. 5 **(c)** and **(d)**. The same phenomenon can be seen in the overall distribution of the DK-based point feature and the Δ_f of each point along the bridge's vertical direction, as shown in **Figs. 6(a)** and **(b)**. Therefore, the characteristics of the bridge vertical layout can be recognised in the generated point feature, which can avoid directly interpreting DK as rules and constraints for point cloud segmentation. It is also worth highlighting that the two features are primarily defined for the segmentation of substructure, including piers and pier caps. This is because the DK-based point feature of substructures shows a larger structural response (**Feature one**) and more noticeable layout characteristics, such as the sharply changed cross-section between piers and pier caps and between pier caps and girders (**Feature two**). For superstructure segmentation, superstructure often exhibits more complex geometry features, such as complex girder types and the indistinguishable boundary between girders and parapets. This may pose more challenges for using DK-based point features for superstructure segmentation. However, the removal of substructures can significantly simplify the superstructure segmentation. The adaptation of the RG algorithm and other DK instead of the DK-based point feature for superstructure segmentation presents a better application range and simplified segmentation procedure.

In summary, the result of FEA would have some differences compared with the actual bridge but also share a certain degree of similarity because point clouds can capture the characteristics of the bridge vertical layout to some extent. As a result, the hypothesis mentioned above holds true, and the response feature of the element can account for the characteristic of the bridge vertical layout considering the absence of the internal part of bridges.

4.2. Phase 2: substructure segmentation

4.2.1. Design of RG algorithm

The proposed DKRG framework can incrementally group the bridge point cloud into a series of clusters with similar features. Multiple features, including the geometric feature and DK-based point feature, are used to enable component-level segmentation. Specifically, the normal angle of points is adopted as the basic measure to evaluate the planarity of point clouds, and the DK-based point feature is further utilised as the second criterion to adjust the threshold of the normal angle. This approach results in the determination of a weak threshold of normal angle (α_{weak}) for growing points within the same bridge components and a strict threshold (α_{strict}) for growing points belonging to different components.

The details of our RG algorithm are illustrated in **Algorithm 1**, where the following symbols are used: P (bridge point cloud), p_i (the location of i_{th} point in P), N (the normal of points), n_i (the normal of i_{th} point in P), F (DK-based point feature), f_i (the DK-based feature of i_{th} point in P), R (a set of point clusters output from our algorithm), U (unallocated points), Q (the queue of potential seed points), $R^{'}$ (allocated points after each growing), C (the neighbour points of seed point *i*), $\underline{C}$ (the neighbour points of point *m*, point *m* belonged to C).

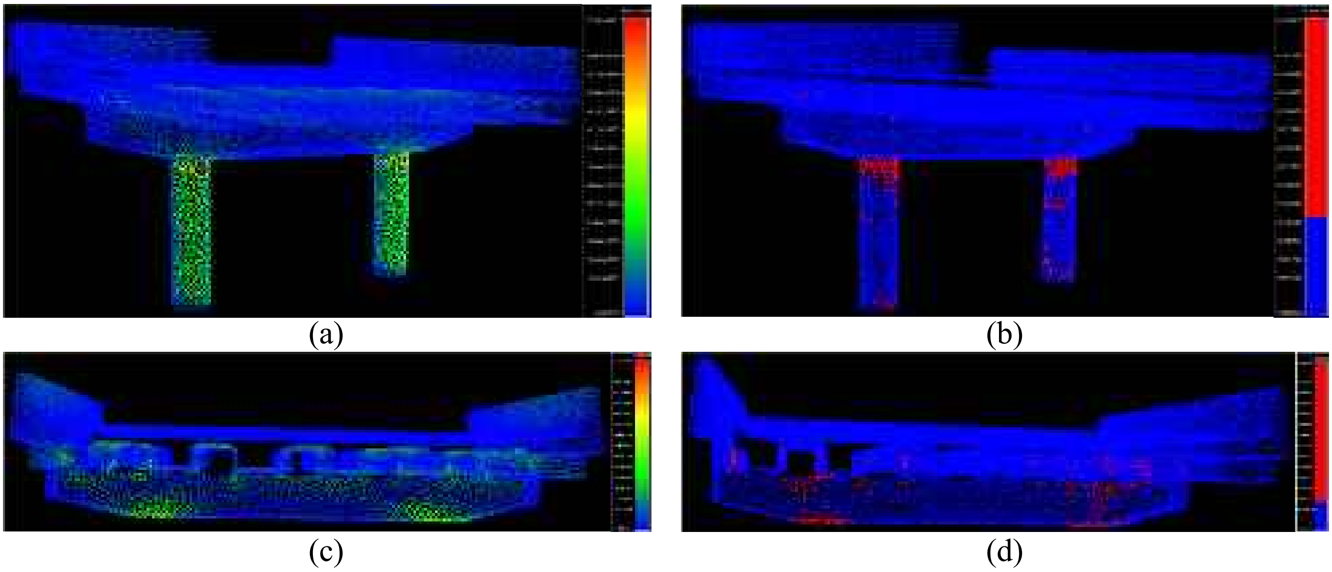

Fig. 5. Visualisation of (a) DK-based point feature; (b) the Δ_f of the feature for each point; (c) DK-based point feature after removing piers; (d) the Δ_f of the feature for each point after removing piers.

Algorithm 1. Region Growing for Substructure Segmentation

```
Input: Point cloud ($P$, $N$, $F$)
Output: Detected point cloud segments $R=\{R_1, R_2 \dots R_n\}$
Initialisation: $p, n, f \leftarrow p_i, n_i, f_i, U \leftarrow P, Q \leftarrow \emptyset$
while number of points in $R \leq 0.99$*number of points in $P$ do
    $R' \leftarrow \emptyset$;
    Seed index $j \leftarrow$ random selected from $U$;
    $Q \leftarrow j, R' \leftarrow j$;
    while $Q \neq \emptyset$ do
        seed index $i \leftarrow$ random selected from $Q$;
        remove $i$ from $Q$;
        $C \leftarrow \{p \mid \|p - p_i\|_2 \leq \varepsilon\}$;                /* sped up by kd-tree */
        for each point $p_m \in C$ do
            $\underline{C} \leftarrow \{p \mid \|p - p_m\|_2 \leq \varepsilon'\}$;         /* $\Delta_f$=max($f_{\underline{C}}$) – min($f_{\underline{C}}$) */
            if $\nabla_f \leq \theta$ and $\Delta_f \leq \mu$ then            /* $\nabla_f$=abs( ($f_i - f_m$) / $f_i$ ) */
                $\alpha = \alpha_{weak}$;
            else
                $\alpha = \alpha_{strict}$;
            end if
            if arccos($n_m^T \bullet n_i$) $\leq \alpha$ then
                insert $p_m$ into $R'$ ;
                insert $p_m$ into $Q$;
            end if
        end for
    end while
    if $R'$ has more than M points then
        insert $R'$ into $R$;
        remove $R'$ from $U$;
    else
        set all points in $R'$ unallocated;
    end if
end while
return $R$;
```

Two distance thresholds, denoted as ε and ε', are used in the algorithm. The ε is the growing radius of the seed point for the neighbour searching and shares the same value for normal estimation. This means that ε can be automatically determined in the feature calculation step. The ε' is the radius for calculating the Δ_f associated with each point in cluster C (the neighbour points of seed point i). For example, if the point i is the seed point, and then the neighbour point cluster C can be determined by radius ε. For points in C, such as point m, the radius ε' is used to determine the neighbour points of point m, denoted as $\underline{C}$. The Δ_f serves to measure the variation of feature F for points in the cluster $\underline{C}$, and is the variation feature of point m. If a large Δ_f for point m is

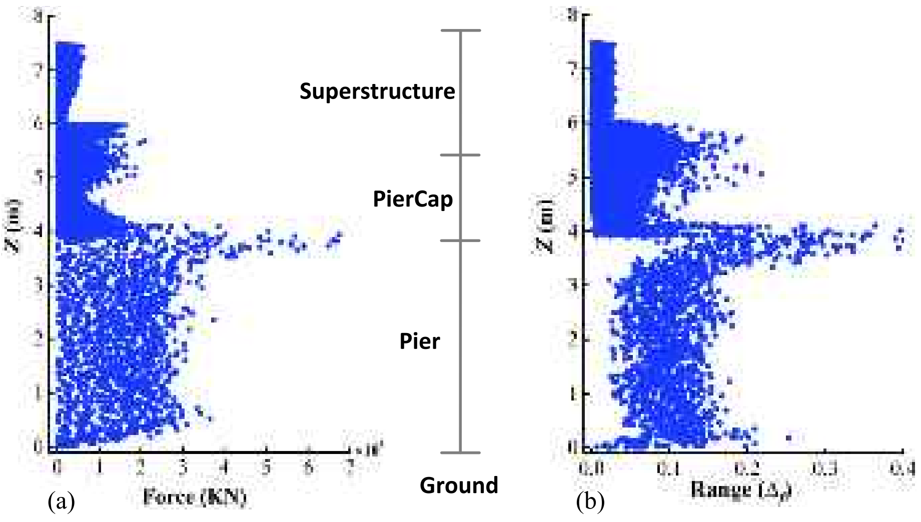

Fig. 6. Distribution of (a) the DK-based point feature and (b) the Δ_f of each point along the bridge vertical direction.

observed, the point *m* is likely to be located around the bridge component boundary, and a strict threshold will be used for growing. The parameter Δ_f is defined by **Eq. (7)**.

$$\Delta_f = max\left(f_{\underline{C}}\right) - min\left(f_{\underline{C}}\right) \tag{7}$$

where $f_{\underline{C}}$ is the feature *F* of points in $\underline{C}$. The ∇_f is the gradient of the DK-based point feature calculated by **Eq. (8)**.

$$\nabla_f = \text{abs}\left((f_i - f_m)/f_i\right) \tag{8}$$

where f_i is the feature *F* of seed point *i*, and f_m is the feature *F* of point *m* (point *m* is the neighbour point of point *i*).

