



Two-stage bridge point cloud segmentation by fusing deep learning and heuristic methods

Tian Zhang^{a,*}, Haonan Chen^a, Pengfei Li^b, Haijiang Li^c

^a Transportation Engineering College, Dalian Maritime University, Dalian 116026, China

^b Research Institute of Highway Ministry of Transport, Beijing 100088, China

^c Cardiff School of Engineering, Cardiff University, Cardiff CF24 3AA, UK

ARTICLE INFO

Keywords:

Point clouds
Deep learning
Heuristic method
Bridge component recognition
Secondary segmentation

ABSTRACT

The point cloud acquired from the data acquisition equipment is not segmented by components, and reverse engineering without component segmentation has limited value. Existing heuristic methods achieve high segmentation accuracy but are computationally intensive. On the other hand, deep learning models are quick but often lack accuracy and depend on limited datasets. To address these issues, this paper introduces a fusion method that utilizes results trained on an easily created synthetic dataset to initially segment the point cloud roughly, aiming for a segmentation accuracy and intersection over union ratio of 80 % and 70 %, respectively. Subsequently, a streamlined heuristic method is applied to comprehensively segment the point cloud. The verification results of the instance indicate that this approach achieves the same high level of accuracy ($\geq 99\%$) as heuristic methods but increases the speed of segmentation by approximately 2.52 times. The method involves using a synthetic dataset, derived from real point cloud data, in conjunction with the fusion method and selecting a segmentation network that is optimized for simple synthetic datasets.

1. Introduction

Transportation infrastructure serves as a fundamental framework ensuring the seamless movement of people and goods, with bridges playing a pivotal role in this system. Over recent years, driven by advancements in infrastructure, there has been a significant increase in bridge construction. Specifically, between the beginning of 2019 and the end of 2022, 181,700 new road bridges were constructed, bringing the total to 1,033,200 by the end of 2022 [1,2]. This expansion underscores the necessity not only for ongoing health monitoring of these bridges but also for comprehensive management strategies and the development of maintenance protocols. Consequently, enhancing the management and maintenance of road bridges has been identified as a critical developmental objective [3]. This includes the imperative to improve the efficiency and effectiveness of maintenance funding, strengthen the management of maintenance projects, and advance the digital transformation of bridge maintenance processes.

In light of this context, the digital transformation of transportation technologies has become a central focus of research [4]. A key challenge in this area is the creation of a digital model that accurately reflects the actual state of a bridge [5]. Point cloud technology, which preserves

detailed three-dimensional (xyz) and multi-dimensional (rgb, intensity) data, has emerged as a vital tool for inverse modeling. However, initial point cloud data collected from bridges often lacks clear demarcations between different structural components, rendering it ineffective for component-specific modeling without further processing. The process of three-dimensional reconstruction of bridge structures from point clouds typically follows one of two methodologies [6]: 1) segmenting the point cloud model prior to component identification, or 2) labeling all point clouds before clustering them. Once the point cloud is segmented by classification, the resultant spatial information becomes useful for reconstruction purposes. Clearly, semantic segmentation of the point cloud is essential, as the accuracy of segmentation not only impacts the overall quality of the 3D reconstruction but also influences model parameters and the geometric fidelity of the bridge model. Addressing the challenge of accurately segmenting bridge components thus becomes crucial for enhancing the precision of bridge modeling using point clouds. The primary aim of this paper is to segment the bridge point cloud effectively to meet the technical specifications required for subsequent reconstruction tasks.

* Corresponding author.

E-mail address: saghb@126.com (T. Zhang).

<https://doi.org/10.1016/j.measurement.2025.117125>

Received 16 June 2024; Received in revised form 13 January 2025; Accepted 24 February 2025

Available online 26 February 2025

0263-2241/© 2025 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

2. Related work

In this section, we categorize the point cloud segmentation methods applied to bridges into two main types: heuristic methods and deep learning methods, with the latter predominantly relying on synthetic datasets.

2.1. Traditional segmentation method

In the segmentation of 3D point clouds, traditional methods are typically bifurcated into top-down and bottom-up approaches. This categorization is inspired by the dual methodologies employed in the instance segmentation of images. (1) Top-down segmentation initially involves object recognition to identify all instances within an image, subsequently conducting semantic segmentation using these instances as the fundamental units. (2) Bottom-up segmentation assigns a label to each pixel of the image. Utilizing these labels, along with an appropriate clustering algorithm, yields instance segmentation results. In this analogy, the entire point cloud is comparable to an image, and individual points in the point cloud are analogous to pixels. The bottom-up method encounters significant obstacles in the segmentation of bridge point clouds, specifics of which are not discussed here. However, top-down approaches, often heuristic in nature [6], segment components by progressively leveraging information on the bridge's geometric structure and topological constraints. This reliance on pre-existing knowledge necessitates tailored approaches for different bridges and involves considerable iterative costs. Recent developments in heuristic segmentation methods are presented below.

For girder road bridges, Lu et al. [6] exemplify heuristic segmentation using a bottom-up method. By analyzing point clouds from ten real bridges, including data on normal vectors, boundaries, and point cloud density, they identified the positions and boundaries of bridge components. The segmentation accuracy achieved is suitable for producing the corresponding geometric digital twins [7]. Lamas et al. [8] examined two real truss bridges using a terrestrial laser scanner (TLS), applied voxel downsampling for uniformity during preprocessing, and autonomously segmented different rod elements to construct a geometric model inclusive of node information. This paper represents the inaugural exploration into the automated segmentation of truss bridge point clouds. Despite enhancing both universality and accuracy, the method necessitates manually measured variables. For arch road bridges, Sánchez-Rodríguez et al. [9] employed azimuth angles to differentiate the point cloud into vertical and non-vertical walls, and subsequently executed fine segmentation on the bridge piers. In the context of girder railway bridges, Soilán et al. [10] automatically differentiated the tracks from collected railway bridge point clouds and standardized the output in BIM format as per the IFC standard, incorporating track details and central routes.

These heuristic methods have consistently achieved exemplary segmentation results, with accuracy generally surpassing 99 % or less than one error per 10,000 points. This precision is a significant advantage of heuristic methods, which do not demand extensive point cloud data as deep learning approaches do. However, these methods often introduce numerous variables that require manual input, complicating the transferability and scalability of the approaches and the learning of the segmentation mechanisms.

2.2. Synthetic data and deep learning methods

It is important to note that deep learning methods utilized for point cloud segmentation can also be categorized into bottom-up or top-down approaches based on their segmentation mechanisms [7]. However, due to their inherently opaque nature, these methods are often treated as a distinct category. The versatility of the same deep learning technique across various bridge types is advantageous, yet it is accompanied by challenges related to dataset adequacy. Although dataset dependency is

a pervasive issue in deep learning, the scarcity of comprehensive point cloud datasets is particularly acute in the domain of bridge analysis. Some researchers attempt to mitigate this issue through cross-validation using limited datasets [8], but this is not a sustainable long-term strategy. Collecting point cloud data is notably more challenging than gathering image datasets due to more complex environmental conditions. Consequently, the creation of synthetic point cloud data has emerged as a popular solution among scholars. For instance, Zhou et al. [9] employed the public sewer defect point cloud dataset [10] as a training set for detecting sewer defects. However, for bridges, a universally accepted public synthetic point cloud dataset does not yet exist. Below are some of the latest or notable examples of using the “synthetic training set + deep learning” approach for bridges.

For girder road bridges, Lamas et al. [11] created a synthetic point cloud of a truss bridge with high variability by specifying the size and components of the bridge. They synthesized two sets of truss bridge point clouds, one without occlusion and one with simulated occlusion, and adapted the JSNet architecture [12] for segmentation training. Based on the real truss bridge point cloud and segmentation results obtained by the team in heuristic work [13], the real bridge point cloud was used as the test set. The comparison of segmentation effects between the two methods concluded that the proposed method was superior to the heuristic approach. This paper also verified the usability of the point cloud synthesis method. Yang et al. [14] first introduced a weighted superpoint graph (WSPG) method, which is an adaptation of the deep learning SPG method [15]. This modification was necessary due to the asymmetry in point cloud proportions among bridge components, which often complicates the identification of individual components during semantic segmentation. WSPG specifically modifies the loss function in SPG, and subsequent work validated the excellent performance of WSPG in tasks involving bridge components [16]. Both experiments utilized synthetic point clouds, starting with a conversion from Revit model to IFC format, then to a point cloud model, and finally to an annotated dataset. Building on the findings in [17], two strategies were proposed [16] to enhance segmentation effectiveness: 1) a synthetic point cloud augmentation strategy that combines synthetic and real point clouds as the training set, and 2) a synthetic superpoint augmentation strategy, which generates superpoints from both real and synthetic point clouds and integrates them with the original point cloud. While these approaches successfully addressed the issue of insufficient point cloud data and yielded positive outcomes, only 22 and 10 synthetic point cloud models with detailed modeling transformations were employed in the respective studies. For arch road bridges, based on the systems of RandLA-Net[17] and FG-Net[18], Jing et al. [19] modified and proposed a deep learning network for point clouds of large-scale masonry arch bridges, called BridgeNet. The network model retains the applicability of the original network to large-scale models, and modifies the local feature aggregation module to learn more complex local features and mitigate the overfitting problem. For the training data, the authors developed a point cloud generator for this kind of bridge, and produced 2000 bridge point clouds as the training set for BridgeNet. A large number of generated point clouds well meet the needs of training data, but the segmentation accuracy will be affected by the sacrifice of part of the geometry of the masonry arch bridge. For girder railway bridges, there is currently no established method for producing synthetic datasets. On one hand, the repetitive nature of railway segments allows for the generation of numerous datasets for training through multiple segmentations [20]. On the other hand, the primary research focus is on segmenting complex cables, tracks, and environments [21], and this area has not been extensively developed. Historically, heuristic algorithms were predominantly used for point cloud segmentation in railway settings [22], but the issue of data insufficiency is increasingly evident. Qiu et al. [23] also recognized this challenge in their work using an improved DGCNN to address complex railway scenes.

The mean accuracy (mAcc) of the studies mentioned earlier ranges from 87 % to 99 %, while the mean intersection over union (mIoU) spans

from 78 % to 98 %. These metrics exhibit significant variance across different datasets, with the highest values contingent on the specific dataset employed. These studies commonly initiate with the construction of a BIM model, which is subsequently used to generate the point cloud dataset. Two primary issues emerge from these works: 1) Batch modeling compromises the geometrical fidelity of the bridge, whereas fine modeling inhibits large-scale dataset production, leading to a scarcity of datasets and insufficiently detailed datasets, which adversely affect segmentation outcomes. 2) It is challenging to amass a sufficient quantity of real datasets for training; moreover, reliance solely on synthetic datasets for training yields limited segmentation effectiveness and lacks robustness, rendering them unsuitable for direct application in 3D reconstruction tasks.

3. Methodology

This paper addresses the limitations discussed in Sections 2.1 and 2.2 regarding the restricted segmentation accuracy of deep learning when utilizing synthetic data as the training set. It is proposed that the results of machine segmentation undergo a secondary segmentation process. We explore the application of an existing heuristic method as this secondary segmentation approach because it not only leverages the high segmentation accuracy of heuristic methods but also enhances the controllability of the segmentation results. During our research, it was observed that this sequential segmentation strategy not only simplifies the current heuristic processes but also diminishes the demands on the outcomes of deep learning and, consequently, the training dataset. This approach allows for the reduction in dataset precision and facilitates batch production, thereby addressing the first problem identified previously. Consequently, the objective of this paper is to synergize the dataset preparation, deep learning method segmentation, and heuristic method segmentation to enhance segmentation accuracy while simplifying the processes, thereby creating an integrated system that links both types of segmentation methods as illustrated in Fig. 1.

