Detection of Key Components of Existing Bridge in Point Cloud Datasets

Ruodan. Lu[1], Ioannis Brilakis[2], Campbell R. Middleton[3]

1) Ph.D. Candidate, Department of Engineering, Cambridge University, UK. Email: rl508@cam.ac.uk
2) Laing O'Rourke Reader, Department of Engineering, Cambridge University, UK. Email: ib340@cam.ac.uk
3) Laing O'Rourke Center Director, Department of Engineering, Cambridge University, UK. Email: crm11@cam.ac.uk

Abstract:

The cost and effort for modelling existing bridges from point clouds currently outweighs the perceived benefits of the resulting model. Automating the point cloud-to-Bridge Information Models process can drastically reduce the manual effort and cost involved. Previous research has achieved the automatic generation of surfaces primitives combined with rule-based classification to create labelled construction models from point clouds. These methods work very well in synthetic dataset or idealized cases. However, real bridge point clouds are often incomplete, and contain unevenly distributed points. Also, bridge geometries are complex. They are defined with horizontal alignments, vertical elevations and cross-sections. These characteristics are the reasons behind the performance issues existing methods have in real datasets. We propose to tackle this challenge via a novel top-down method for major bridge component detection in this paper. Our method bypasses the surface generation process altogether. Firstly, this method uses a slicing algorithm to separate deck assembly from pier assemblies. It then detects pier caps using their surface normal, and uses oriented bounding boxes and density histograms to segment the girders. Finally, the method terminates by merging over-segments into individual labelled point clusters. Experimental results indicate an average detection precision of 99.2%, recall of 98.3%, and F1-score of 98.7%. This is the first method to achieve reliable detection performance in real bridge datasets. This sets a solid foundation for researchers attempting to derive rich IFC (Industry Foundation Classes) models from individual point clusters.

Keywords: BrIM, point cloud data, object detection, recursive segmentation, top-down

1. INTRODUCTION

The global infrastructure market is poised for an explosive adoption of BrIM (Bridge Information Models). However, there is only 20% BIM (Building Information Models) usage in the US on infrastructure projects (McGraw Hill Construction, 2014). Based on this demographic, the usage of BrIM is very limited despite the implementation of laser-scanning-based data collection. This is because that the automatic generation of BrIM from point cloud data (PCD) remains an unsolved problem. The time required to manually detect objects in a PCD and convert them to as-is 3D solid models using cutting edge modelling software tends to be ten times greater than that required to obtain the point cloud (Lu & Brilakis, 2017).

There are more than 600,000 highway bridges in the United States (US) (FHWA, 2013). According to an in-house report, Highways England manages more than 30,000 road bridges on its motorways and major A-roads. Based on a two-year inspection cycle, there is a need for at least 315,000 bridge inspections per annum across the US and England alone. This explains why there is a huge market demand for a less labour-intensive bridge documentation technique, which can efficiently boost bridge management productivity.

In general, the from-PCD-to-BrIM modelling process consists of two steps: 1) detecting bridge components in point clouds in the form of labelled point clusters; and 2) generating a geometric BrIM through fitting IFC (Industry Foundation Classes) entities and spatial relationships in labelled point clusters. This study intends to automate Step 1, i.e. bridge component detection in the PCD, which is currently largely achieved manually using modelling software.

Major vendors such as Autodesk, Bentley and ClearEdge3D provide the most advanced software solutions for BIM modelling. These tools can automatically recognize standardized geometric shapes for building and industrial elements embedded in point clusters. However, this is largely assisted by manually segmenting the point cloud in advance. Modelers need to repeatedly rotate the PCD to various views and try to select regions of interest using clipping polygons.

Existing methods usually segment a PCD into parametric surface primitives. They then classify the surfaces or the shapes converted from the surfaces using rule sets. These surface-based segmentation methods and rule-based classification methods perform quite well in case studies under strict constraints. However, given that a real bridge PCD is not as perfect as synthetic data and the topology of real bridges is far more complex than idealized cases, these methods exhibit poor performance when they are subject to real data.

In this paper, we address the challenges above using a novel top-down segmentation method. We use a slicing algorithm to tackle the complexity of bridge topology. Experiment results suggest that our method can push forward the progress of the automation in bridge modelling, as it can effectively segment bridge components from a PCD and directly label them without converting points into surfaces.