In this study, the points with Δ_f larger than threshold μ and ∇_f larger than θ are considered to be likely situated around the boundary between two bridge components. The relatively strict angle threshold (α_{strict}) is adopted for growing, enabling over-segmentation for bridge point clouds and retaining the component boundary. Conversely, the points that fail to meet these two conditions are likely to be unlocated at the boundary of bridge components and a relatively weak threshold (α_{weak}) is correspondingly used to cluster the points in the same components. After the loop of each growing, the number of points in R' larger than M would be added into R as a segment; otherwise, it would be regarded as unallocated points for further growing.

It should be noted that, in regular RG-based plane segmentation, a plane fitting is usually present to refine the growing process or seed point selection. However, this process is removed in our method because non-plane growing results are expected. Correspondingly, two parameters are introduced to refine the growing process. The first parameter ∇_f is used as the basic rule to measure the variation trends of feature F. The parameter Δ_f is adopted as the second rule to estimate the feature fluctuation degree in the region around each point, which can avoid using a larger threshold α_{weak} for the growing of the points that may be located at the boundary.

In addition, RG-based point cloud segmentation usually involves an edge-point refinement step since the points close to edges (the points close to the component boundary) are usually not part of any segments. An edge-refinement strategy introduced by Poux et al. [32] is adopted in this study, where all the unallocated points will be assigned to the segment that it is found closest to. In this way, all the points can be clustered into their corresponding groups at the end of our algorithm.

4.2.2. Pier segmentation

As mentioned in Fig. 1, bridge piers in vertical layouts primarily include the following three situations: (1) the pier connects with the girder directly; (2) the pier connects the girder through bearings; (3) the pier connects with the pier cap.

On the one hand, the cross-section often sharply changed in the boundary of the pier with other bridge components in the above three cases, which will result in a noticeable variation of structural response features. On the other hand, the pier tends to resist more gravity compared to other bridge components, leading to more pronounced gradient variations at the borders between the pier and other bridge components. Therefore, the bridge pier can be detected before the pier cap in this study. Fig. 7 displays the growing result of example bridges with different vertical layouts, where the entire bridge point cloud can be grouped into two clusters by employing specific thresholds, namely θ_{Pier} and μ_{Pier}.

Since the significant gradient of the DK-based point feature occurs at the boundary of the pier with other bridge components, the points with a ∇_f smaller than θ_{Pier} and Δ_f smaller than μ_{Pier} can be assumed to be away from the boundary, and the relatively larger α_{weak} will be used for region growing, which is defined by **Eq. (9).**

$$\text{If } \nabla_f \le \theta_{Pier} \text{ and } \Delta_f \le \mu_{Pier} \tag{9}$$

then,

$$\alpha = \alpha_{weak}$$

otherwise,

$$\alpha = \alpha_{strict}$$

Finally, the pier areas can be detected by recognising the segments with relatively lower centroids. It should be noted that other rules might also be designed to achieve the distinction of bridge components from the segmentation result. For instance, the pier areas can also be detected by choosing the segments with fewer points. After the successful segmentation of piers $P_P = \{P_{P1}, P_{P2}, \ldots, P_{PN}\}$, the remaining point cloud will be used for further component segmentation.

4.2.3. Pier cap segmentation

The remaining bridge point clouds with a pier cap, as shown in Figs. 7**(a)** and 7**(c)**, will serve as input for further pier cap segmentation. Once the pier areas have been removed from point clouds, the boundary between the pier cap and girder will display the significant variation gradient of DK-based point features.

Similarly, the rule for adjusting the threshold α is defined by **Eq. (10)**. Compared with pier segmentation, points with a distance smaller than 0.2 m to the lowest point (Z_{min}) in the remaining point clouds will also undergo growth with a relatively larger α_{weak} in pier cap segmentation. This aims to reduce the influence of the DK-based feature located in the boundary between piers and pier caps on the segmentation of the pier cap.

$$\text{If } \left(\nabla_f \le \theta_{Piercap} \text{ and } \Delta_f \le \mu_{Piercap}\right) \text{ or } Z_{point} \le Z_{min} + 0.2 \tag{10}$$

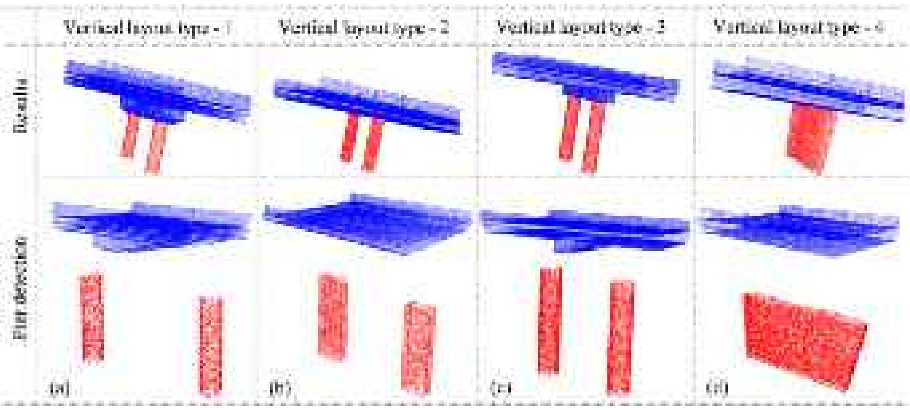

Fig. 7. Pier segmentation in four vertical layout types: (a) bridges with bearing and pier cap; (b) bridges with bearing but without pier cap; (c) bridges with pier cap but without bearing; (d) bridges without pier cap and bearing.

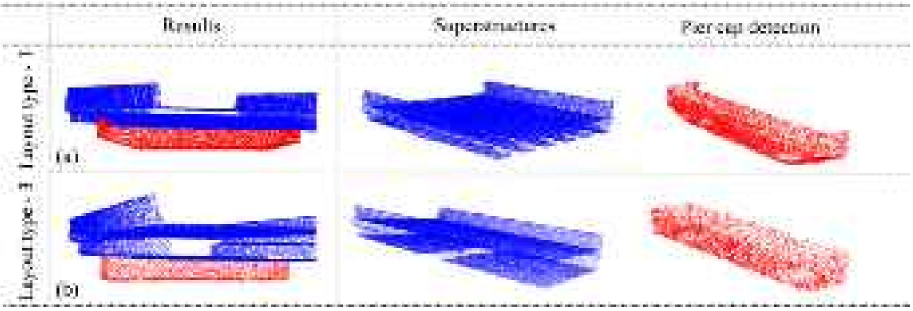

Fig. 8. Pier cap segmentation: (a) with bearing and pier cap; (b) with pier cap but without bearing.

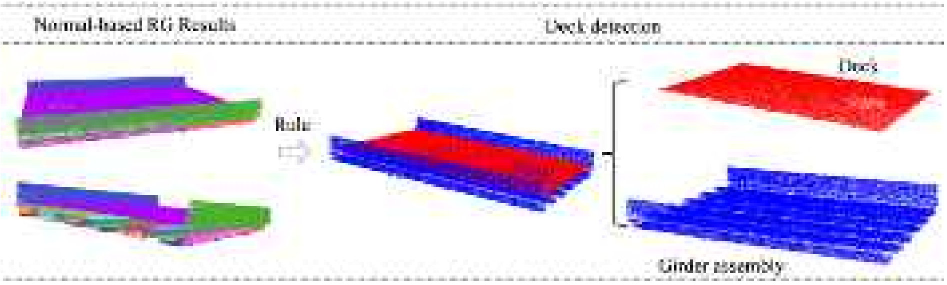

Fig. 9. Deck component segmentation.

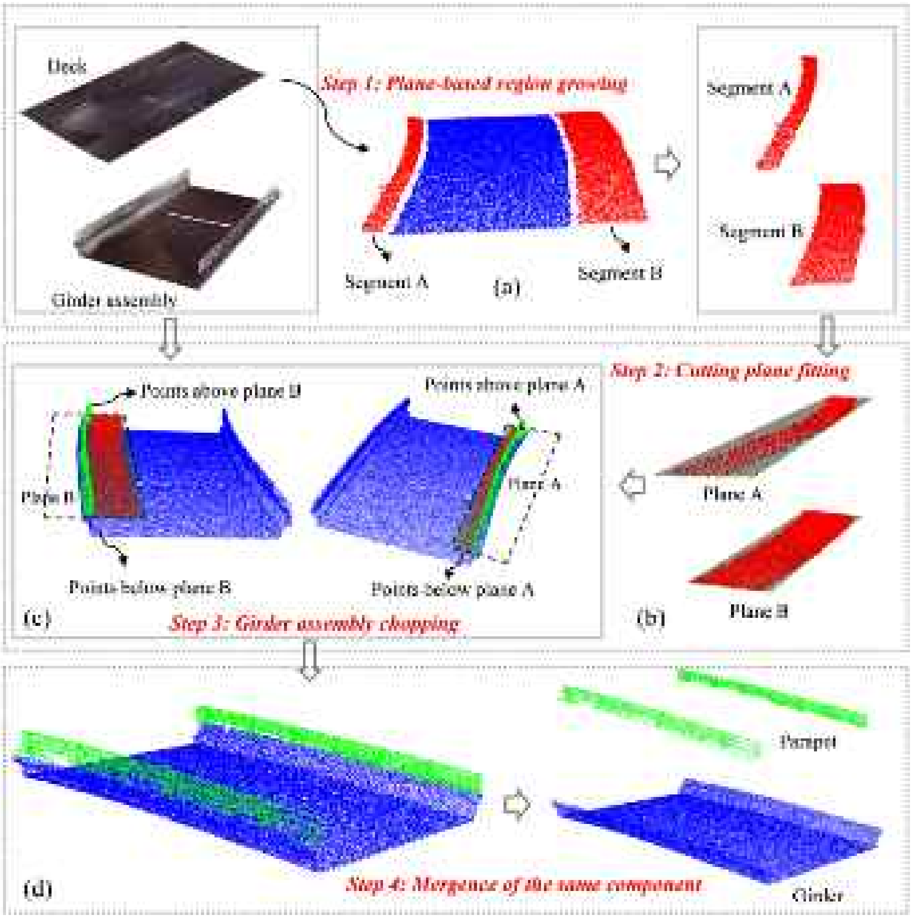

Fig. 10. Workflow for girder and parapet segmentation.

then,

$\alpha = \alpha_{\text{weak}}$

otherwise,

$\alpha = \alpha_{\text{strict}}$

The examples of growing results for pier cap segmentation are shown in Fig. 8. The points belonging to the pier cap can be similarly detected by selecting the segment with a relatively lower centroid point $\text{P}_{Cap} = \{\text{P}_{\text{Cap1}}, \text{P}_{\text{Cap2}}, \ldots, \text{P}_{\text{CapN}}\}$. The remaining point clouds are regarded as the superstructure for further processing.