In summary, the point cloud is initially segmented coarsely using the results trained on the synthetic dataset, where the segmentation accuracy and intersection over union ratio are only required to reach 80 % and 70 %, respectively. The refined heuristic is then employed to achieve comprehensive segmentation.

This section proposes a methodology for constructing such a system, detailing how to select the data structure and segmentation method, how to build a synthetic dataset, how to achieve better results in the pre-segmentation process, and how to utilize the pre-segmentation results for secondary segmentation.

3.1. Synthetic point cloud dataset

The purpose of this section is to build the Synthetic point cloud dataset required for point cloud component segmentation. The construction method should meet two aspects: 1) It is easy and fast to build datasets that meet the requirements in batch. 2) It is closer to the real bridge point cloud, for example, the pier column may have different lengths due to occlusion, there is no point cloud between components, the left and right sections of the main beam can not obtain point cloud, the orientation is random, there is slope, etc. In short, the core purpose is to make the results of deep learning training meet the input requirements of the second segmentation while batch and fast. As shown in Fig. 2, this process can be divided into three parts, batch modeling, classifying the bridge component point cloud, and adjusting the data.

3.1.1. Batch modeling

The purpose of batch modeling is to realize the batch construction of bridge solid models for conversion to point cloud models, and realize the size random containing topological relationships.

The bridge structure data is constructed by Excel, not only because it can interact with Dynamo directly, but also because the process of data input is the process of realizing the first data random and mutual constraint between data. The bridge data with the same components are saved as the same set of data to meet the data structure in 3.1. The size range of the topological relationship of each component is drawn up through the specification [24–26]. For example, the height of the bridge pier can be randomly sampled from greater than 3.5 m according to the clear height of the car passage, but it will also be affected by the height of the pier cap. The position of piers will be affected by the length, width and number of piers of the bridge. Topological relations and randomness which are difficult to construct in Excel can be further realized in Dynamo, and the cumbersome functions in Dynamo can also be realized by adjusting the structure of Excel. Each row in each group of data represents a bridge, and the bridges in the same group of data are placed in the same Revit file, which is a batch of solid model data.

3.1.2. Classify the point clouds of bridge components

The purpose of classifying the point cloud of bridge components is to transform the solid model into the point cloud model and solve the problem of batch assignment to the target.

The basic processing of the point cloud in this paper will be with the help of the point cloud processing software CloudCompare (CC), and the dataset in Fig. 2 can be read by CC by converting.rvt to.fbx without the help of plugins. The experiment shows that this operation will result in the overall scale scaling, but the scale of different batches of models is almost the same. The error is acceptable for data constructed from random data. Since there are differences between the volumes, point

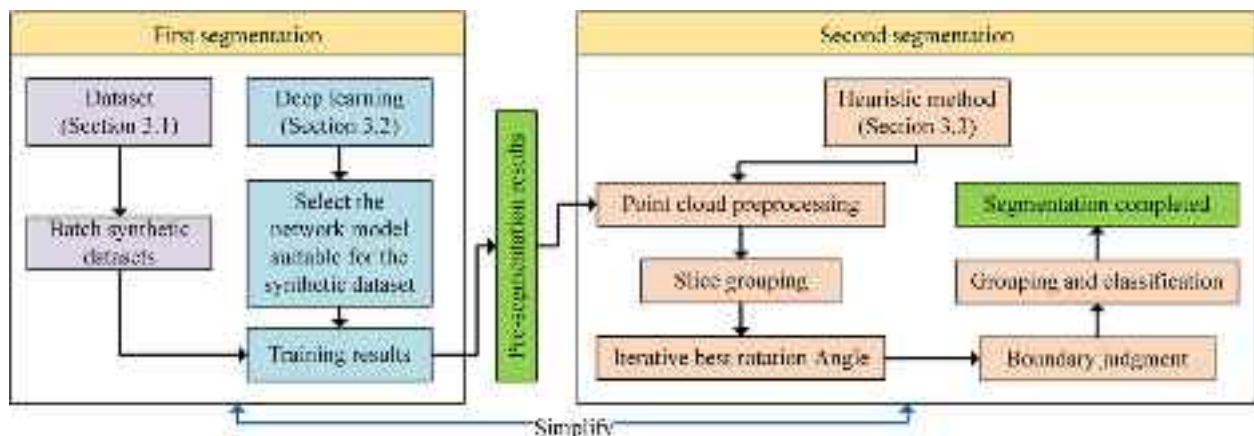


Fig. 1. Two-stage bridge segmentation process.

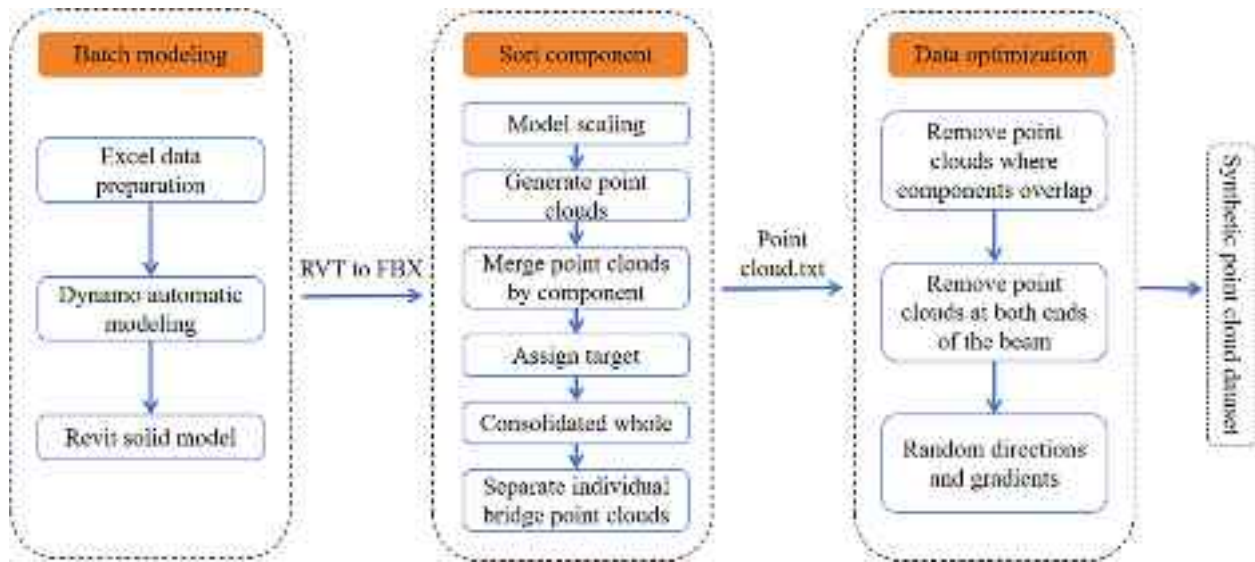


Fig. 2. Process for batch synthesis of point cloud datasets.

clouds are generated according to the density of the mesh model.

FBX files are read in groups of components, making it easier to assign targets, which is one of the reasons why we will convert to.fbx. When assigning target, it is difficult to assign target by directly merging point

clouds in units of bridges. Therefore, we can follow the process of merging point clouds with components, giving scalar fields, overall merging and manually dividing bridge point clouds, and finally output txt for subsequent modification.

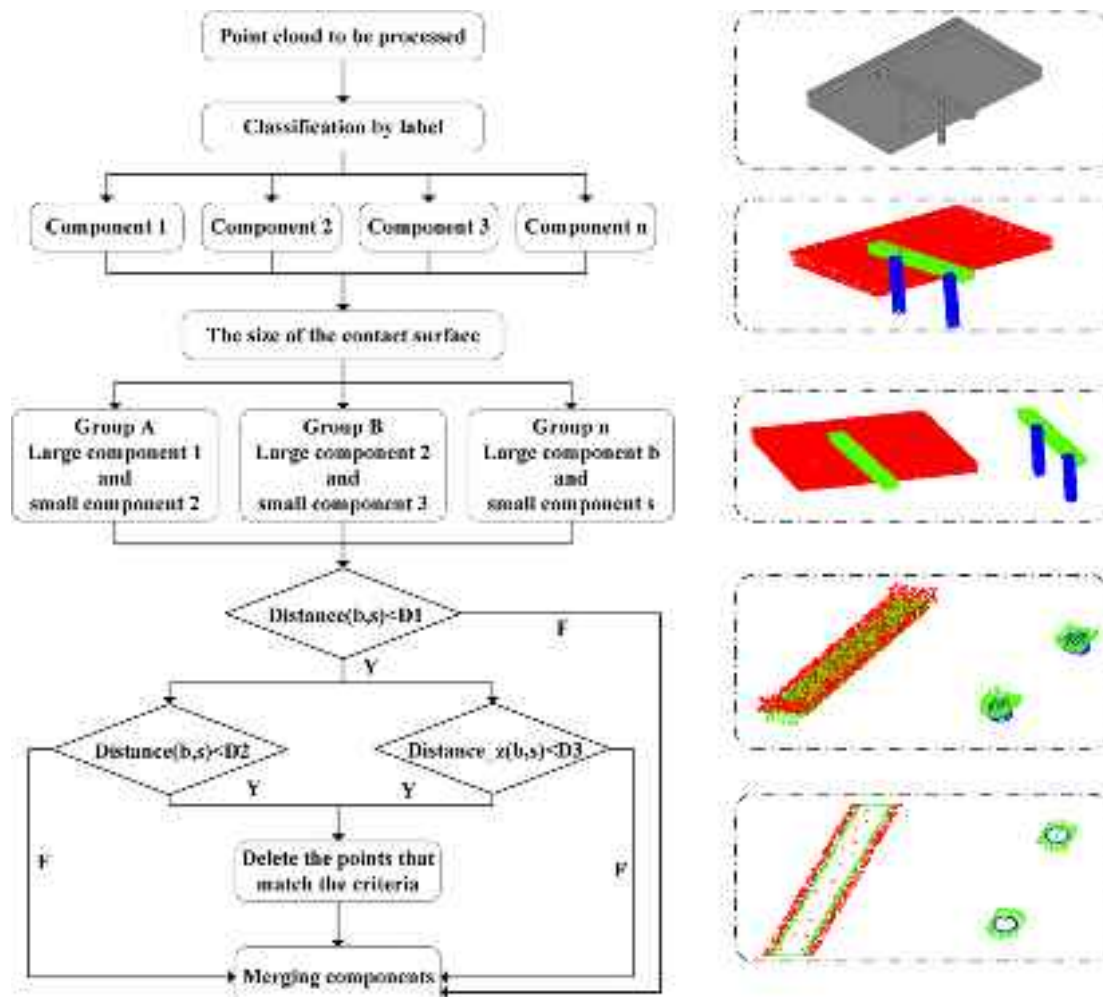


Fig. 3. Remove the coincidence points of synthetic point clouds.

3.1.3. Adjusting the data

Since the point cloud is generated in units of components, according to the above process, there will be point clouds in the contact surface of components. This difference will have a bad impact in the process of deep learning. For example, many Encoder modules of the current point cloud classification network rely on the K-Nearest Neighbor (KNN) algorithm. For some components, these overlapping points affect the proximity values of the sample points in KNN to a large extent, resulting in that the features of Synthetic point clouds cannot be generalized to real point clouds. It is difficult to remove the point clouds at the overlap, so the principle of deleting most point clouds with large components and all point clouds with small components at the contact surface will be used to remove the overlapping point clouds according to the process in Fig. 3. For the case of no point cloud in the left and right sections of the main beam, it is only necessary to delete with the help of obtaining the maximum value at both ends.