2. BACKGROUND

Much effort has been devoted to automating the process of object detection. We define 'detection' in this context as the combination of clustering (point cloud-to-point clusters) and classification (labelling the clusters). Current methods of PCD clustering generally follow a "bottom-up" approach, which goes from points to surfaces or patches followed by semantic labelling to derive objects. Most PCD classification methods follow a "top-down" approach, which employs human knowledge (e.g. relationships, context) to detect specific instances embedded in a PCD or infer the semantics of components in a geometric model.

2.1 Bottom-up detection

The bottom-up approach pieces together primitive features like points to generate higher-level features successively, such as surface normals, meshes, patches and non-uniform B-Spline surfaces (Dimitrov et al., 2016).

Zhang et al. (2015) present a sparsity-inducing optimization-based method to detect parametric planar patches in noisy bridge PCD. However, this method can work only with planar-surface objects and cannot detect piers when point densities of these regions are low. Walsh et al. (2013) present a Region Growing (RG) algorithm to detect objects in a PCD. However, this method cannot detect the edge between a pier cap and a pier. The segmentation was finally achieved after manually choosing key points. Likewise, Dimitrov and Golparvar-fard (2015) suggest an upgraded RG method. This method can deal with curved surfaces and it excels when the input PCD does not suffer from substantive occlusions. However, the method over-segments objects when occlusions are present and the data is incomplete. – These persistent problems in real point clouds were addressed by Xiong et al. (2013) through a learning-paradigm that detects occluded planar surfaces in a building PCD. However, their method cannot be applied in bridge settings, because occluded surfaces in a bridge PCD do not follow a specific pattern like in a building PCD. Schnabel et al. (2007) detect basic shapes using RANdom SAmple Consensus (RANSAC). However, given the computational-expensive nature of RANSAC, it is unrealistic to use it to detect complex geometries. Hence, these methods tend to perform well in relatively simplified scenarios and synthetic data, but are not ready to tackle the complexity of real bridges whose as-constructed and as-weathered shapes further increase the as-designed complexity.

2.2 Top-down detection

In general, the bottom-up detection is not suitable for PCD classification task. Classification through surfaces are insufficient, because it is difficult to determine whether they belong to the same instance. The intervention of object-level information is required to overcome such challenges. The top-down approach usually combines a set of engineering criteria and classifies objects in a PCD that meet the criteria. Prior studies show that knowledge-based classification methods are robust, as domain-specific information such as topological relationships (Koppula et al., 2011) and known parameters (e.g. diameter, direction) (Ahmed et al., 2014) are invariant to pose and appearance. Belsky et al., (2014) encapsulate domain expert knowledge in the form of rule sets in order to enrich semantics for a building model. However, these aforementioned methods are tailored for buildings and industrial objects and cannot be applied in bridge settings as the geometric properties of bridge components are quite different to those objects. Recent studies start to employ top-down strategy to detect bridge components in point clouds. For example, Riveiro et al. (2016) use specific constraints to segment masonry bridge PCD into surfaces. However, their algorithm is based on histograms, which largely depend on data quality. It is difficult to generalize this algorithm to large highway bridges, as the real point clouds are often incomplete and suffer from non-uniformly distributed points. Ma et al. (2017) leverage relationship knowledge and shape features to classify bridge 3D solid objects. First, the input of this method should be a solid bridge model but not a bridge PCD. Second, the method assumes the pairwise relationship between two 3D solid objects are very well-defined. These assumptions are too tight to make the method feasible to real cases, as bridges usually possess various curved horizontal and vertical alignments and cross-sections.

2.3 Other detection methods

Data-driven, learning-based methods have been widely applied to predict unknown instance labels based on training feature sets and manually added labels that facilitate supervised learning. For example, Xiong et al. (2013) propose a probabilistic graphical model to label the extracted planar surfaces for buildings. Zhang et al. (2014) use surface features to train a multi-class classifier, which assigns bridge component labels to surface primitives. However, this classifier is trained with generic Computer-Aided Design (CAD) shapes and tested on a synthetic

simplified bridge PCD. This method may not work with real data, because real bridge components contain skews and irregularities. Numerous volumetric Convolutional Neural Network and Deep Learning frameworks are proposed by transforming points into voxel grids (Qi et al., 2017) to detect objects in point clouds. The major restrictions to apply these machine learning schemes to bridge component detection task include: (1) the lack of sufficient amount of labelled large-scale real bridge PCD to train a good classifier, and (2) the high computing burdens (e.g. a typical sliding window method is computational expensive (Armeni et al. 2016)).