4.3. Phase 3: Superstructure segmentation

After removing the substructure in bridge point clouds, the layout characteristics of bridge components in the superstructure are more pronounced, which can significantly simplify the segmentation tasks for superstructural component segmentation from point clouds. Thus, the following bridge layout features can be observed and adopted in this stage.

- **Feature three**: The deck is the component that often has the biggest planar area with a relatively higher position in the superstructure.
- **Feature four**: The parapet is usually located on the deck or connects the girder with a relatively lower position below the deck.

4.3.1. Deck segmentation

For detection of deck components denoted as $\text{P}_\text{D} = \{\text{P}_{\text{D1}}, \text{P}_{\text{D2}}, \ldots, \text{P}_{\text{DN}}\}$, a normal-based RG [32] is first leveraged to identify all planes in the superstructure, where the threshold α_{strict} is adopted to evaluate the angle between the normal of two points. This can enable the sufficient over-segmentation of superstructure point clouds and result in a number of planar segments, as shown in Fig. 9. After that, two segments with the largest number of points would be selected and labelled as R_1 and R_2, respectively. The number of points in R_1 is larger than in R_2, and Z_1 and Z_2 represent the height of the centroids for segment R_1 and R_2, respectively. Then, **Feature three** can be used to guide the segmentation of deck components from RG-based results, as defined by **Eq. (11)**.

$$\text{If number}(R_2)/\text{number }(R_1) \geq 0.8 \text{ and } Z_2 \geq Z_1 \tag{11}$$

then,

Deck ← $\text{R}_{2.}$

otherwise,

Deck ← R_1

where *number(*R_2*)* and *number* (R_1*)* are the number of points in segments R_1 and R_2, respectively. Due to the fact that the deck in the superstructure point cloud often occupies the largest plane area (**Feature three**), the deck in bridge point clouds often contains more points than other bridge components. Thus, R_1 is generally classified as a deck component based on this criterion. However, for the consideration of bridges with narrow decks or relatively larger bottom surfaces of girders, the R_2 is only considered as a deck when it reaches 80% of the size of R_1 and has a higher centroid position. The remaining point cloud is regarded as the girder assembly for the further segmentation of the girder and parapet.

4.3.2. Girder and parapet segmentation

After the detection of decks, the spatial relationship of deck, parapet, and girder, as mentioned in **Feature four**, is employed for detecting the girder and parapet from the girder assembly. This process primarily involves four steps: (1) Further RG; (2) Cutting plane fitting; (3) Girder assembly chopping; (4) Mergence of the same component. The detailed information is presented in Fig. 10.

Step 1: Further RG

This step aims to detect the sidewalk P_{SW} that will be used for plane fitting and chopping of the girder assembly. A normal-based RG [32] with a smaller normal angle threshold ($0.5\alpha_{\text{strict}}$) is adopted to segment the deck point clouds further. The example of a growing result is shown in Fig. 10(a). Notably, if only one segment is found in growing results, we will assume that no sidewalk exists in the bridge and the P_D will only contain the point clouds from P_{RW}. Conversely, if multiple segments are detected, it indicates the existence of sidewalks. The P_D consist of P_{RW} and P_{SW}. Then, the P_{SW} can be recognised according to their relatively smaller sizes and lower positions compared to the roadway, such as Segment A and Segment B in Fig. 10(a).

Step 2: Cutting plane fitting

Once the sidewalk or deck is identified, the Random Sample Consensus (RANSAC) algorithm [41] is adopted for plane fitting due to its better robustness for noise points. As shown in Fig. 10(b), the planes

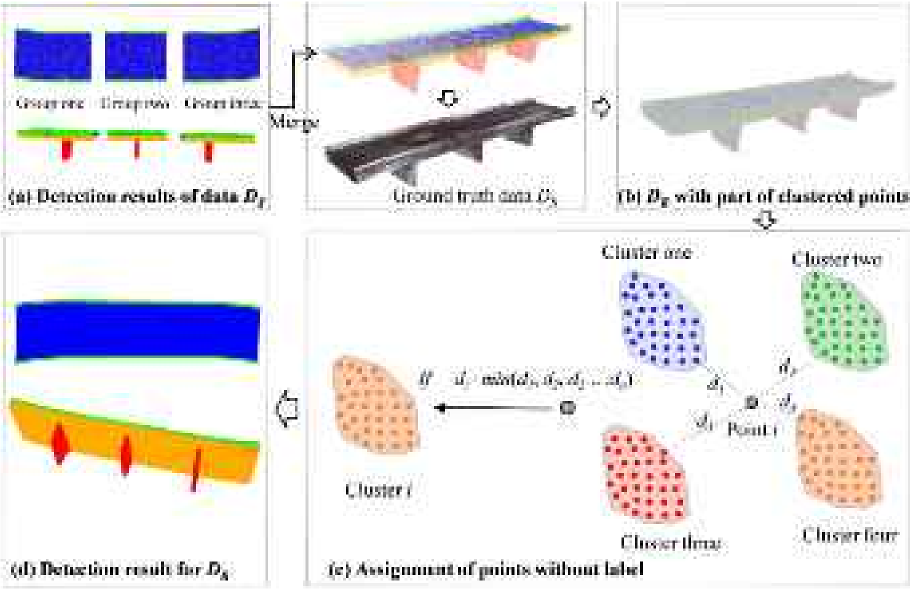
Group one
Group two
Group three
Merge
(a) Detection results of data D_T
Ground truth data D_S
(b) D_R with part of clustered points
Cluster one
Cluster two
If $d_i = \min(d_1, d_2, d_3 \ldots d_n)$
d_1
d_2
d_3
d_4
Point i
Cluster i
Cluster three
Cluster four
(d) Detection result for D_R
(c) Assignment of points without label

Fig. 11. Method for the segmentation of original point cloud D_R.

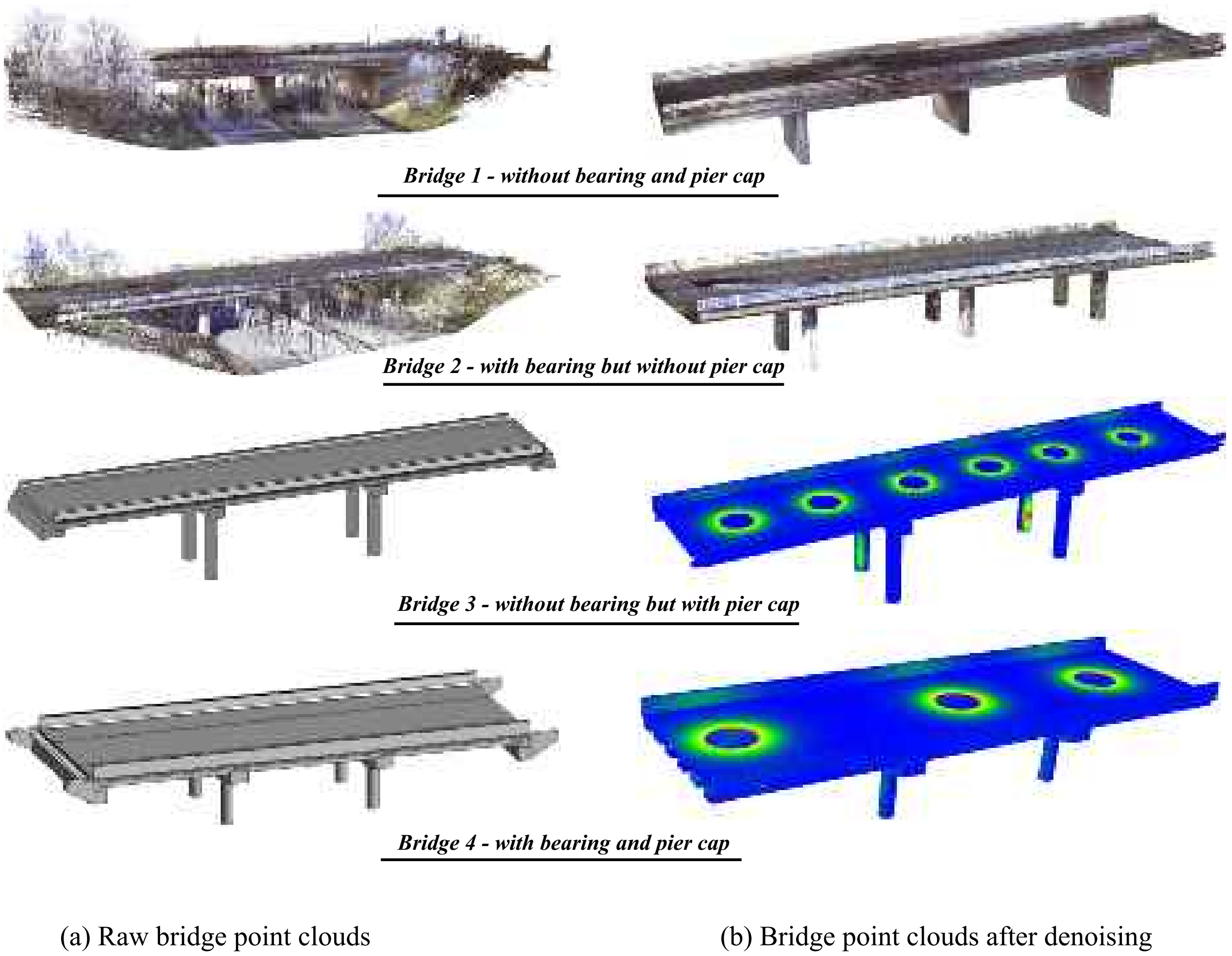
Bridge 1 - without bearing and pier cap
Bridge 2 - with bearing but without pier cap
Bridge 3 - without bearing but with pier cap
Bridge 4 - with bearing and pier cap
(a) Raw bridge point clouds
(b) Bridge point clouds after denoising

Fig. 12. Point cloud of four bridges in experimental tests.