Since the point cloud text information input to the segmentation network itself is not a real spatial structure, the text information will also change greatly when the point cloud is rotated by a small Angle in space. In the experiment process of 3.3, it is also proved that the segmentation effect will change significantly when the point cloud of the same bridge is rotated by different angles. Therefore, all Synthetic point clouds were randomly rotated around the z-axis with a normal distribution of slope between -4% and 4% [24]. Although the point cloud is also randomly located from the center point, it is common to have a centralized operation during segmentation training, so this step can be omitted when creating the dataset.

3.2. Deep-learning-based pre-segmentation

The aim of this section is to utilize the synthetic point cloud dataset from section 3.2 as the training set to evaluate segmentation networks that are apt for segmenting components within synthetic datasets. This involves assessing the viability of various classification approaches for segmenting synthetic bridge datasets and comparing segmentation performance in terms of data quality and normals. Initially, the segmentation of real point cloud components is achieved, and by increasing the down-sampling points, higher accuracy and IoU are obtained to enhance the subsequent segmentation.

Drawing upon the classification methods outlined in reviews [27–29], representative networks were selected based on their foundational approach to handling original point clouds. These include PointNet++ [30], which builds upon the original PointNet, DGCNN [31], known for its improved performance in graph convolutional neural networks, and PCT [32], notable for its fewer parameters and enhanced performance in classification networks utilizing an attention mechanism. Each of these networks exemplifies a different classification ideology. However, it is important to note that no single network fully embodies a classification ideology, and their effectiveness must be empirically tested.

During dataset creation, a high degree of randomness is preferred because it suggests that synthetic point clouds closely mimic the general characteristics of real point clouds, emphasizing the importance of randomness in this process. Conversely, when synthetic datasets serve as training data, the process requires a more deterministic approach to optimize the dataset. For instance, in computer vision for damage recognition, the images comprising the dataset are typically random; however, during the research process, data are often intentionally repeated to highlight features and improve recognition outcomes [33]. For the research discussed in this paper, the training dataset aims to assemble a collection of bridge components, such as box girders or pier caps, that share the same name but exhibit varying geometric characteristics. For example, box girders could differ significantly in the length of their cantilever sections, which profoundly impacts the geometry of the cross-section. Similarly, a pier cap might be rectangular or resemble an inverted trapezoid. These variations suggest that certain

combinations might yield optimal segmentation results under the same classification network. Hence, the goal of the classification experiments is to make the virtual dataset more effective and achieve the best possible segmentation outcomes. Therefore, when using synthetic datasets as training data, the objective is to discover the most effective combinations, rather than relying on random combinations.

As a significant piece of geometric information in bridge point cloud analysis, normals play a crucial role in classification tasks. Unlike RGB data, which can be easily distorted during the point cloud collection process, normals are not only resistant to such interference, but can also be generated at a later stage. Generally, incorporating normals as training data improves segmentation results. However, our experiments indicate that different point cloud classification networks vary in their responsiveness to normals—some may not effectively utilize this information at all. This variation in effectiveness primarily relates to the underlying mechanisms of the network model and is not indicative of the network's overall performance.

Due to significant differences in surface area among bridge components, there is also a notable variance in the proportion of point clouds. For example, in bridge1 (Fig. 11), the ratio of pier to pier cap to beam is approximately 1:1.1:10. This imbalance is addressed by increasing the down-sampling points.

The efforts detailed in this section aim to achieve a minimum mAcc and mIoU of 80 % and 70 %, respectively. Achieving these metrics not only validates the effectiveness of the dataset creation method described in Section 3.2 but also demonstrates that a simplified, purely synthetic dataset can sufficiently meet the input requirements for subsequent segmentation tasks. If mAcc and mIoU fail to reach the minimum, the results of Boundary judgment and Grouping and classification in Section 3.3 will be deviated, thereby affecting the final segmentation results.

3.3. Secondary segmentation method

Modify the heuristic to obtain the same accuracy level as the heuristic (the heuristic in Section 2.1 is for unsegmented point clouds) Based on the classification ability of the current point cloud classification technology, under the condition of pure synthetic data set as the training set, the effect of deep learning on point cloud segmentation is eventually limited. At present, it is difficult to achieve application-level component segmentation only with deep learning, so it is imperative to perform secondary segmentation on the already segmented point cloud. The first segmentation is to classify most of the point clouds correctly, and the classification results often have two problems: 1) mixed wrong points in each component, and 2) the boundary of each component is not clear. These two problems will seriously affect the subsequent work based on the classified point cloud, and the second segmentation is to solve these two problems.

The second segmentation takes the beam bridge as an example. Different from the traditional heuristic method, the pre-segmented point cloud does not need complex segmentation methods and segmentation steps, but only needs to solve the above two problems and make the final classification results controllable. For girder bridges, in addition to special components such as diaphragms, among the components that can be obtained by point cloud scanning tools, there is generally an obvious hierarchical relationship between each component of the bridge in the vertical or horizontal direction. With the help of this characteristic, the secondary classification is realized by combining the incompletely classified point clouds. The classification method is as follows:

- (1) Point cloud preprocessing: As shown in Fig. 4, Principal Component Analysis (PCA) is used to make the main body of the bridge basically parallel to the X-axis, and centralization can be realized.
- (2) Slice grouping: cut the bridge point cloud into N slices $S = \{S_i : i = 1, 2, \dots, N\}$ along the X-axis with the same thickness (Fig. 5

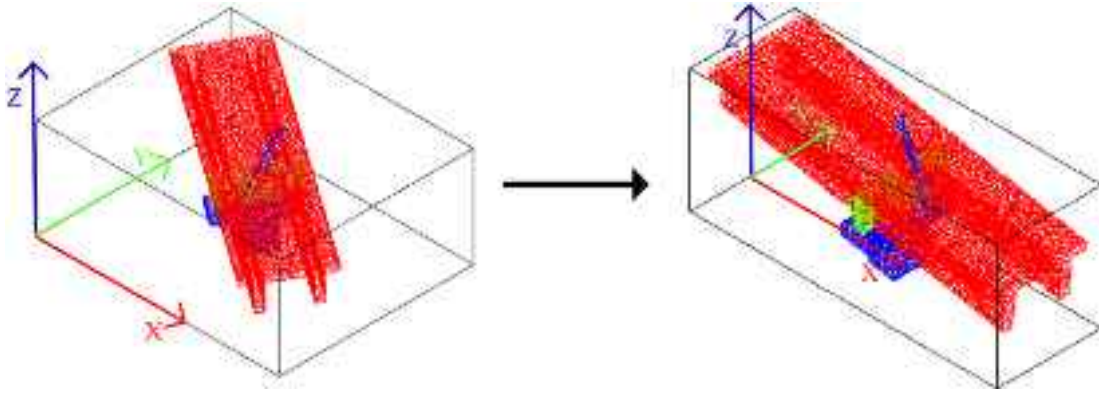


Fig. 4. The point cloud position is adjusted using PCA.

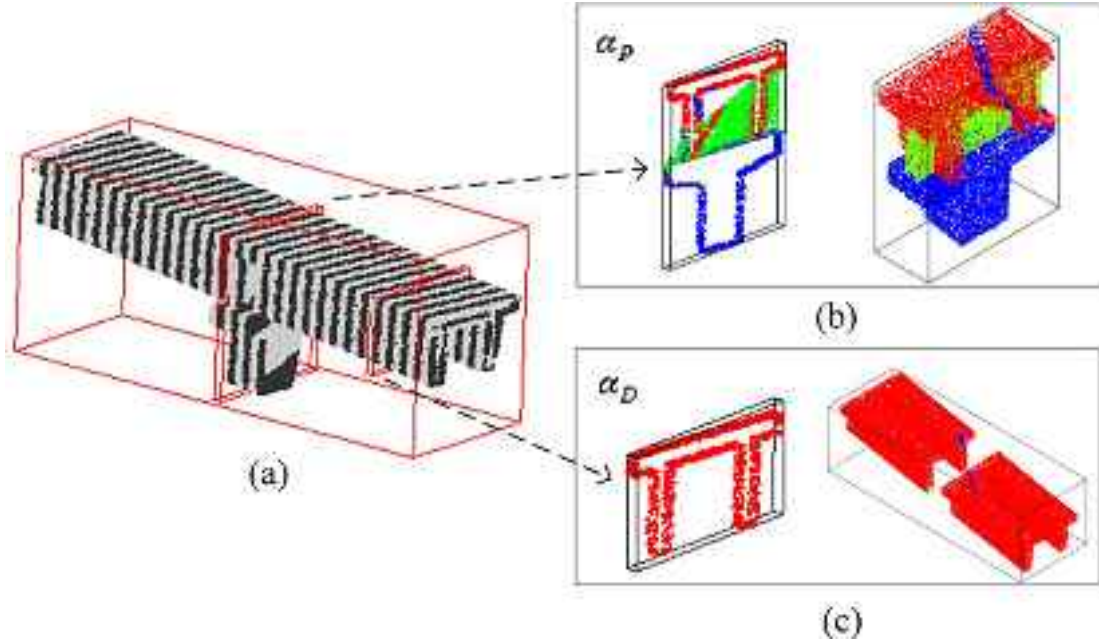


Fig. 5. (a) Slice along the X-axis; (b) groups with piers α_p ; (c) groups without piers α_D .

(a)). Using the height of each slice, the slices are divided into groups with piers α_p and groups without piers α_D :

$$H = |\max\{z_i|S\} - \min\{z_i|S\}| \quad (1)$$

$$f(S_i) = \begin{cases} S_i \in \alpha_p, \rho H < |\max\{z_i|S_i\} - \min\{z_i|S_i\}| \\ S_i \in \alpha_D, \rho H > |\max\{z_i|S_i\} - \min\{z_i|S_i\}| \end{cases} \quad (2)$$

where H is the height difference between the highest and lowest points of the whole bridge point cloud, ρ is a constant as a discrimination parameter to distinguish the two types of slices.

- (3) Iterative best rotation Angle: In order to more accurately determine the boundary information between components, the rotation Angle iteration will be carried out for the group containing piers α_p . As shown in Fig. 6, the groups with piers α_p were rotated around the X-axis and Y-axis respectively and projected onto the YZ plane, and the histogram of the point cloud density of the z axis was made respectively, that is, the z coordinate of the point was taken as the horizontal coordinate, and the number of point clouds in the unit area along the Z axis was taken as the vertical coordinate. Each rotation Angle corresponds to a histogram and at the same time corresponds to a standard deviation, and the size

of the standard deviation means the size of the coincidence degree of the point clouds projected on the YZ plane. The maximum standard deviation is obtained through iteration, that is, the optimal rotation Angle is obtained, and then rotate around the two coordinate axes respectively.

- (4) Boundary judgment: M components $D = \{D_i : i = 1, 2, \dots, M\}$ are separated according to target, and the point clouds of different components D_i are projected on the YZ plane using the first segmentation result. The density histogram of Z axis is also made. Boundary information can be obtained through the extreme value in the histogram, and the boundary error is positively correlated with the column width of the histogram. Instead of dividing the boundaries directly without presegmentation (Fig. 7), the boundaries of the separated constructed histogram are clearer (Fig. 8). The method of mapping according to component will correlate the boundary information with the component itself, which can greatly simplify the heuristic segmentation method, and there is no need to combine the information such as normal lines for complicated subsequent segmentation. Additionally, if the pre-segmentation results do not meet the requirements, the acquisition of extreme values will shift in the density histogram, and the lower the mIoU, the greater the offset.