2.4 Gaps in knowledge, Objectives and Research questions

Most relevant works focus on clustering point clouds through generating surfaces, which are not robust with regard to occlusions and sparseness. Furthermore, it is difficult to transfer these methods to object detection in a real bridge PCD because bridges are neither straight nor flat. The only few bridge-related classification studies work well under tight constraints or on synthetic dataset. We therefore contend that the problem of generating bridge objects from real bridge point clouds has yet to be solved. The objective of this work is to develop a robust top-down object detection method to tackle with the challenges encountered in real bridge PCDs by answering: how to reliably segment a real bridge PCD with complex geometries and classify the major bridge components.

3. PROPOSED SOLUTION

This research focuses on slab and beam-slab highway bridges in the UK and we deal with the key components detection which are slab, pier, pier cap and girder. We illustrate the workflow of the proposed method in Fig. 1. Dashed frames refer to ambiguous components that may or may not exist in a bridge PCD.

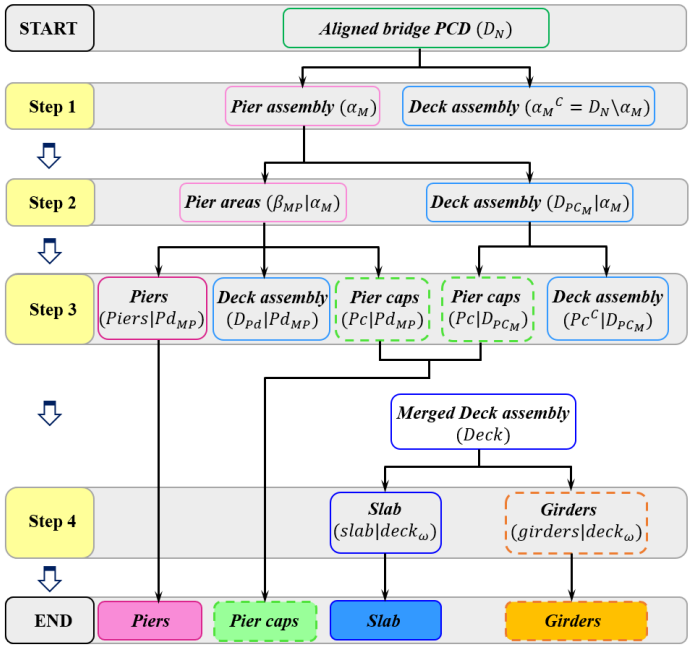


Fig. 1. Workflow of the proposed method

Step 1 – Pier assembly and deck assembly detection

We use Principal Component Analysis (PCA) to align a bridge such that the horizontal alignment of the bridge is positioned roughly parallel to the global X-axis. In this step, we aim to classify the whole bridge points into two groups: pier assembly group $\alpha_M = \{\alpha_1, \alpha_2 \dots, \alpha_m\}$, where m is the number of pier assembly and deck assembly group $\alpha_M{}^C$. We chop D_N into multiple slices along the X-axis (Fig. 2) and let J be the number of slices which is initialized proportionally to the length of the bridge. We obtain slices $S_X = \{S_{j\langle x\rangle} : j = 1,2, \dots, J\}$, where $\langle x \rangle$ refers to the axis of slicing. When the "virtual scalpel" encounters an empty or a single-point slice, it will infer the geometric feature from the nearest sound slice. Then, a 2D skeleton $sk_{j\langle x\rangle}$ is drawn for each slice $S_{j\langle x\rangle}$ using the mid-plane of its 3D axis-aligned bounding box (Fig. 2). We extract $range_{j\langle z\rangle}$ which is the height of $S_{j\langle x\rangle}$. We classify $S_{j\langle x\rangle}$ as a pier assembly slice if the Eq. (1) is satisfied; otherwise, it is a deck assembly slice:

$$range_{j\langle z\rangle} > \rho_1 |\max\{z_i|D_N\} - \min\{z_i|D_N\}| \quad (1)$$

where ρ_1 is a discrimination parameter that refers to the thickness ratio of the deck assembly relative to the height of the bridge. The adjacent slices with the same assembly property are merged into a cluster. Finally, we acquire pier assembly α_M and deck assembly ${\alpha_M}^C$.