A and B can be obtained from the plane fitting of segments A and B.

Step 3: Girder assembly chopping

Step 3 involves the process of girder assembly chopping. Each cutting plane (Plane A and Plane B) is utilised to chop the girder assembly into two groups: points above and below the cutting plane, as shown in Fig. 10(c). The sidewalk on each side is specifically used to obtain the parapet of the corresponding side from the chopping result. To achieve this, DBSCAN [40] is applied to group the points above the cutting plane, and the parameter setting of the ε_D and minPts are 2 m and 200 points, respectively. The parapet on the same side can be detected by finding the nearest cluster for each corresponding sidewalk.

Step 4: Mergence of the same component

Finally, the points of the girder P_G and parapets P_{Pt} would be successfully labelled and detected from the girder assembly after merging the points belonging to the same component, as shown in Fig. 10(d).

4.4. Phase 4: Original point cloud segmentation

Once the segmentation of each group (G_1, G_2, …, G_N) is completed, as shown in Fig. 11(a), we merge the points with the same semantic label and output the final segmentation result of the down-sampled bridge point cloud D_S. This means that part of the point cloud in D_R has been assigned different semantic labels, including pier, pier cap, deck, girder, and parapet. As shown in Fig. 11(b), the hybrid point cloud consists of points with labels and unclassified points (grey). Therefore, the challenge of original point cloud segmentation can be translated into the problem of how to assign labels for the unclassified points based on the labelled points around them.

Fig. 11(c) shows the process of assigning labels to each unclassified point. For point *i*, the process first involves calculating the closest distances between point *i* and each point cluster (e.g., cluster one, cluster two, cluster three, and cluster four), and the result is denoted as d_1, d_2, …, d_n. Then, the label of point *i* is determined by finding the cluster with the minimum distance to point *i*. Thus, point *i* would be assigned with the same label as the nearest cluster. This process continues iteratively until all the unclassified points in the hybrid point cloud have been given a label. The final segmentation result of the original point cloud is shown in Fig. 11(d).

5. Experiment and results

5.1. Data preparation

To validate the proposed method, four bridges with different vertical layouts corresponding to Fig. 1 are adopted, as shown in Fig. 12. The background of these bridge point clouds was manually removed beforehand. Then, a voxel-based filter algorithm was applied to reduce the redundancy of raw point clouds. The filtered data is assumed to be the original point cloud D_R for further segmentation and performance evaluation. Notably, the abutment components are excluded and not considered in the current method. This is because the point cloud of abutments often has serious data quality issues due to occlusions of the surrounding environment. The segmentation of abutments might be treated as the problem of complex component segmentation in future research.

For the dataset, bridge 1 and bridge 2 are both concrete slab-beam bridges from the real-world dataset collected by Lu et al. [42] (3_{th} and 6_{th} bridge of the database). The Heidelberg LiDAR Operations Simulator (HELIOS++) [43–45], an open-access laser scanning simulation platform, was adopted here for synthetic point cloud generation of bridge 3 and bridge 4, where the terrestrial laser scanning (TLS) equipment (RIEGL VZ-400) is adopted in this study. The bridge 3 is a slab bridge with a pier cap, and bridge 4 is a multi-box beam bridge with pier caps and bearings.

5.2. Implementation

We implemented the proposed framework using Python and ran it on a desktop computer (Intel Core i7–12,700 2.10 GHz CPU, 32 GB RAM). The structural response feature was analysed in SAP2000 v24.0.0 Ultimate, and the CSi API tool was adopted to automate the modelling process.

5.3. Parameter selection

The RG-based method usually involves a set of parameters that may be automatically or manually determined. In this study, the parameter selection was needed in two stages: data pre-processing and component segmentation.

In the data pre-processing stage, we employed the widely used voxel width of 0.03 m for the voxel-based down-sampling process [13,26], which can effectively retain the geometric features of the point clouds while reducing data redundancy. The space-based down-sampling with a fixed value of 0.1 m (parameter d in section 4.1.1.) was adopted to reduce the number of input points and ensure evenly distributed point clouds. Based on the space distance of parameter d, the double distance of 0.2 m (parameter r in section 4.1.3.2) was used in the neighbour searching for connection generation and feature translation. For estimating the normal of point clouds, the radius ε for neighbour searching can be automatically determined by eigenanalysis. The parameter ε was also adopted for the neighbour searching of seed points in RG.

For the component segmentation stage, three fixed parameters were determined beforehand, including the strict angle threshold (α_{strict}), radius (ε') for Δ_{f} calculation, and the minimum number of points in segments (M). The basic angle threshold α_{basic} is set as 8 degrees, which has been found to successfully over-segment the four bridges in our study, and the 4 degrees ($0.5\alpha_{\mathrm{strict}}$) is used to detect the sidewalk. The radius for Δ_{f} the calculation is set as 0.2 m. The minimum number of points in segments is set as 200 points due to the component-level growing results. Additionally, there are still three parameters that need to be determined from parameter tuning: the weak angle threshold (α_{weak}), gradient threshold (θ), and Range threshold (μ). Detailed information for the three parameters in terms of the four bridges is shown in Table 1. Notably, the DK-based point feature F is normalised in this study. For the strategy of parameter tuning, the range of θ and μ can be roughly determined by referring to the distribution of Δ_{f} and ∇_{f}. Then, the further parameter tuning of θ and μ will enable the distinguish of components boundary, and meanwhile, incrementally increasing the value of α_{weak} to enable sufficient growth in the same bridge components. The parameter tuning process will stop until the successful segmentation of target bridge components and sufficient growth in other bridge components.

Table 1
Parameter settings in the experiment.

	Pier			Pier cap		
	α_{weak} (°)	θ	μ	α_{weak} (°)	θ	μ
Bridge-1	21	0.1	0.2	–	–	–
Bridge-2	50	0.2	0.05	–	–	–
Bridge-3	24.5	0.1	0.04	25	0.08	0.02
Bridge-4	20	0.1	0.2	20	0.05	0.05

5.4. Results and evaluation

To estimate the performance of the proposed solution, the following point-level performance metrics are adopted in this section [46], including Precision (Pr), Recall (R), Intersection over Union (IoU), and mean Intersection over Union ($mIoU$). These metrics are defined as follows:

$$Pr = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{12}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{13}$$

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{14}$$

$$\mathrm{mIoU} = \frac{1}{N}\sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{(\mathrm{TP})_{\mathrm{i}}}{(\mathrm{TP})_{\mathrm{i}} + (\mathrm{FP})_{\mathrm{i}} + (\mathrm{FN})_{\mathrm{i}}} \tag{15}$$

where N is the total number of component types in bridge point clouds, and the TP, FP, and FN are the number of correctly labelled points (e.g., the point from the pier is assigned with a pier label), the number of wrongly labelled points (e.g., the point from the pier is assigned with a label that not pier), and the number of miss-labelled points (e.g., the point not belonged to pier component is assigned with a pier label), respectively.

For the pre-processing phase, Fig. 13 illustrates the results of pier location determination and bridge point cloud slicing. The figure shows that the pier locations are successfully detected from the point cloud $\mathrm{D_S}$, and the entire bridge point cloud is also chopped into a series of single-pier groups for further segmentation. The overview of the segmentation results of data $\mathrm{D_S}$ and $\mathrm{D_R}$ is shown in Fig. 14, and Fig. 15 further shows the result in more detail for different bridge components for data $\mathrm{D_R}$. In general, the different bridge components, including piers, pier caps, girders, deck, and bridge parapets, are successfully detected from bridge point clouds, and some mislabelled points can also be found along the boundary of adjacent bridge components.

For quantitative evaluation, the point-level evaluation results of data $\mathrm{D_S}$ and data $\mathrm{D_R}$ are presented in Table 2 and Table 3, respectively. Overall, our method demonstrates excellent segmentation performance, which can be evident from the various evaluation metrics. The mean precision, recall and IoU for data $\mathrm{D_S}$ are calculated as 97.96%, 98.65% and 96.64%. The mIoU values for four bridge point clouds range from 95.55% as the lowest to 97.32% as the highest. For data $\mathrm{D_R}$, the mean precision, recall and IoU for data $\mathrm{D_S}$ are calculated as 98.30%, 98.53% and 96.87%. the mIoU values for four bridge point clouds vary from 95.47% as the lowest to an impressive 97.75% as the highest. The high mIoU values indicate that the proposed solution successfully achieves the accurate segmentation of different bridge components. It shows that our method performs consistently well across various component types in both the sampled data and the original point clouds.