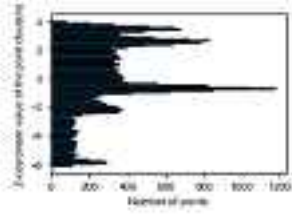

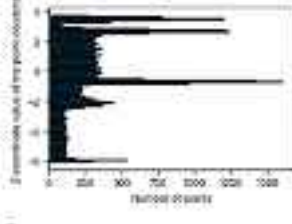
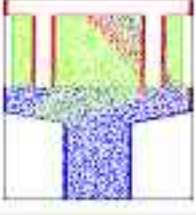
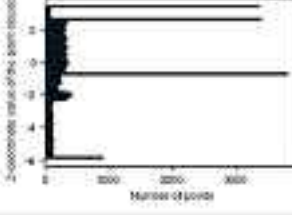
| Projection (YZ-plane) | Density histogram (z-axis) | Revolve around axis | | Standard deviation |
|---|---|---------------------|--------|--------------------|
| | | X-axis | Y-axis | |
|  |  | 0 | 0 | 212.36 |
|  |  | 1.4° | -4.6° | 289.97 |
|  |  | 3.3° | -7.9° | 579.00 |

Fig. 6. Best rotation iteration.

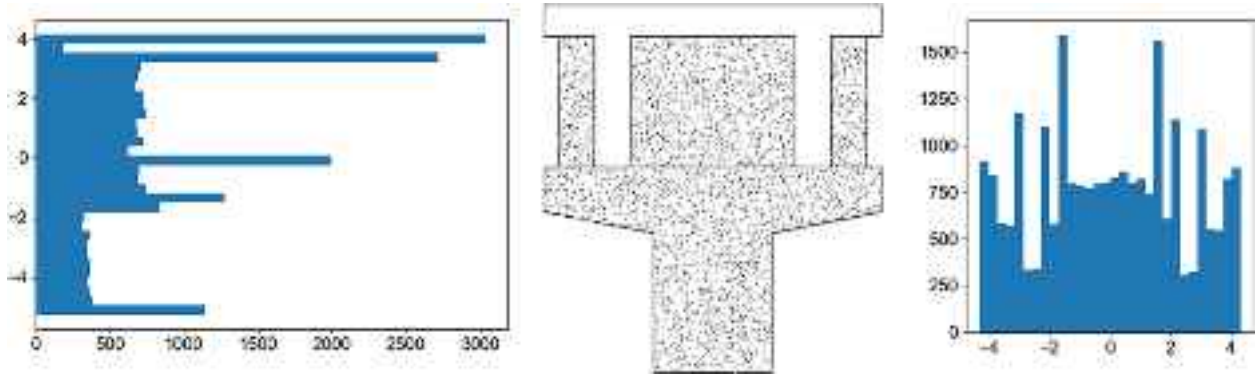


Fig. 7. Projections and density histograms in the Z and Y axes when no pre-segmentation is performed.

- (5) Grouping and classification: As shown in Fig. 9, the obtained boundary information is used to divide the point cloud of each component D_i into two parts: the misclassified point cloud $FD_i = \{FD_i|D_i\}$ and the correctly classified point cloud $TD_i = \{TD_i|D_i\}$, and then FD_i is further classified by boundary information. The classification method of FD_i changes with the change of components, and finally all components are combined. Furthermore, should the pre-segmentation outcomes fail to satisfy the specified criteria, it would be impossible to achieve precise boundaries in step four. Consequently, a portion of TD_i instances may be erroneously categorized as FD_i , which could compromise the overall accuracy of the classification results.
- (6) Return: Finally, the displacement and rotation information recorded in the process is reversed, and the geographical location information of the point cloud can be retained for subsequent work.

This classification method itself is to identify the boundary as the goal, so it can solve the problem that the boundary is not clear in the first segmentation, and the use of density histogram can reduce the impact of the error points in the first classification, get the boundary, eliminate the error points, and solve the problem mentioned at the beginning of this section. This method is also related to the effect of the first segmentation, that is, the better the first segmentation effect, the second segmentation can be closer to the truth, mainly in steps 4 and 5. The second segmentation result can be optimized while the first segmentation result is optimized.

4. Experiment and results

In this chapter, the beam bridge is chosen as the representative model, using the publicly available bridge point cloud dataset provided by Lu et al. [34]. To enable valid comparisons, the synthetic dataset mimics the same bridge components as those in the original study.

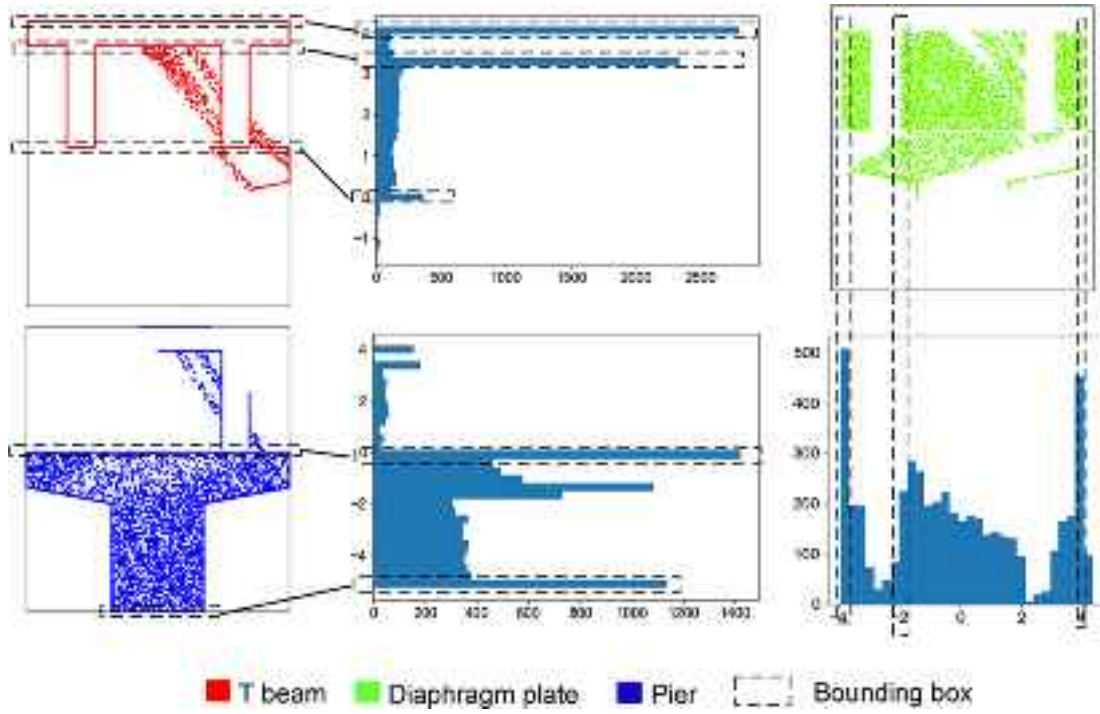


Fig. 8. The density histogram captures the boundary information.

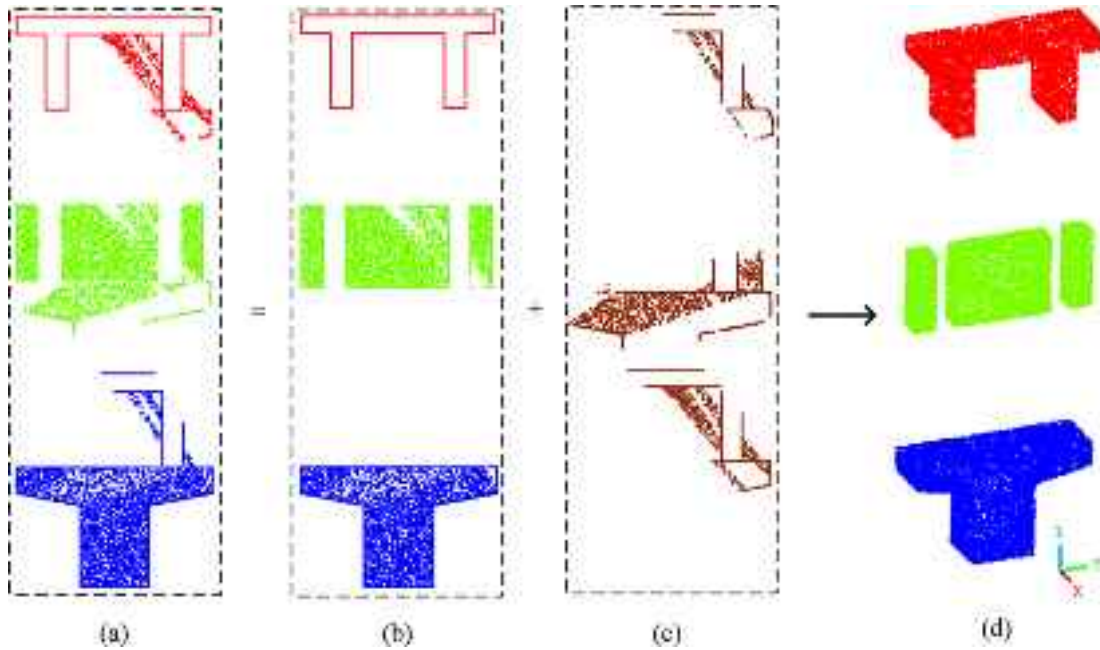


Fig. 9. (a) Pre-segmentation results for the group with piers α_p (b) the correctly classified point cloud TD_i (c) the misclassified point cloud FD_i (d) Correct segmentation results.

4.1. Build the point clouds and datasets

The objective of this section is to create multiple sets of different bridge point cloud data for use as the training set in Section 4.2. The experiment involves four types of components: beam plate, pier cap, cylindrical pier, and trapezoidal pier. Based on Fig. 10, the configurations are divided into two categories: one includes three components (beam, pier cap, and cylindrical pier) and the other includes two components (beam and trapezoidal pier). Each category consists of three sections: the short cantilever box beam section based on a real point

cloud (test set), the box beam section of a conventional structure, and the rectangular section, which excludes pier caps. Each section is subdivided into three configurations based on pier arrangement: 3×3 , 3×2 , and 2×2 . A dataset is produced for each configuration, as shown in Fig. 11. Each group, comprising 25 bridges, has five datasets removed due to inconsistencies, resulting in 20 datasets per group. Therefore, Train1 and Train2 each contain $3 \times 3 \times 20 = 180$ bridge datasets as training sets. The assignment of bridge point clouds is detailed in Table 1, where Group1 and Group2 are each divided into three training subsets A, B, and C, with A1 also referred to as Train1-A, utilized to

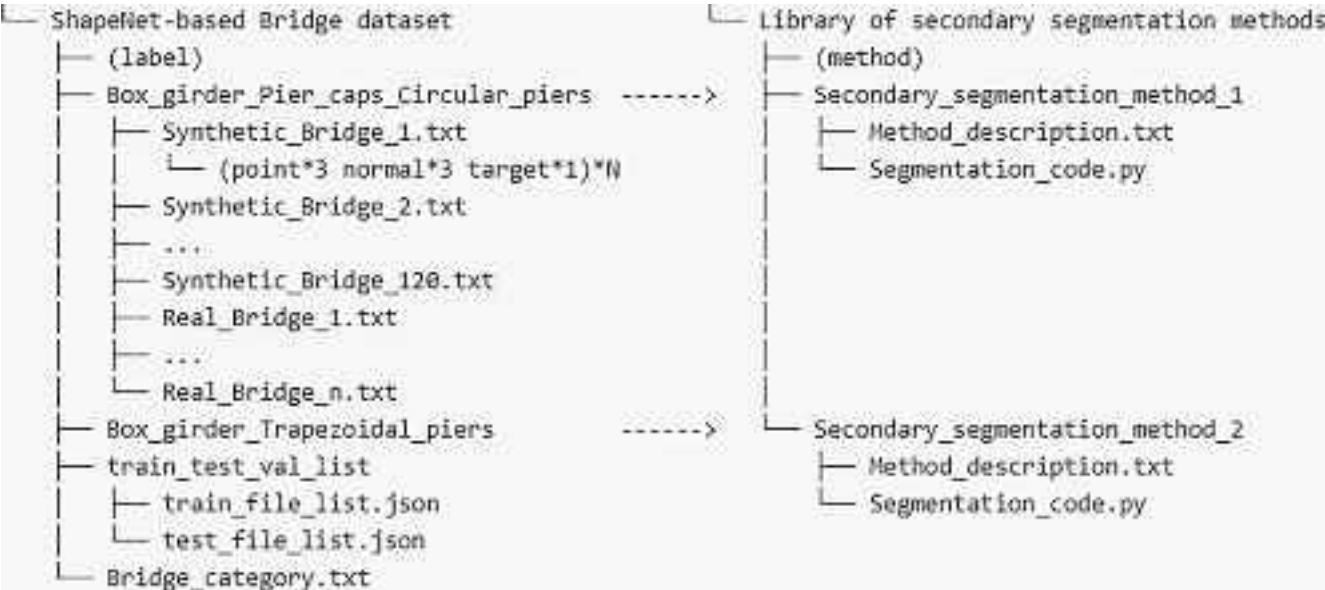


Fig. 10. Experiment file structure.

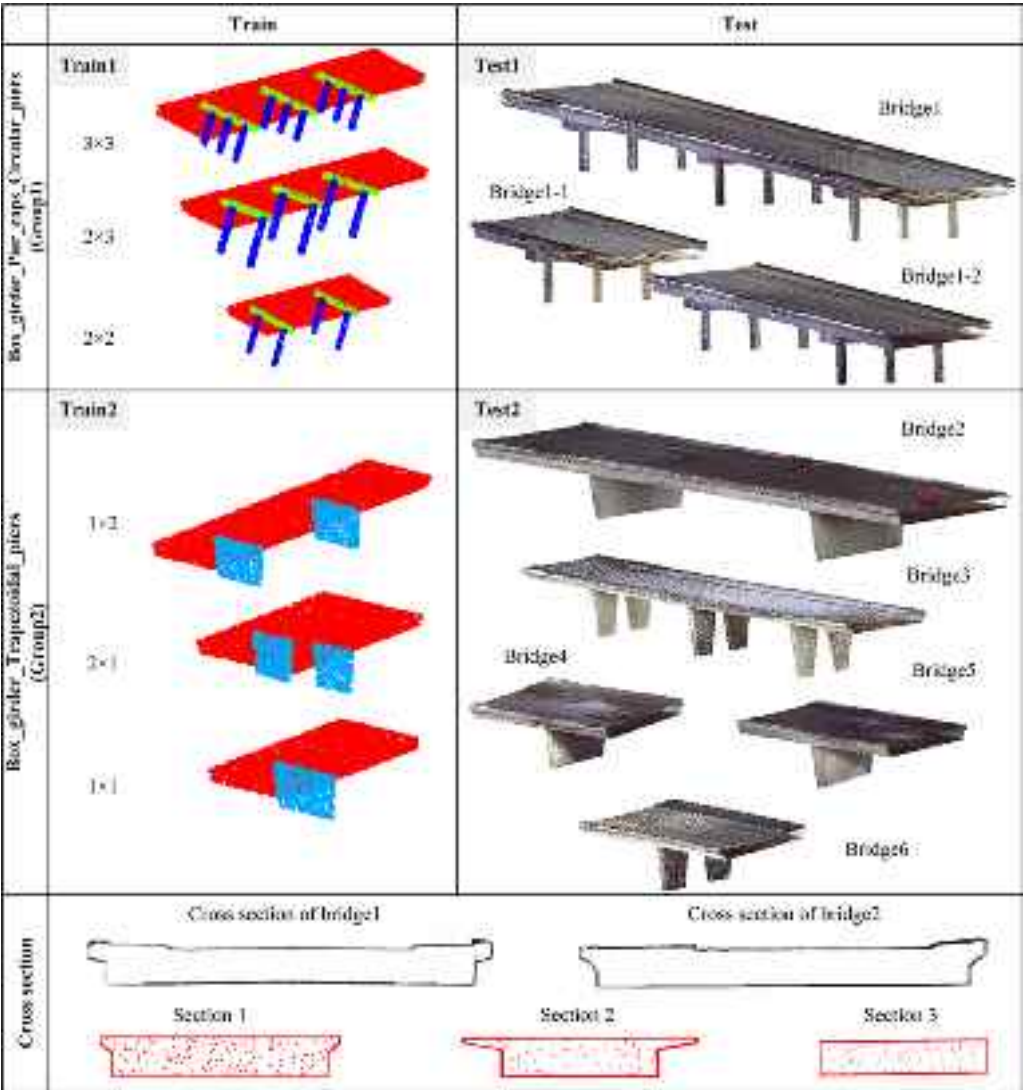


Fig. 11. The content of the training and test sets.

Table 1
Distribution and number of datasets.

| Group | | Training set | | | | | | | | | Test set | |
|--------|----|--------------|-------|-------|-----------|-------|-------|-----------|-------|-------|----------|------------------|
| | | Section 1 | | | Section 2 | | | Section 3 | | | | Total |
| | | 3 × 3 | 3 × 2 | 2 × 2 | 3 × 3 | 3 × 2 | 2 × 2 | 3 × 3 | 3 × 2 | 2 × 2 | | |
| Group1 | A1 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 180 | Bridge1 |
| | B1 | 30 | 30 | 30 | 30 | 30 | 30 | 0 | 0 | 0 | 180 | Bridge1-1 |
| | C1 | 60 | 60 | 60 | 0 | 0 | 0 | 0 | 0 | 0 | 180 | Bridge1-2 |
| Group2 | A2 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 180 | Bridge2, Bridge3 |
| | B2 | 30 | 30 | 30 | 30 | 30 | 30 | 0 | 0 | 0 | 180 | Bridge4, Bridge5 |
| | C2 | 60 | 60 | 60 | 0 | 0 | 0 | 0 | 0 | 0 | 180 | Bridge6 |

Note 1: The grouping and content correspond to Fig. 11.

Note 2: “section” represents different cross-sections; “n × n” represents the distribution of piers.

assess the impact of training set content on segmentation performance. The test sets are sourced from the public bridge point cloud dataset by Lu et al. [34], where Bridge1-1 and Bridge1-2 are modifications of Bridge1, and Bridge3 and Bridge4 are situated on curved road sections. The cross-sections of the beams in the test set resemble those of Bridge1 and Bridge2.

Using the 3x3 configuration in the conventional cantilever section as an example, the parameter settings and dataset production process are detailed in Table 2. Most parameters follow standard guidelines, with the exception of the Excel portion, which primarily sets initial data parameters based on current standards. The process for setting parameters and producing datasets is as follows: 1) In Excel, initial parameters are batch-set according to current standards. 2) Data is processed using Dynamo, and the adaptive entity model is automatically constructed using Revit, with connections exported in FBX format. 3) CC is used to adjust size, generate the point cloud model, and perform segmentation. 4) Python is employed to conduct the Data Optimization process depicted in Fig. 4, resulting in the final usable dataset.

4.2. Deep learning pre-segmentation

Using the dataset outlined previously, three point cloud deep learning networks—PointNet++, PCT, and DGCNN—are employed for segmentation tasks. To ensure consistency and control over the experiment’s variables, the training settings adhere to the parameters specified in the original publications of these network models [30–32]. The Stochastic Gradient Descent (SGD) optimizer was used, featuring an initial learning rate of 0.1 and a momentum of 0.9. This learning rate is progressively reduced to 0.001 using cosine annealing. The batch size is uniformly set at 4, and the training is conducted over 100 epochs. The experiments include downsampling the dataset to both 4096 and 8192 points for comparative analysis, with PointNet++ and PCT additionally incorporating normal information as part of the experimental setup.

In the segmentation tasks, mAcc and mIoU are selected as the evaluation metrics. Unlike typical deep learning-based segmentation tasks where the results are final, in this paper, the segmentation outputs are preliminary and serve as the basis for secondary segmentation. Consequently, the focus is on ensuring that these key indicators meet certain thresholds rather than on extensive performance metrics. For the secondary segmentation approach described herein, mIoU is crucial as it directly influences the effectiveness of the secondary segmentation, while mAcc serves as a supportive metric.

$$mAcc = \frac{1}{n} \sum_{i=1}^n \frac{P_{ii}}{\sum_{j=1}^n P_{ij}} \quad (3)$$

$$IoU = \frac{P_{ii}}{\sum_{j=1}^n P_{ij} + \sum_{j=1}^n P_{ji} - P_{ii}} \quad (4)$$

$$mIoU = \frac{1}{n} \sum_{i=1}^n \frac{P_{ii}}{\sum_{j=1}^n P_{ij} + \sum_{j=1}^n P_{ji} - P_{ii}} \quad (5)$$

where n denotes the number of bridge members and p_{ij} denotes that a point with label i is incorrectly predicted as label j .

Table 3 displays the original data from the experiment. The segmentation results for Group (2) show both mAcc and mIoU exceeding 97 %, indicating that the synthetic datasets alone can effectively handle the binary classification task, independent of the network model used. For the multi-class classification tasks represented by Group (1), the process is markedly more complex.

Firstly, regarding network performance, the data in Fig. 9 is based on original experimental results, as supplemented by Fig. 13. Fig. 12 (a) and (b) clearly indicate that PointNet++ achieves higher mAcc and mIoU under various conditions. Additionally, with the inclusion of normal information, only simple, purely synthetic datasets can attain an mIoU exceeding 80 %. Although PCT demonstrates superior segmentation capabilities in large datasets compared to the other two models, its performance with synthetic datasets is suboptimal, showing the lowest mIoU when normals are not included.

From the analysis of the response to normal information, relying solely on the 3D coordinates of the point cloud for segmentation tasks proves to be very unstable. Without normals, the mIoU for the A1 training group in PointNet, as shown in Fig. 12(b), falls significantly below the average. Similarly, DGCNN also underperforms in the B2-8192 experimental group. Moreover, a comparison of Fig. 12(d) and (f) shows that both PointNet and PCT enhance their segmentation effects with the inclusion of normals, with PointNet showing a greater degree of improvement. Notably, unlike PointNet++, PCT did not initially incorporate normals in its design. However, due to its simpler network framework and more adaptable mechanism, it easily integrates normals into its training parameters, thus positively impacting experimental outcomes. In contrast, graph convolutional networks such as DGCNN struggle to adapt to normal information, occasionally yielding negative results. Therefore, the responsiveness of PointNet++ to normals in synthetic datasets is also a critical indicator in these experiments.

Regarding the datasets, the mmIoU analysis in Fig. 12 and Table 3 illustrates that different datasets are better suited to specific network models. DGCNN, which derives geometric information from various dimensions, shows that the geometric characteristics of a dataset significantly influence segmentation outcomes. For instance, the C1 group, characterized by its uniform data content, achieves the highest mAcc and mIoU. Conversely, under conditions of 8192 points, PointNet++ approaches the ideal scenario, with segmentation gradually improving as the richness of the training set increases.