Since the segmentation of $\mathrm{D_R}$ is based on the segmentation result of data after sampling, the mIoU for each component type in $\mathrm{D_S}$ and $\mathrm{D_R}$ is illustrated in **Fig. 16**. The figure shows that the mIoU for the pier, girder, and deck are around 98% both in $\mathrm{D_S}$ and $\mathrm{D_R}$, which indicates the reliable segmentation result of the proposed method. However, the segmentation results exhibit relatively lower accuracy for the pier cap and parapet components, with the minimum mIoU still being higher than 93.40%. This observation is consistent with findings from previous research for RC bridges [13,26].

To better understand the phenomenon, the points with incorrect semantic labels (from both false positives-FP and false negatives-FN) are displayed in Fig. 17. The figure shows that these incorrectly labelled points are primarily distributed along the boundary of bridge components and are less frequently found in other regions. In general, this problem is primarily due to the strategy of allocating the unclassified

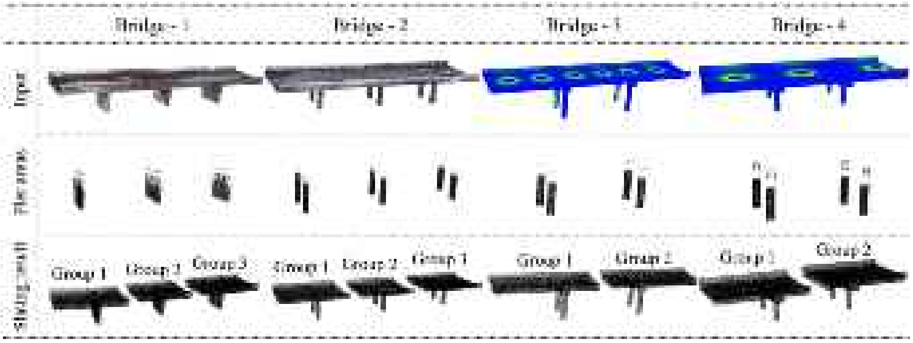

Fig. 13. Results of pier location determination and bridge point cloud slicing.

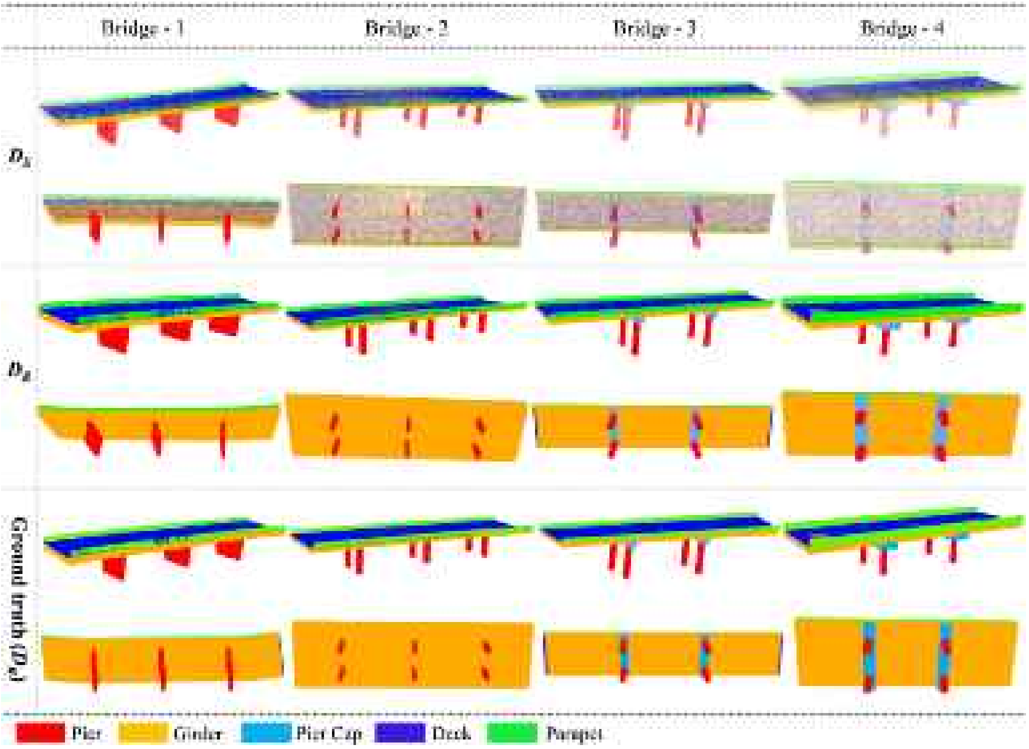

Fig. 14. Segmentation results of data D_S and D_R.

points and the precision of the fixed length for neighbour searching and element generation. The allocation of unclassified points is a challenging problem in the RG algorithm, which needs more research in this field. For the precision of the fixed length, the larger value may lead to misaligned connections around boundaries and the response feature with low precision. To improve the boundary point segmentation and avoid other potentially misaligned connections, two potential solutions will be investigated in our future research: (1) adopting an adaptive length for neighbour searching and element generation, which involves using varying lengths based on local conditions. For instance, employing a smaller length when there are sharp changes in the normal or curvature of points within a region and a larger length in areas of moderate change; (2) obtaining neighbouring relations beforehand to guide subsequent element generation. It involves initially over-segmenting the raw point clouds into a series of patches. Subsequently, constructing neighbouring relations among these patches facilitates the prevention of misaligned connections by eliminating connections between points that are not from the same patches or neighbouring patches.

For the sensitivity of different bridge components to boundary segmentation, the parapet often has more boundary regions since it connects the deck and girder along the longitudinal direction of the bridge. This characteristic may lead to unsatisfactory segmentation results. Regarding the pier cap, it can be found that the segmentation results for bridge 3 without bearings exhibit better segmentation performance compared to bridge 4 with bearings. The relatively low accuracy for bridge 4 may be attributed to miss-labelled points from the bearing, as shown in Fig. 17. As a possible solution, future research focusing on detecting complex components like bearings may lead to improved segmentation performance for pier caps. By examining the distribution of incorrectly labelled points and understanding the challenges posed by certain bridge components, potential areas for enhancement and optimisation can be identified for further research and development.

Furthermore, the comparison of segmentation performance for $\mathrm{D_S}$ and $\mathrm{D_R}$ is shown in Fig. 18. The mIoU for four bridges and different bridge components are shown in Figs. 18**(a)** and 18**(b)**, respectively. The ΔmIoU is defined as follows:

$$\Delta \mathrm{mIoU} = \mathrm{mIoU}_{DR} - \mathrm{mIoU}_{DS} \tag{16}$$

where the mIoU_{DR} and mIoU_{DS} represent the mIoU for data D_R and data

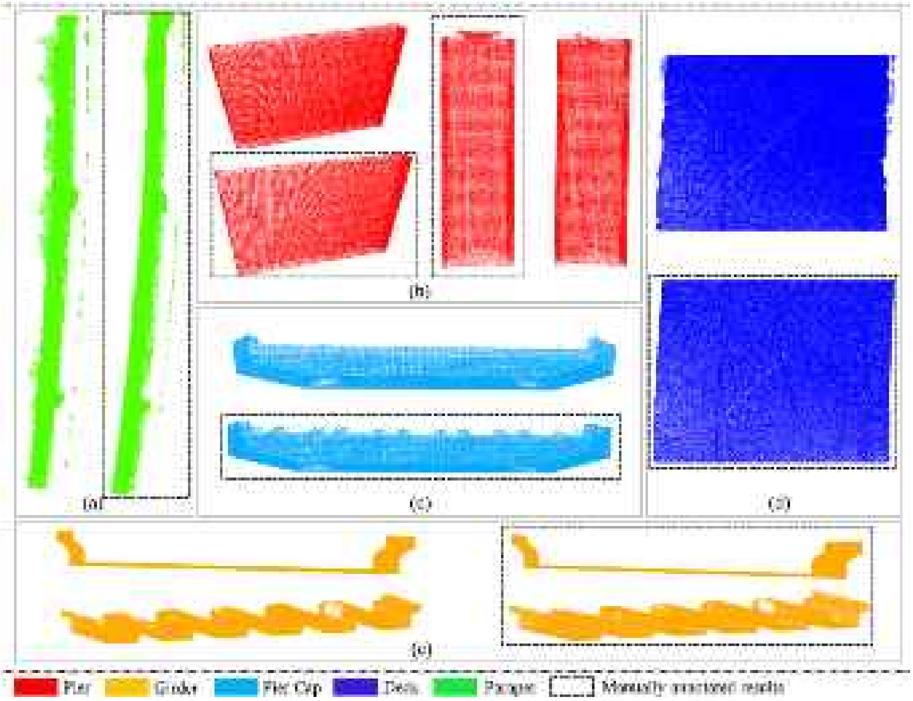

Fig. 15. Visualisation of the segmentation results of different bridge components.

Table 2
Point-based evaluation for sampled point clouds (D_S).