In summary, PointNet++ network models, characterized by their strong adaptability to synthetic datasets, heightened responsiveness to normals, and enhanced performance with richer training data, are particularly well-suited for pre-segmentation of synthetic data. Moreover, both 4096 points and 8192 points are sparse point clouds for

Table 2
Dataset construction process and parameters.

| Tool | Parameter or task | Parameter setting | Topological or parameter relationship |
|---------------|--|--|---------------------------------------|
| Excel | Box girder length (GL) | Random(40 m–80 m) | – |
| | Box girder width (GW) | Random(4.5 m–10.5 m) | – |
| | Girder bottom width (GBW) | BW* Random (0.8–0.6) | related to BW |
| | Height of girder (GBH) | Random(1 m–2 m) | – |
| | Flange plate thickness | Random (0.17 m–0.19 m) | related to GBH |
| | Height of web (WH) | GBH – Random(0.4 m–0.5 m) | related to GBH |
| | Width of pier cap (PCW) | 2*CPR*Random(1.2 m–1.4 m) | related to CPR |
| | Thickness of pier cap (PCT) | Random(0.6 m ~ 0.7 m) | – |
| | Circular pier radius (CPR) | Random(0.3 m–0.45 m) | – |
| | Height of the circular pier (CPH) | Random(3 m– 9 m) | – |
| | Position of pier | – | related to BL, GBW |
| Dynamo | Processing Excel data | Excel data to actionable list | related to adaptive families |
| | Calculation of each box girder position | Spacing = 20 m (Along the Y axis) | – |
| | Calculate pier cap position and offset | Random(–1 m to 1 m) (Along the X axis) | – |
| | Offset pier position | Random(–0.1 m ~ 0.1 m) | – |
| | Offset pier height | Random(–0.5 m to 0.5 m) | Based on CPH |
| | Assign parameters to the adaptive family | Dynamo to Revit data | related to adaptive families |
| Revit | Make adaptive families | – | related to Excel parameter |
| | Export the FBX file | Revit model to FBX model | – |
| Cloud Compare | Scale the model | 0.35 | Real size / model size |
| | Generate point clouds | Density = 2.85 pts/ square unit | related to number of training points |
| | component classification | Circular pier: 0, pier cap: 1, girder: 2 | related to number of components |
| | Output txt | Point*3 + target + normal*3 | Based on Fig. 10 |
| Python | Adjust txt format | Point*3 + normal*3 + target | Based on Fig. 10 |
| | Remove the coincident point cloud | D1 = 1, D2 = 0.3, D3 = 0.01 | related to point cloud density |
| | Rotate around the Y axis | Normal distribution (–6° 6°) | related to longitudinal slope |
| | Rotate around the Z axis | Random (0–2 π) | – |
| | Remove unreasonable data in Excel | 25–5 = 20 | – |
| | Create a total training set | 20*9 = 180 | Used in Section 4.2 |

Note: The point cloud density reported by CloudCompare reflects the density after model scaling.

bridge point clouds. When increasing the density on a small scale, the result may not increase linearly [35]. These models are expected to achieve pre-segmentation accuracy and intersection over union ratios exceeding 80 % and 70 %, respectively.

4.3. Secondary segmentation

The core of this section is to solve the two problems of point cloud confusion and unclear boundaries proposed in Section 3.4.

Taking Bridge1 as an example, PointNet++_normal is used to output 100 000 points for experiments in order to facilitate operation. Firstly, PCA is performed to eliminate the geographic information carried by the point cloud, and the displacement x_1 and rotation Angle β_1 are recorded in the process. Slice thickness $\delta = 0.5m$, $\rho = 0.30$ based on reference [34]. Based on equations (1) and (2), the group with piers α_{p1} (Fig. 14 (c)) and the group without piers α_{d1} are obtained.

Next, the iteration of the rotation Angle. During the experiment, it was found that the real bridge point cloud often fell into the local optimal solution due to the noise and irregular deformation of the bridge. Considering the small range of the rotation Angle [J, K], the slope rotation Angle (around the Y axis) only needed to be controlled at $J = -K = 6^\circ$ [34]. Using the exhaustive method can meet the accuracy requirements, the number of samples:

$$S_U = \frac{|J - K|}{P_R} \quad (6)$$

where P_R is precision, in degrees. For the projection of point clouds, the accuracy of Angle has no use value, but the recognition accuracy of point cloud spacing is valuable:

$$P_L = L \bullet \tan P_R \quad (7)$$

where P_L is minimum recognition precision in meters and L is length of bridge. To ensure that at $P_R = 0.1^\circ$, only 120 samples need to be uniformly taken and Bridge1 has a bridge length of 6 m, in which case $P_L \approx 0.01m$. Since the distribution of the standard deviation is close to normal under uniform sampling (Fig. 15), using normal sampling can further improve accuracy. The same thing is done around the X axis at $J = -K = 1^\circ$. The maximum standard deviation of 1309.14 was obtained by rotating $\beta_2 = 0.039^\circ$, $\beta_3 = -0.157^\circ$ around the x and y axes, respectively, by cross exhaustive.

After iteration, it can be seen from Fig. 16 that the boundary of each component has basically overlapped, and the point cloud after the optimal rotation transformation is denoted as α_{p2} . Similarly transform α_{d1} to α_{d2} for the concatenated model.

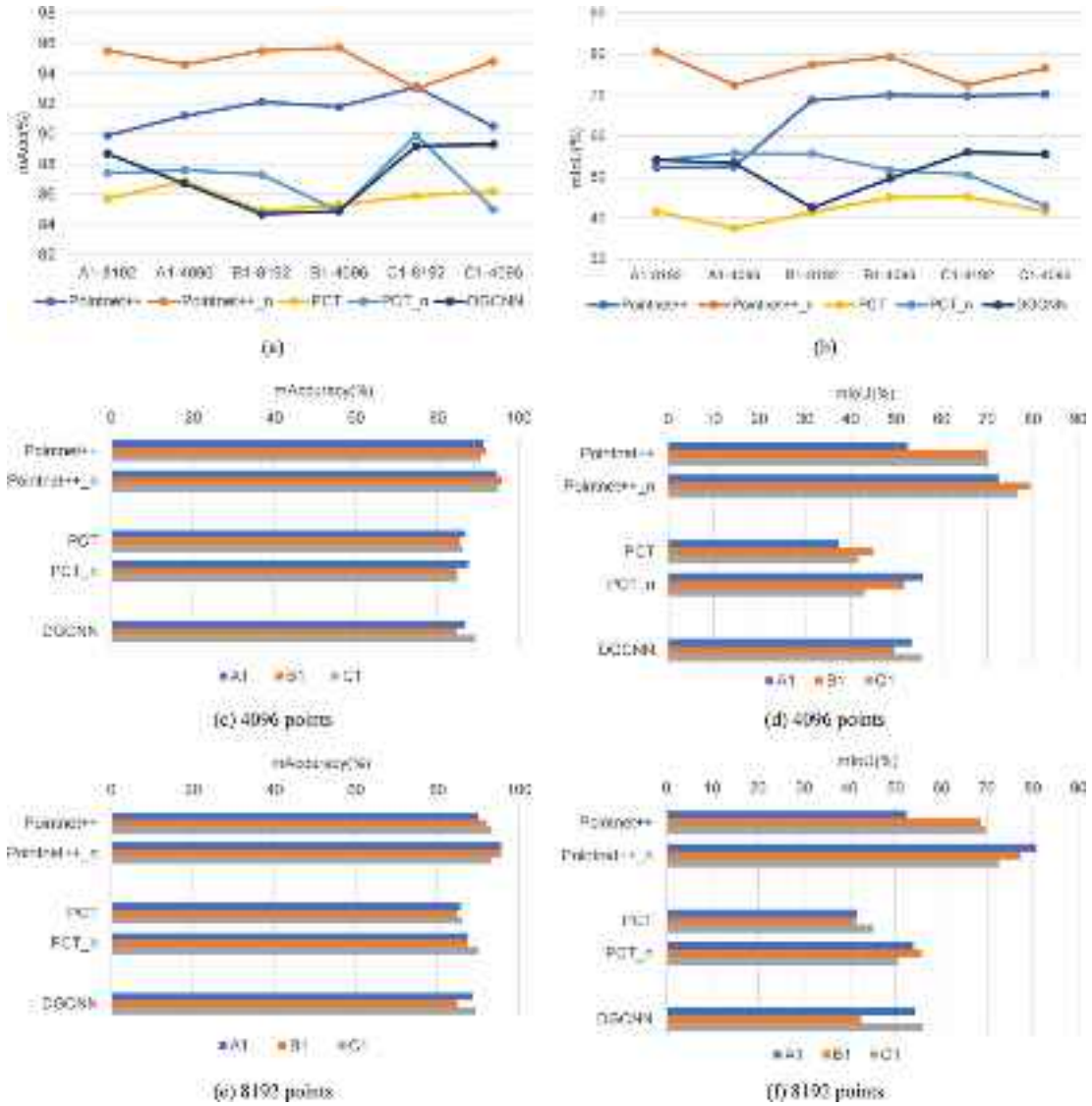
The boundary is not directly obtained through the histogram, but the maximum value $\rho_{max} = \{\rho_{max_i} : i = 1, 2, \dots, n\}$ of n histograms is first obtained through the number of components, and then the bounding box B is obtained. The axis coordinates of the upper and lower bounds of the bounding box are B_U, B_L , respectively. In the real point cloud, the boundary information is not as ideal as the boundary information of the synthetic point cloud in Fig. 8. With the decrease of the histogram width, the concentration of the point cloud will be relatively reduced, so the boundary points of the real point cloud will not be a line but a small range of surface after projection, and the points in the bounding box are mainly the boundary points of the bridge components. For Bridge1 Fig. 17, $B = \{B_1 : \text{Bound box 1}, B_2 : \text{Bound box 2}\}$, $B_U = \{B_{1U} : 4.91, B_{2U} : 3.93\}$, $B_L = \{B_{1L} : 4.73, B_{2L} : 3.85\}$. There are two issues involved in the process of finding B:

1) The value of the bin of the histogram. Although the bounding box would ideally be more accurate with a smaller column width, the accuracy of the iteration step for point cloud density and rotation Angle is limited. Point cloud density can basically reach the accuracy $P_p = 0.001m$ with the current point cloud acquisition equipment, while the accuracy of Angle iteration is P_L , and the overall accuracy is $P_T = \max\{P_p, P_L\}$. To ensure the accuracy of the boundary points, then

Table 3

Each network model uses synthetic data segmentation of the experimental results of the original data.

| 4096 points | | | | | | | | | | | | | |
|-------------|------------|-------|--------------------|-------------|-------|-------|-------------|-------|-------|-------|-------|--|-------|
| DS | PointNet++ | | PointNet++(normal) | | PCT | | PCT(normal) | | DGCNN | | mmACC | | mmIoU |
| | mAcc | mIoU | mAcc | mIoU | mAcc | mIoU | mAcc | mIoU | mAcc | mIoU | | | |
| A1 | 91.2 | 52.6 | 94.6 | 72.4 | 86.9 | 37.5 | 87.6 | 55.8 | 86.7 | 45.4 | 89.4 | | 52.7 |
| B1 | 91.8 | 70.0 | 95.7 | 79.3 | 85.3 | 45.0 | 84.9 | 51.7 | 84.9 | 45.9 | 88.5 | | 58.4 |
| C1 | 90.5 | 70.2 | 94.8 | 76.5 | 86.2 | 41.7 | 85.0 | 43.0 | 89.3 | 52.0 | 89.2 | | 56.7 |
| A2 | 99.46 | 99.22 | 98.50 | 99.24 | 99.28 | 98.48 | 98.90 | 99.83 | 98.51 | 99.05 | 98.93 | | 99.16 |
| B2 | 99.90 | 98.54 | 98.64 | 99.01 | 99.78 | 99.96 | 98.50 | 98.49 | 99.82 | 98.61 | 99.33 | | 98.92 |
| C2 | 99.03 | 98.1 | 99.94 | 99.97 | 99.14 | 98.39 | 99.55 | 99.13 | 99.54 | 99.12 | 99.44 | | 98.94 |
| 8192 points | | | | | | | | | | | | | |
| A1 | 89.9 | 52.4 | 95.5 | 80.7 | 85.7 | 41.6 | 87.4 | 54.0 | 88.7 | 54.2 | 89.4 | | 56.6 |
| B1 | 92.1 | 68.7 | 95.5 | 77.4 | 84.9 | 41.4 | 87.3 | 55.7 | 84.7 | 42.5 | 88.9 | | 57.1 |
| C1 | 93.1 | 69.8 | 93.0 | 72.5 | 85.9 | 45.2 | 89.9 | 50.6 | 89.2 | 56.0 | 90.2 | | 58.8 |
| A2 | 98.91 | 98.32 | 99.19 | 98.25 | 98.90 | 98.03 | 99.36 | 99.63 | 99.35 | 97.72 | 99.14 | | 98.39 |
| B2 | 99.78 | 98.54 | 99.90 | 97.77 | 99.00 | 99.36 | 99.23 | 99.93 | 99.90 | 98.54 | 99.56 | | 98.83 |
| C2 | 99.01 | 98.07 | 99.95 | 99.91 | 98.7 | 97.61 | 99.56 | 99.13 | 99.8 | 99.59 | 99.40 | | 98.86 |

**Fig. 12.** Comparison of training effects of each network model under different conditions.

$$bin = 2P_T \quad (8)$$

2) The value of the lower limit ρ_{min} of the column height of the upper and lower bounds of the bounding box. The value of the column height

should be based on the maximum of the histogram and correlated with the bin height, that is $\rho_{min} \propto \rho_{max} \cdot \rho_{min} \cdot bin$. Combined with the experiment can be obtained:



Fig. 13. Demo: Split Bridge 1 using PointNet++ (using normals).

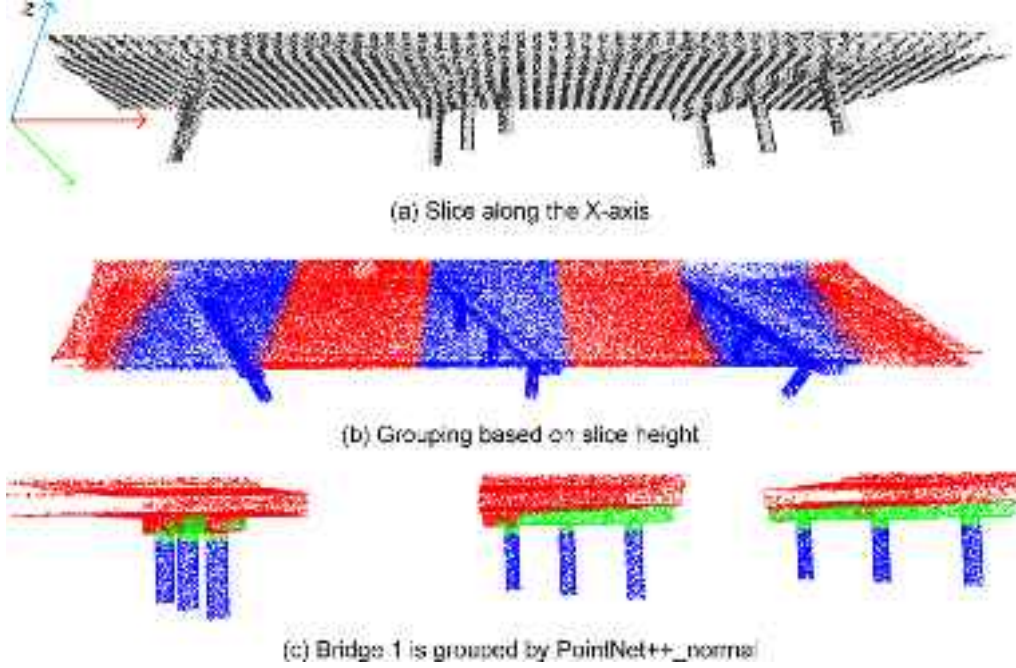


Fig. 14. Point cloud slicing and grouping based on pre-segmentation results.

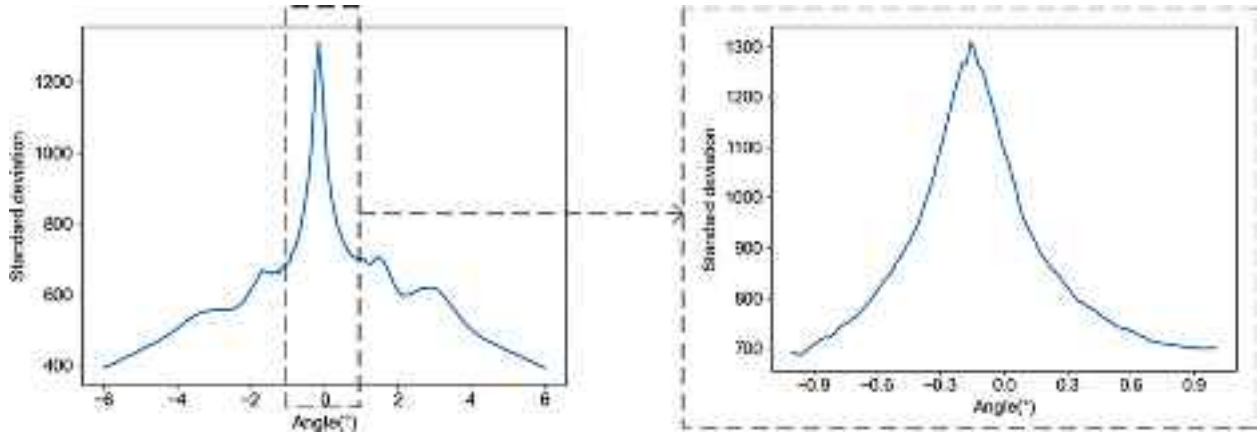


Fig. 15. Relation between rotation Angle and standard deviation under uniform sampling of α_1 .

$$\rho_{min} = 10\rho_{max} \bullet bin \quad (9)$$

After determining the boundary information, the point cloud of each component D_i is divided into two parts, FD_i and TD_i , by using the boundary information B_i obtained by each component d_i itself, and then the FD_i is correctly classified by using the boundary information obtained by other components. Finally, $x1, \beta1, \beta2, \beta3$ are used to do the inverse transformation to restore the geographical location information of the bridge.

Since the whole secondary segmentation process is more concise and the error is easier to control than the general heuristic method, it is more robust than the pure deep learning method or heuristic method. In this paper, the precision is controlled by the precision parameter $S_U, P_L, bin, \rho_{min}$ all realize the association with P_R , and realize the secondary segmentation while controlling the global precision. Fig. 18 shows the segmentation result at $P_R = 0.1^\circ$, which illustrates the usability of the proposed method. However, the error between the beam (red) and the

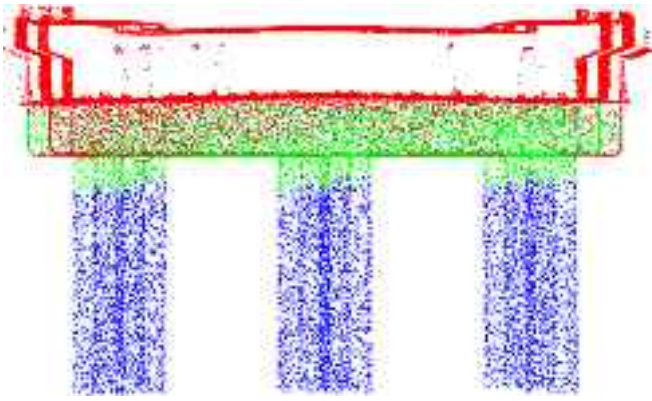


Fig. 16. Projection of α_{p2} into the YZ plane.

pier cap (green) also shows that a straight boundary is not suitable for all cases, and related work on curved boundaries is needed in the future.

Replicating the original code, the runtime for the pure heuristic method is approximately 42.3 s. In contrast, Table 4, show that the average total runtime for the two-stage method is 16.78 s. All code

executions were conducted under identical hardware conditions on point clouds downsampled to 100,000 points (entire test set), with target components consistent with the original. By comparing the time taken by the heuristic segmentation method, the average time saved per bridge was 63.2 %. With the same segmentation accuracy, the total segmentation time was reduced by 60.3 %, and the processing speed was

Table 4

Final segmentation results for 8 real-world bridges (100,000 points each).

| 10^5 points | mAcc | mIoU | Average pre- segmentation time (s) | Secondary segmentation time(s) | Total time(s) |
|------------------|-------|-------|--|--------------------------------------|------------------|
| Bridge1 | 99.54 | 99.13 | 1.2 | 18.4 | 19.6 |
| Bridge1- 1 | 99.63 | 99.58 | | 17.7 | 18.9 |
| Bridge1- 2 | 99.55 | 99.28 | | 17.5 | 18.7 |
| Bridge2 | 99.89 | 99.71 | | 14.5 | 15.7 |
| Bridge3 | 99.93 | 99.85 | | 14.4 | 15.6 |
| Bridge4 | 99.91 | 99.86 | | 14.1 | 15.3 |
| Bridge5 | 99.87 | 99.69 | | 13.9 | 15.1 |
| Bridge6 | 99.92 | 99.88 | | 14.1 | 15.3 |
| Average | 99.78 | 99.62 | 1.2 | 15.58 | 16.78 |

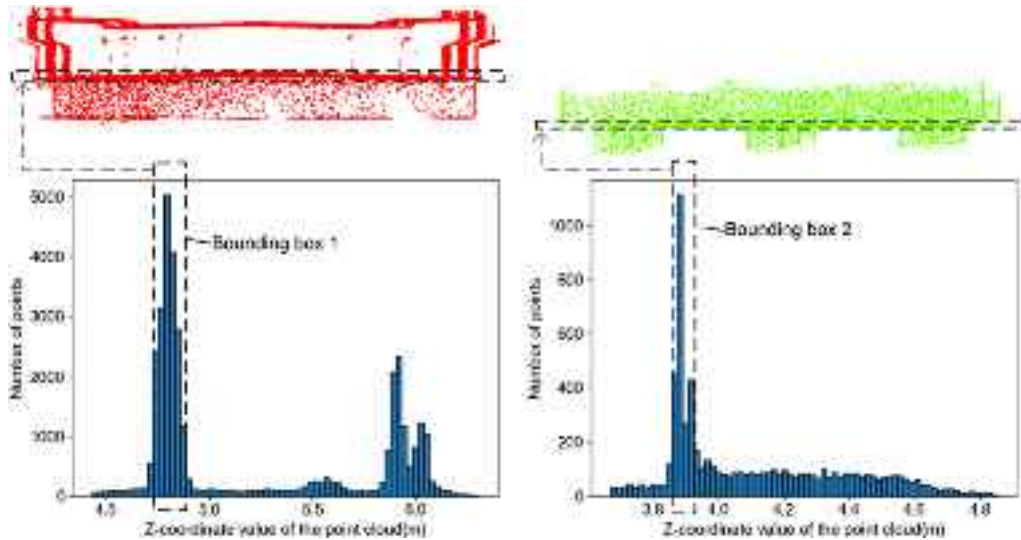


Fig. 17. The density histogram is made according to the target based on the pre-segmentation results.