Bridge type	Component	TP	FP	FN	Precision	Recall	IoU	mIoU
Without bearing and pier cap (Bridge 1)	Pier	16,492	538	3	96.84%	99.98%	96.82%	95.55%
	Girder	46,243	137	578	99.70%	98.76%	98.48%	
	Deck	37,032	229	1044	99.39%	97.26%	96.68%	
	Parapet	11,246	1110	149	91.02%	98.69%	89.93%	
With bearing but without pier cap (Bridge 2)	Pier	8884	278	3	96.97%	99.97%	96.93%	97.32%
	Girder	51,631	138	311	99.73%	99.40%	99.14%	
	Deck	36,428	344	198	99.06%	99.46%	98.53%	
	Parapet	10,643	242	356	97.78%	96.76%	94.68%	
Without bearing but with pier cap (Bridge 3)	Pier	6888	34	19	99.51%	99.72%	99.24%	96.88%
	PierCap	4258	54	90	98.75%	97.93%	96.73%	
	Girder	34,833	681	35	98.08%	99.90%	97.99%	
	Deck	28,258	456	25	98.41%	99.91%	98.33%	
	Parapet	12,921	25	1081	99.81%	92.28%	92.11%	
With bearing and pier cap (Bridge 4)	Pier	3047	36	5	98.83%	99.84%	98.67%	96.79%
	PierCap	5925	598	55	90.83%	99.08%	90.07%	
	Girder	42,788	41	604	99.90%	98.61%	98.51%	
	Deck	25,480	166	79	99.35%	99.69%	99.05%	
	Parapet	11,398	83	189	99.28%	98.37%	97.67%	
Avg.	–	–	–	–	97.96%	98.65%	96.64%	–

D_S, respectively. The negative value of ΔmIoU indicates a reduction in segmentation accuracy from D_S to D_R, and a positive value signifies an improvement in accuracy during this process. It can be observed that the variation in segmentation accuracy is commonly lower than 1%, suggesting that the segmentation accuracy remains relatively stable and experiences only minor changes in this process.

5.5. Computational time

The proposed framework leveraged the RG algorithm to incrementally cluster the entire bridge point cloud into different bridge components. To assess its efficiency, the computational time of the proposed framework was calculated, as presented in Table 4. The total time required for segmenting entire bridge point clouds ranges from 651.5 s to 824.1 s. Specifically, the RG can segment each group of point clouds in Bridges 1 and 2 for approximately 10 s and within <20 s for Bridges 3 and 4. Notably, a longer duration is needed for Bridges 3 and 4 due to the extra segmentation of pier caps. The original point cloud segmentation time for the four bridges ranges from 22.0 s to 42.7 s. For data preprocessing, the down-sampling and slicing process can be efficiently completed within a few seconds. Nonetheless, it is noteworthy that the total time was dominated by feature generation, particularly the DK feature generation. Normal estimation of points requires between 34.8 s and 49.2 s, while DK feature generation demands a substantial time ranging from 492.4 s to 681.1 s.

Overall, the point cloud segmentation process can be efficiently finished since phases 2 and 3 are carried out on the point cloud after voxel and space down-sampling. Furthermore, the segmentation of the original point cloud with several millions of points in phase 4 can also be completed in around half a minute owing to the efficient neighbour-searching method Kd-tree. The main efficiency problem of this framework is feature generation. Specifically, over thirty seconds are needed

Table 3
Point-based evaluation for original point clouds (D_R).

Bridge type	Component	TP	FP	FN	Precision	Recall	IoU	mIoU
Without bearing and pier cap (Bridge 1)	Pier	334,346	11,305	144	96.73%	99.96%	96.69%	95.47%
	Girder	931,974	3472	12,946	99.63%	98.63%	98.27%	
	Deck	645,293	3249	19,179	99.50%	97.11%	96.64%	
	Parapet	250,131	20,339	6578	92.48%	97.44%	90.28%	
With bearing but without pier cap (Bridge 2)	Pier	209,732	4931	90	97.70%	99.96%	97.66%	97.10%
	Girder	1,098,868	5333	6054	99.52%	99.45%	98.97%	
	Deck	364,151	3387	2460	99.08%	99.33%	98.42%	
	Parapet	171,611	3614	8634	97.94%	95.21%	93.33%	
Without bearing but with pier cap (Bridge 3)	Pier	116,387	689	264	99.41%	99.77%	99.19%	96.94%
	PierCap	55,951	1160	1345	97.97%	97.65%	95.71%	
	Girder	446,633	6504	907	98.56%	99.80%	98.37%	
	Deck	424,793	7118	816	98.35%	99.81%	98.17%	
	Parapet	190,885	816	12,969	99.57%	93.64%	93.26%	
With bearing and pier cap (Bridge 4)	Pier	75,621	1300	191	98.31%	99.75%	98.07%	97.75%
	PierCap	134,010	4503	1565	96.75%	98.85%	95.67%	
	Girder	841,358	1319	5155	99.84%	99.39%	99.24%	
	Deck	422,032	2596	2145	99.39%	99.49%	98.89%	
	Parapet	204,239	2904	3679	98.60%	98.23%	96.88%	
Avg.	–	–	–	–	98.30%	98.53%	96.87%	–

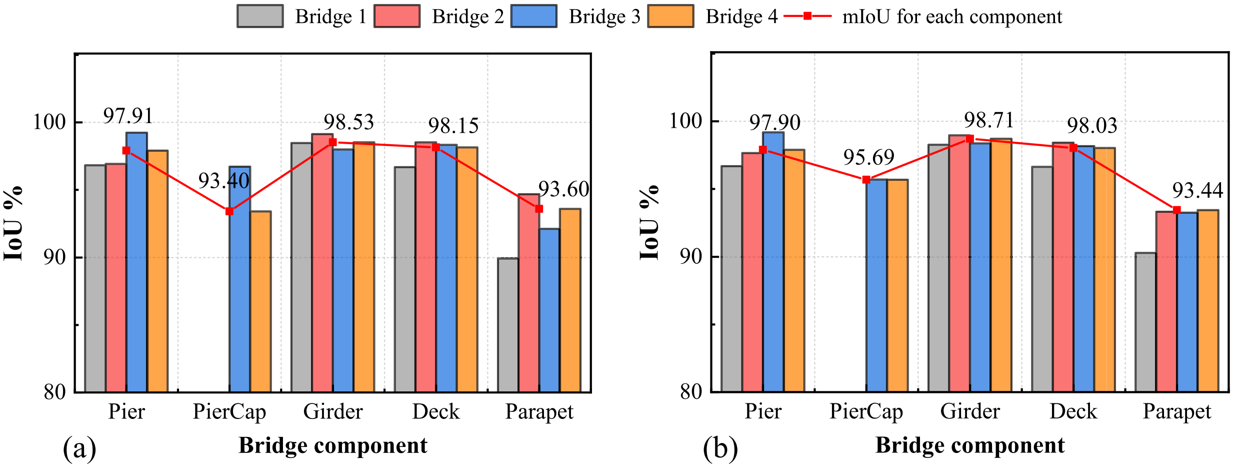

Fig. 16. mIoU for different bridge components in (a) D_S and (b) D_R.

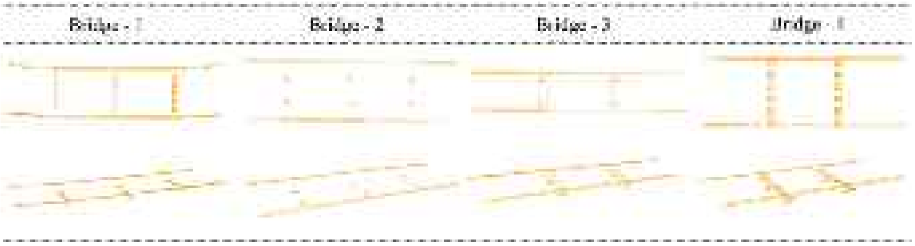

Fig. 17. Visualisation of the points with incorrect semantic labels.

for point normal estimation, including searching-radius determination and PCA-based normal calculation. Regarding DK feature generation, the long operation time is primarily spent on element generation. The node coordinate information can be inputted into FEA software in several seconds, but the input of element information will result in a long-time response from the software. This is because the element generation process often involves the automatic assignment of system-defined material information and other mechanical properties, which takes quite a lot of time for information generation and storage. There is a need to note that our method does not rely on complex and accurate FEA results and focuses on capturing structural response regularity. The software SAP2000 was adopted for the proof-of-concept. Therefore, the potential development of a tailored structural response analysis tool may significantly address the efficiency problem in DK feature generation.

6. Discussion

This paper introduces a DKRG framework for the semantic segmentation of bridge point clouds, and the experimental results revealed a superior performance in detecting key bridge components from point clouds. The effectiveness and robustness of the proposed framework were also compared with four state-of-the-art semantic segmentation models for bridge point clouds, as shown in Table 5. Although the

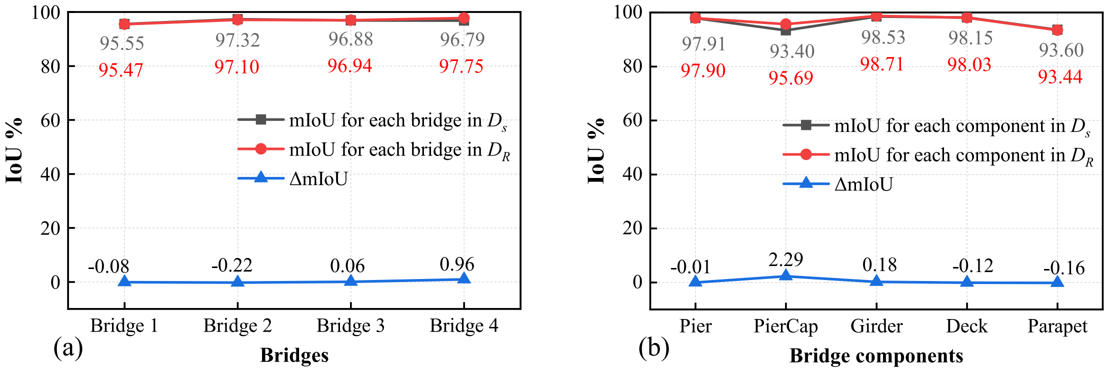

Fig. 18. Comparison of the loss of mIoU from D_S to D_R: (a) different bridges; (b) different bridge components.

Table 4
Computation time (seconds) in different steps of the proposed method.