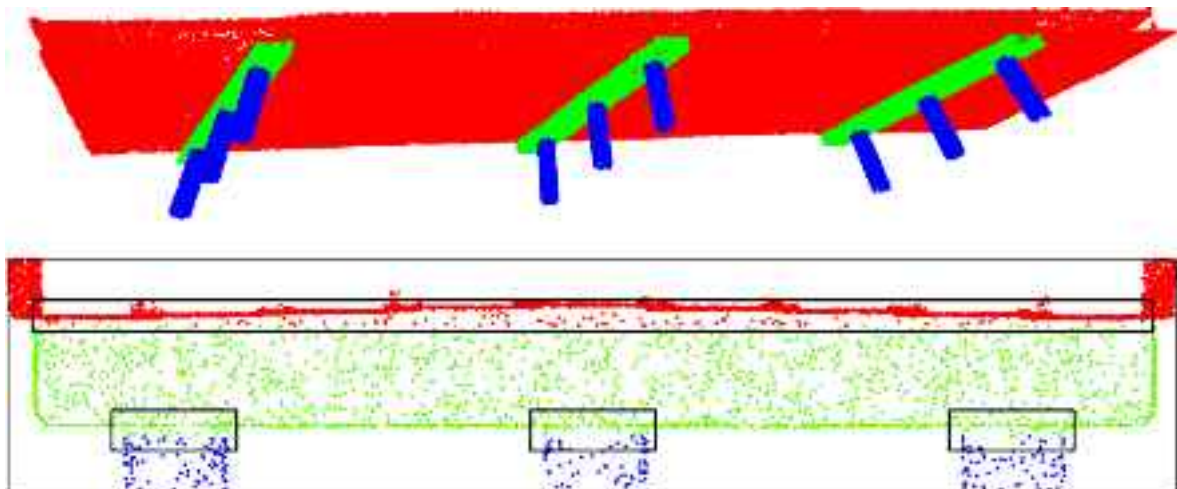


Fig. 18. Segmentation result of Bridge1 and slight error at the boundary.

2.52 times faster than the pure heuristic method. The mAcc of the heuristic method is about 99.45 %, while the mAcc and mIoU for the two-stage method are 99.54 % and 99.13 %, respectively.

5. Conclusion

In this paper, we propose a two-stage segmentation approach for bridge component point clouds, leveraging both deep learning and heuristic methods. And this method is used for component segmentation from 8 real-world bridges. The results all achieve the ideal effect, which proves the robustness of the method. Through a detailed analysis and application of this proposed method, the following conclusions are obtained:

- (1) Clarify the implementation process of this method. Initially, a substantial dataset required for training is synthesized from actual bridge point clouds. Subsequently, a suitable deep learning network is employed for pre-segmentation, requiring only that the mIoU exceed 70 %. Based on these preliminary results, a streamlined heuristic method is then applied to achieve the final segmentation. We conducted experiments on eight real-world bridges, and the results consistently met our expectations, thereby demonstrating the robustness of our method. Notably, the preliminary segmentation not only simplifies the subsequent heuristic segmentation but also diminishes the reliance on the accuracy of the dataset and the performance of the deep learning network within the segmentation process. The integration of these two methods—rather than merely concatenating them or using them independently—marks a significant departure from existing techniques
- (2) First, the bridge models are created in bulk using Excel, Dynamo, and Revit. Second, the models are converted into point clouds and preprocessed using CloudCompare. Finally, Python is used to make the synthetic data more similar to real-world point clouds, creating a point cloud dataset. Synthetic data sets can solve the problem of insufficient data sets, but the synthetic data sets cannot meet the requirements of deep learning for the accuracy of training data and the quantity of training data at the same time. However, the low dependency of the two-stage method on the performance of deep learning segmentation makes the method well-suited to ignore the precision issue.
- (3) First, due to the relatively low requirement for pre-segmentation accuracy, the criterion for selecting deep learning networks for screening is: simple structure but stable in achieving the required pre-segmentation accuracy. Three representative networks were selected based on network mechanisms for quantitative comparative experiments. The experimental conclusion indicates that the PointNet++ network incorporating normal information is more appropriate for use as a pre-segmentation network. Second, heuristic methods exhibit high segmentation accuracy but suffer from slow segmentation speed. However, the pre-segmented point cloud significantly reduces the time required for iterative angle and boundary information acquisition, thereby saving approximately 63.2 % of the time in the heuristic segmentation process. In other words, the two-stage segmentation method substitutes some of the complex steps in the purely heuristic approach with deep learning techniques.
- (4) The average segmentation mIoU achieved by this method is ≥ 99 %, and the segmentation speed is 2.52 times faster than that of conventional heuristic methods, thereby realizing both high accuracy and high speed in automatic segmentation.

CRedit authorship contribution statement

Tian Zhang: Writing – original draft, Supervision, Funding acquisition. **Haonan Chen:** Writing – original draft. **Pengfei Li:** Writing –

review & editing, Resources, Funding acquisition. **Haijiang Li:** Software.

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.

Acknowledgement

The research work in this paper was supported by the National Natural Science Foundation of China (Grant No. 51608087), the Open Project of National Engineering Research Center of High-speed Railway Construction Technology in China (No. HSR202010), the Scientific Research Project from the Educational Department of Liaoning Province in China (No. LJKMZ20220379), and the Scientific Research Project from the Transportation Department of Liaoning Province in China (No. 2024-09).

Data availability

Data will be made available on request.

References

- [1] 2019 Transport Industry Development Statistical Bulletin [Online]. Available: https://xxgk.mot.gov.cn/2020/jigou/zhghs/202006/20200630_3321335.html.
- [2] 2022 Transport Industry Development Statistical Bulletin [Online] Available: https://xxgk.mot.gov.cn/2020/jigou/zhghs/202306/t20230615_3847023.html.
- [3] Opinions of the Ministry of Transport on further improving the safety and durability level of highway Bridges [Online]. Available: https://xxgk.mot.gov.cn/2020/jigou/glj/202012/t20201228_3509089.html.
- [4] W. Cao, Three years to build a number of demonstration channels and networks, ed: China transportation paper, p. 001.
- [5] M.S. Mafipour, S. Vilgertshofer, A. Borrmann, Automated geometric digital twinning of bridges from segmented point clouds by parametric prototype models, *Autom. Constr.* 156 (2023) 105101.
- [6] Y. Xu, U. Stilla, Toward Building and Civil Infrastructure Reconstruction From Point Clouds: A Review on Data and Key Techniques, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 14 (2021) 2857–2885.
- [7] L. Zhang, C. Yang, H. Lu, X. Ruan, M.-H. Yang, Ranking Saliency, *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (9) (2017) 1892–1904.
- [8] H. Kim, C. Kim, Deep-learning-based classification of point clouds for bridge inspection, *Remote Sensing* 12 (22) (2020).
- [9] Y. Zhou, A. Ji, L. Zhang, Sewer defect detection from 3D point clouds using a transformer-based deep learning model, *Autom. Constr.* 136 (2022) 104163.
- [10] J.B. Haurum, M.M.J. Allahham, M.S. Lyne, K.S. Henriksen, I.A. Nikolov, T. B. Moeslund, Sewer Defect Classification using Synthetic Point Clouds, in: *16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP) / 16th International Conference on Computer Vision Theory and Applications (VISAPP)*, 2021, pp. 891–900.
- [11] D. Lamas, A. Justo, M. Soñán, B. Riveiro, Automated production of synthetic point clouds of truss bridges for semantic and instance segmentation using deep learning models, *Autom. Constr.* 158 (2024) 105176.
- [12] L. Zhao, W. Tao, I. Assoc Advancement Artificial, JSNet: Joint Instance and Semantic Segmentation of 3D Point Clouds, in: *34th AAAI Conference on Artificial Intelligence / 32nd Innovative Applications of Artificial Intelligence Conference / 10th AAAI Symposium on Educational Advances in Artificial Intelligence*, New York, vol. 34, 2020, pp. 12951–12958.
- [13] D. Lamas, A. Justo, M. Soñán, M. Cabaleiro, B. Riveiro, Instance and semantic segmentation of point clouds of large metallic truss bridges, *Autom. Constr.* 151 (2023) 104865.
- [14] 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, *Automation in Construction* 142 (2022).
- [15] L. Landrieu, M. Simonovsky, Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs, presented at the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
- [16] 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, *Automation in Construction* 150 (2023).
- [17] Q. Hu, et al., RandLA-Net: efficient semantic segmentation of large-scale point clouds, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2019 (2020) 11105–11114.
- [18] K. Liu, Z. Gao, F. Lin, B.M. Chen, FG-Net: a fast and accurate framework for large-scale LiDAR point cloud understanding, *IEEE Trans. Cybern.* 53 (1) (2023) 553–564.

- [19] Y. Jing, B. Sheil, S. Acikgoz, Segmentation of large-scale masonry arch bridge point clouds with a synthetic simulator and the BridgeNet neural network, *Automation in Construction* 142 (2022).
- [20] J. Chen, Q. Su, Y. Niu, Z. Zhang, J. Liu, A handheld LiDAR-based semantic automatic segmentation method for complex railroad line model reconstruction, *Remote Sens. (Basel)* 15 (18) (2023) 4504.
- [21] J. Grandio, B. Riveiro, M. Soilán, P. Arias, Point cloud semantic segmentation of complex railway environments using deep learning, *Autom. Constr.* 141 (2022) 104425.
- [22] Y. Xie, J. Tian, X.X. Zhu, Linking points with labels in 3D: a review of point cloud semantic segmentation, *IEEE Geosci. Remote Sens. Mag.* 8 (4) (2020) 38–59.
- [23] S. Qiu, X. Liu, J. Peng, et al., Fine-grained point cloud semantic segmentation of complex railway bridge scenes from UAVs using improved DGCNN, *Struct. Control Health Monit.* 2023 (2023) 3733799.
- [24] China Standard (JTG D60-2015). General Specifications for Design of Highway Bridges and Culverts, 2015.
- [25] China Standard (CJJ 37-2012). Code for design of urban road engineering, 2012.
- [26] China Standard (JTG B01-2014). Technical Standard of Highway Engineering, 2014.
- [27] W. Wang, L. Li, Review of deep learning in point cloud classification, *Comput. Eng. Appl.* 58 (01) (2022) 26–40.
- [28] 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 (12) (2021) 4338–4364.
- [29] W. Liu, J. Sun, W. Li, T. Hu, P. Wang, Deep learning on point clouds and its application: a survey, *Sensors* 19 (19) (2019).
- [30] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, in: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, California, USA, 2017, pp. 5105–5114.
- [31] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph CNN for learning on point clouds, *ACM Trans. Graph.* 38 (5) (2019) 146.
- [32] M.H. Guo, J.X. Cai, Z.N. Liu, T.J. Mu, R.R. Martin, S.M. Hu, PCT: point cloud transformer, *Computational Visual Media* 7 (2) (2021) 187–199.
- [33] M. Mundt, S. Majumder, S. Murali, P. Panetsos, V. Ramesh, I.C. Soc, Meta-learning Convolutional Neural Architectures for Multi-target Concrete Defect Classification with the CONcrete DEfect BRIDGE IMage Dataset, in: 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, California, USA, 2019, pp. 11188–11197.
- [34] R. Lu, I. Brilakis, C.R. Middleton, Detection of structural components in point clouds of existing RC bridges, *Comput. Aided Civ. Inf. Eng.* 34 (3) (2018) 191–212.
- [35] Y. Yan, H. Yan, J. Guo, H. Dai, Classification and segmentation of mining area objects in large-scale spars lidar point cloud using a novel rotated density network, *ISPRS Int. J. Geo Inf.* 9 (3) (2020) 182.