Dataset	Point number	Data pre-processing			Segmentation			Original point cloud segmentation	Total
		Down-sampling and slicing	Normal estimation	DK feature generation	Group1	Group2	Group3		
Bridge 1	2,200,591	2.9	49.2	681.1	9.7	7.9	10.2	42.7	824.1
Bridge 2	1,861,600	17.7	45.4	492.4	8.1	6.0	8.6	28.4	651.5
Bridge 3	1,250,950	2.1	37.6	560.2	12.4	12.5	–	22.0	677.3
Bridge 4	1,689,995	3.4	34.8	578.2	17.5	18.8	–	27.5	722.5

Table 5
Comparison of state-of-the-art methods for bridge point cloud segmentation.

Reference	Methodology	Training data	Recognition Capability					Average mIoU
			Pier	Pier cap	Girder	deck	parapet	
[8]	Algorithm-based	No	Yes	Yes	Yes	No	No	96.867%
[18]	Algorithm-based	No	Yes	Yes	Yes	Yes	Yes	91.540%
[13]	Algorithm-and-learning-based	Yes	Yes	Yes	Yes	No	No	95.380%
[26]	Learning-based	Yes	Yes	Yes	Yes	Yes	Yes	96.490%
Our	Algorithm-based	No	Yes	Yes	Yes	Yes	Yes	96.873%

approaches selected for comparison adopted different datasets for testing, the mIoU is a relatively objective and well-accepted metric for evaluating their performance [46]. Due to various evaluation criteria, the mIoU may not be presented in some research. Therefore, we calculated the average mIoU for different bridge point clouds based on the point-wise evaluation results listed in these studies. It can be found that our method demonstrates remarkable precision in detecting various bridge components from point clouds. Notably, it surpasses the recognition capabilities of both the algorithm-based approach introduced by Lu et al. [8] and the algorithm-and-learning combined method proposed by Xia et al. [13]. Moreover, it exhibits superior segmentation accuracy compared to the algorithm-based method suggested by Truong-Hong et al. [18] and achieves competitive accuracy levels comparable to the learning-based method presented by Yang et al. [26].

To highlight its advantages, we will further distinguish our method from previous algorithm-based research in the following three aspects: (1) the overall segmentation strategy, (2) the usage of DK, and (3) the generalizability of the proposed method.

Firstly, the overall segmentation strategy outlines the order for identifying bridge components from point clouds. This segmentation order is usually determined through a systematic analysis of component topological constraints, contextual knowledge, and engineering criteria. For instance, Lu et al. [8] proposed a slicing method based on the observation of design rules and topological layouts of different RC bridge components. They divided the bridge point cloud into a set of assemblies, which can make the DK, such as design rules, more distinguishable in the assemblies. Then, the pier and pier cap are first detected from assemblies simultaneously, and the slab and girder are subsequently extracted from point clouds. Similarly, Truong-Hong et al. [18] proposed a strategy to detect component surfaces from superstructure to substructure by utilising spatial relationships and contextual knowledge associated with the surface of each component. However, the assumptions and defined rules in the previous method are sensitive to changes in girder geometry features since the girder is often the most complex and changeable component in RC bridges. In this study, we introduced a new segmentation strategy that starts with an “easy-to-difficult” approach. We initially identify the relatively simple substructures, such as piers and pier caps, which simplifies the subsequent segmentation of the superstructure, including girders and parapets. As a result, our segmentation strategy within this framework enhances robustness when dealing with more intricate superstructures, particularly those featuring complex girder sections.

Secondly, in previous research, the use of DK primarily revolves around the topologic relation of bridge components, engineering knowledge, and design rules. The topological relationships of bridge components, such as the direction of point normals [14] or density histograms from different views, have often been used as constraints for point cloud segmentation [8,15,30]. The design rules are also adopted as constraints to guide the segmentation process, such as removing upper slab surface points from pier areas [8] and extracting bridge component surfaces from point clouds [18]. Engineering knowledge is frequently leveraged for object classification in point clouds or to design the overall segmentation framework with a holistic perspective. As mentioned, the direct use of DK as segmentation rules for a small group of bridges, such as specific design parameters or density histograms, can restrict its applicability. In this study, we proposed a pioneering method that employs FEA to reveal and translate DK into accessible point features for algorithm processing. This approach significantly enhances the robustness of our approach to dealing with bridge components with different geometric features. Also, it sheds new light on using FEA-generated point features from structural knowledge.

Thirdly, concerning the generalizability of our proposed methods, the framework introduced in this study is specifically tailored for the point cloud segmentation of RC girder bridges with four different vertical layouts, which can effectively cover the majority of girder and slab-beam bridges in the real world [6,8]. The results reveal that the method can effectively identify piers, pier caps, girders, decks, and parapets for slab, slab-beam bridges, and single or multi-box girder bridges. Compared with previous DK-based methods [8,18,28,30], the proposed framework offers a wider application scope and displays better robustness for multi-type component segmentation from point clouds. Notably, this study provides a fundamental framework for RC bridge, and additional segmentation processes can be integrated seamlessly into this framework for more complex components. For instance, the lamp post can be further detected from the deck assembly, and the diaphragm can be further detected from the girder assembly. Additionally, the proposed strategy for using DK features also has the potential to deal with more complex bridge structures, such as girder bridges with more complex components and other bridge types, e.g., truss bridges and cable-stayed bridges.

7. Conclusions and future work

The scarcity of training data in learning-based methods has driven the exploration of DK-based segmentation methods for bridge point clouds. However, many existing DK-based methods are often tailored for specific bridges and struggle to adapt to different component types and engineering criteria. To address this challenge, this paper presents a DKRG framework for detecting RC bridge components from point clouds, enhancing the generalizability of the existing DK-based segmentation framework and RG-based approaches. To achieve this, an "easy-to-difficult" segmentation strategy was applied by considering the complexity of bridge components. For the method of using DK, we proposed a method of translating the bridge vertical layout characteristic into the accessible point-level feature through FEA for point cloud segmentation instead of directly interpreting DK as rules or constraints. Finally, a multi-feature-based RG segmentation framework that incorporates geometric features and DK-based point features was designed for the component-level segmentation of bridge point clouds.

The proposed method was tested on four bridge point clouds with different vertical layouts, including two real recorded bridge point clouds and two synthetic bridge point clouds generated by virtual laser scanning simulation. The results revealed that the proposed method could successfully detect the key components from bridge point clouds. For point-level performance evaluation, our method could achieve the lowest mIoU of 95.55% and 95.47% for the segmentation of data D_S and D_R, respectively. For different component types, the mIoU can reach an accuracy of approximately 98% for the segmentation of pier, girder, and deck components. For pier caps and parapets, the mIoU was around 94%. The results revealed high precision in detecting various bridge components from point clouds. The proposed segmentation strategy can successfully segment bridge point clouds from substructure to superstructure, which can be applied to a wider range of RC girder bridges and enables the segmentation of bridges with more complex superstructure components, such as multi-box girder bridges.

Nonetheless, the current framework still has some limitations. Firstly, although the proposed method avoids direct reliance on design parameters, some parameters still need to be set manually in our algorithm. Secondly, the absence of specialised structural feature calculation software hinders the direct processing of original point clouds. The current segmentation of original point clouds relies on the segmentation result of point clouds after down-sampling, which might pose a challenge for handling large-scale bridge point clouds. Thirdly, the significant occlusion of point clouds may cause the failure of FEA modelling and inaccurate results of FEA, and how to effectively extract DK-based point features when suffering serious point cloud quality problems still needs more research. Fourthly, for segmentation accuracy, it can be found that the incorrectly labelled points are primarily distributed along the component boundary, which can be considered as an edge-refinement problem of the RG algorithm and may need further investigation.

To address these challenges, future research will consider (1) refining and optimising the proposed framework (e.g., automating the process of parameter tuning, reducing the number of input points for handling large-scale point cloud segmentation and optimising the allocation of unclassified points to further improve the segmentation accuracy; proposing the feasible solution for dealing with the FEA modelling for incomplete point clouds). The potential idea here is to over-segment the raw point cloud, resulting in some point patches, and extract limited points from these patches to represent the raw point cloud for semantic segmentation. The limited points may provide a precise feature distribution and offer the potential to determine parameter thresholds automatically. For the point cloud occlusion problem, our segmentation framework will consider the necessary data pre-processing steps, such as training a bridge point cloud completion model and fusion of multi-point cloud sources. (2) further expanding the proposed framework to include the segmentation of more geometrically complex components of RC girder bridges, such as bearings and abutments; and (3) expanding the framework to deal with other types of bridges such as truss bridges and cable-stayed bridges.

CRediT authorship contribution statement

Tao Yang: Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Data curation, Conceptualization. **Yang Zou:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Xiaofei Yang:** Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Enrique del Rey Castillo:** Writing – review & editing, Writing – original draft, Supervision.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

The authors would like to acknowledge the financial support from the University of Auckland and China Scholarship Council (Project No. 202206690016).

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.

[2] A. Žnidarič, V. Pakrashi, E. O'Brien, A. O'Connor, A review of road structure data in six European countries, Proceed. Institut. Civil Eng. Urban Des. Plann. 164 (2011) 225–232, https://doi.org/10.1680/udap.900054.

[3] K.D. Flaig, R.J. Lark, The development of UK bridge management systems, Proc. Inst. Civ. Eng. Transp. 141 (2000) 99–106, https://doi.org/10.1680/tran.2000.141.2.99.

[4] Y. Fujino, D.M. Siringoringo, Structural health monitoring of bridges in Japan: an overview of the current trend, in: Proceedings of the 4th International Conference on FRP Composites in Civil Engineering 2008, CICE, 2008. ISBN 9783905594508.

[5] B. Fanning, C.M. Clevenger, M.E. Ozbek, H. Mahmoud, Implementing BIM on infrastructure: comparison of two bridge construction projects, Pract. Period. Struct. Des. Constr. 20 (2015) 1–8, https://doi.org/10.1061/(ASCE)SC.1943-5576.0000239.

[6] R. Sacks, A. Kedar, A. Borrmann, L. Ma, I. Brilakis, P. Hüthwohl, S. Daum, U. Kattel, R. Yosef, T. Liebich, B.E. Barutcu, S. Muhic, SeeBridge as next generation bridge inspection: overview, information delivery manual and model view definition, Autom. Constr. 90 (2018) 134–145, https://doi.org/10.1016/j.autcon.2018.02.033.

[7] P. Tang, D. Huber, B. Akinci, R. Lipman, A. Lytle, Automatic reconstruction of as-built building information models from laser-scanned point clouds: a review of related techniques, Autom. Constr. 19 (2010) 829–843, https://doi.org/10.1016/j.autcon.2010.06.007.

[8] R. Lu, I. Brilakis, C.R. Middleton, Detection of structural components in point clouds of existing RC bridges, Comput. Aided Civ. Inf. Eng. 34 (2019) 191–212, https://doi.org/10.1111/mice.12407.

[9] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: deep learning on point sets For 3D classification and segmentation, in: 2017 IEEE Conference on computer vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 77–85, https://doi.org/10.1109/CVPR.2017.16.

[10] H. Kim, J. Yoon, S.H. Sim, Automated bridge component recognition from point clouds using deep learning, Struct. Control. Health Monit. 27 (2020), https://doi.org/10.1002/stc.2591.

[11] X. Yang, E. del Rey Castillo, Y. Zou, L. Wotherspoon, Semantic segmentation of bridge point clouds with a synthetic data augmentation strategy and graph-structured deep metric learning, Autom. Constr. 150 (2023) 104838, https://doi.org/10.1016/j.autcon.2023.104838.

[12] Y. Jing, B. Sheil, S. Acikgoz, Segmentation of large-scale masonry arch bridge point clouds with a synthetic simulator and the BridgeNet neural network, Autom. Constr. 142 (2022) 104459, https://doi.org/10.1016/j.autcon.2022.104459.

[13] T. Xia, J. Yang, L. Chen, Automated semantic segmentation of bridge point cloud based on local descriptor and machine learning, Autom. Constr. 133 (2022) 103992, https://doi.org/10.1016/j.autcon.2021.103992.

[14] B. Riveiro, M.J. DeJong, B. Conde, Automated processing of large point clouds for structural health monitoring of masonry arch bridges, Autom. Constr. 72 (2016) 258–268, https://doi.org/10.1016/j.autcon.2016.02.009.

[15] D. Lamas, A. Justo, M. Soilán, M. Cabaleiro, B. Riveiro, Instance and semantic segmentation of point clouds of large metallic truss bridges, Autom. Constr. 151 (2023) 104865, https://doi.org/10.1016/j.autcon.2023.104865.

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

[17] S.B. Walsh, D.J. Borello, B. Guldur, J.F. Hajjar, Data processing of point clouds for object detection for structural engineering applications, Comput. Aided Civ. Inf. Eng. 28 (2013) 495–508, https://doi.org/10.1111/mice.12016.

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

[19] H. Kim, C. Kim, Deep-learning-based classification of point clouds for bridge inspection, Remote Sens. 12 (2020) 1–13, https://doi.org/10.3390/rs12223757.

[20] Y. Li, R. Bu, M. Sun, W. Wei, X. Di, C. Baoquan, PointCNN : convolution on X -transformed points, in: Advances in Neural Information Processing Systems, 2018, pp. 820–830 (ISSN 10495258).

[21] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph cnn for learning on point clouds, in: ACM Transactions on Graphics, 2019, pp. 1–12, https://doi.org/10.1145/3326362.

[22] J.S. Lee, J. Park, Y.M. Ryu, Semantic segmentation of bridge components based on hierarchical point cloud model, Autom. Constr. 130 (2021), https://doi.org/10.1016/j.autcon.2021.103847.

[23] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, Randla-net: efficient semantic segmentation of large-scale point clouds, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020, pp. 11105–11114, https://doi.org/10.1109/CVPR42600.2020.01112.

[24] L. Landrieu, M. Simonovsky, Large-scale point cloud semantic segmentation with superpoint graphs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4558–4567, https://doi.org/10.1134/1.559035.

[25] Y. Lin, A. Habib, Semantic segmentation of bridge components and road infrastructure from mobile LiDAR data, ISPRS Open J. Photogram. Rem. Sens. 6 (2022) 100023, https://doi.org/10.1016/j.ophoto.2022.100023.

[26] X. Yang, E. del Rey Castillo, Y. Zou, L. Wotherspoon, Y. Tan, Automated semantic segmentation of bridge components from large-scale point clouds using a weighted superpoint graph, Autom. Constr. 142 (2022) 104519, https://doi.org/10.1016/j.autcon.2022.104519.

[27] J.W. Ma, T. Czerniawski, F. Leite, Semantic segmentation of point clouds of building interiors with deep learning: augmenting training datasets with synthetic BIM-based point clouds, Autom. Constr. 113 (2020), https://doi.org/10.1016/j.autcon.2020.103144.

[28] L. Truong-Hong, R. Lindenbergh, Extracting bridge components from a laser scanning point cloud, in: Lecture Notes in Civil Engineering 98, 2021, pp. 721–739, https://doi.org/10.1007/978-3-030-51295-8_50.

[29] L. Ma, R. Sacks, U. Kattel, T. Bloch, 3D object classification using geometric features and pairwise relationships, Comput. Aided Civ. Inf. Eng. 33 (2018) 152–164, https://doi.org/10.1111/mice.12336.

[30] Y.-P. Zhao, H. Wu, P.A. Vela, Top-down partitioning of reinforced concrete bridge components, in: Computing in Civil Engineering 2019, American Society of Civil Engineers, Reston, VA, 2019, pp. 275–283, https://doi.org/10.1061/9780784482445.035.

[31] P.J. Besl, R.C. Jain, Segmentation through variable-order surface fitting, IEEE Trans. Pattern Anal. Mach. Intell. 10 (1988) 167–192, https://doi.org/10.1109/34.3881.

[32] F. Poux, C. Mattes, Z. Selman, L. Kobbelt, Automatic region-growing system for the segmentation of large point clouds, Autom. Constr. 138 (2022) 104250, https://doi.org/10.1016/j.autcon.2022.104250.

[33] Y. Xie, J. Tian, X.X. Zhu, Linking points with labels in 3D: a review of point cloud semantic segmentation, IEEE Geosci. Rem. Sens. Magaz. 8 (2020) 38–59, https://doi.org/10.1109/MGRS.2019.2937630.

[34] A.V. Vo, L. Truong-Hong, D.F. Laefer, M. Bertolotto, Octree-based region growing for point cloud segmentation, ISPRS J. Photogramm. Remote Sens. 104 (2015) 88–100, https://doi.org/10.1016/j.isprsjprs.2015.01.011.

[35] A. Dimitrov, M. Golparvar-Fard, Segmentation of building point cloud models including detailed architectural/structural features and MEP systems, Autom. Constr. 51 (2015) 32–45, https://doi.org/10.1016/j.autcon.2014.12.015.

[36] T. Rabbani, F.A. Van Den Heuvel, G. Vosselman, Segmentation of point clouds using smoothness constraints, in: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2006, pp. 248–253 (ISSN 16821750).

[37] L. Truong-hong, S. Chen, V.L. Cao, D.F. Laefer, Automatic bridge deck damage using low cost UAV-based images, in: TU1406 Quality Specifications for Roadway Bridges Standardization at a European Level, 2018, pp. 1–6. http://hdl.handle.net/2451/43479.

[38] M.K. Kim, S. McGovern, M. Belsky, C. Middleton, I. Brilakis, A suitability analysis of precast components for standardized bridge construction in the United Kingdom, Procedia Eng. 164 (2016) 188–195, https://doi.org/10.1016/j.proeng.2016.11.609.

[39] D.N. Farhey, Structural performances of bridge types in the U.S. National Bridge Inventory, Infrastructures 3 (2018), https://doi.org/10.3390/infrastructures3010006.

[40] X. Ester, M. Kriegel, H.P. Sander, J. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, In Kdd 96 (1996) 226–2331, https://doi.org/10.1016/B978-0-444-64165-6.03005-6.

[41] M.A. Fischler, R.C. Bolles, Random sample consensus, Commun. ACM 24 (1981) 381–395, https://doi.org/10.1145/358669.358692.

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

[43] M. Saeed Mafipour, C. Alici, S.S. Shakeel, A. Kalkavan, Semantic Segmentation of Real and Synthetic Point Cloud Data for Digital Twinning of Bridges, in: Proceedings of 33, Forum Bauinformatik, 2022. https://mediatum.ub.tum.de/doc/1688410/.

[44] L. Winiwarter, A.M. Esmorís Pena, H. Weiser, K. Anders, J. Martínez Sánchez, M. Searle, B. Höfle, Virtual laser scanning with HELIOS++: a novel take on ray tracing-based simulation of topographic full-waveform 3D laser scanning, Remote Sens. Environ. 269 (2022), https://doi.org/10.1016/j.rse.2021.112772.

[45] S. Bechtold, B. Höfle, HELIOS: A Multi-purpose Lidar Simulation Framework for Research, Planning and Training of Laser Scanning Operations with Airborne, Ground-based Mobile and Stationary Platforms, ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences III–3, 2016, pp. 161–168, https://doi.org/10.5194/isprsannals-iii-3-161-2016.

[46] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, M. Bennamoun, Deep learning for 3D point clouds: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 4338–4364, https://doi.org/10.1109/TPAMI.2020.3005434.