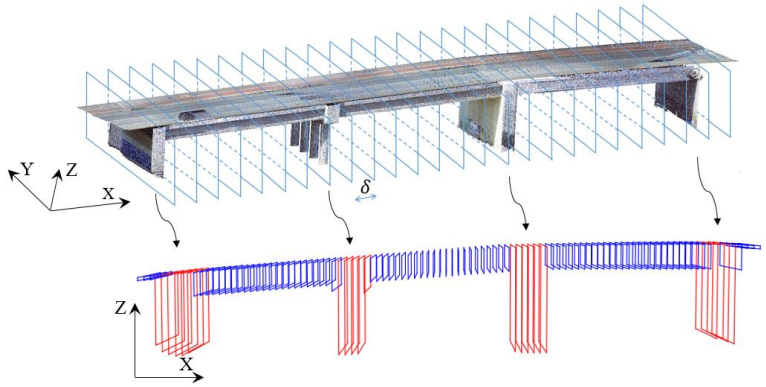


Fig. 2. Slicing along X-axis

Step 2 – Pier area detection in pier assembly

Each pier assembly α_m can be considered as a smaller scale of a bridge PCD so that Step 2 follows the same procedure as Step 1, except that the slicing is performed along the Y-axis of α_m. We classify $S_{j\langle y\rangle}$ as a pier area slice if the Eq. (2) is satisfied; otherwise, it is considered as a deck assembly slice:

$$range_{j\langle z\rangle} > \rho_2 |\max\{z_i|\alpha_m\} - \min\{z_i|\alpha_m\}| \quad (2)$$

where ρ_2 is another discrimination parameter that is used to separate the pier area β_{mp} from the rest in α_m.

Step 3 – Pier cap detection

We aim to detect pier caps using the surface normal in the upper part of the pier area.

Step 3.1 – Remove upper deck slab

We first remove the upper slab surface points from the pier area(s) $\{\beta_{mp}\}$ output from Step 2. The general transverse maximum gradient is defined to be 5% so that the lower bound of upper slab points is λ_{min}=5%$W_{\beta_{mp}}$, where $W_{\beta_{mp}}$ is the width of β_{mp} and the upper bound is $\lambda_{max} = \rho_1 H_{\beta_{mp}}$, where $H_{\beta_{mp}}$ is the height of β_{mp}. Define $\Delta\lambda$ to be the range where upper slab surface points are located. We should have $5\%W_{\beta_{mp}} < \Delta\lambda < \rho_{3a}H_{\beta_{mp}} < \rho_1 H_{\beta_{mp}}$, where ρ_{3a} is the slab thickness ratio estimation. The points in $\Delta\lambda$ are then removed and the remaining points in pier area(s) are denoted as $\{Pd_{mp}\}$.

Step 3.2 – Pier cap detection at top of piers

For each pier assembly α_m:

Scenario 1: a single pier area is detected in the pier assembly (i.e. $\|\beta_M\| = 1$) and the pier area extends almost the full width of the pier assembly (i.e. $W_{\beta_{mp}} \cong W_{\alpha_m}$). Then, it is a wall-type-pier pier assembly. A pier cap does not exist.

Scenario 2: for a pier assembly with capped pile pier or cap and column pier (i.e. $\|\beta_M\| > 1$) or, for a single detected pier area β_{mp}, but $W_{\beta_{mp}} \ll W_{\alpha_m}$, in these two cases, further pier cap detection is required.

We use the upper part of Pd_{mp} (i.e. denoted as $upper_{Pd_{mp}}$) to detect pier cap. We generate mesh for $upper_{Pd_{mp}}$ and compute the normal of each triangular surface. If a cluster of downward- or upward-oriented surface normals are revealed, and if these normals are found around the level $\rho_1(\max\{z_i|\beta_{mp}\} - \min\{z_i|\beta_{mp}\})$ (Fig. 3 red), then, we classify the feature points constituting these surfaces together with the points in $upper_{Pd_{mp}}$ that above the feature points as deck assembly. Otherwise, the pier cap feature points are detected if a cluster of downward- or upward-oriented normals are found around the level $\rho_2(\max\{z_i|\beta_{mp}\} - \min\{z_i|\beta_{mp}\})$ (Fig. 3 green).

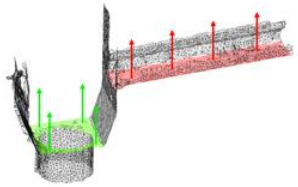
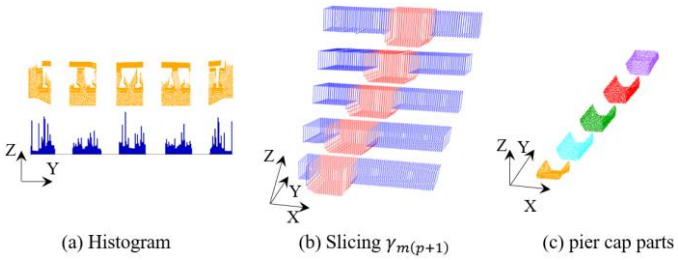


Fig. 3. Pier cap detection



(a) Histogram (b) Slicing $\gamma_{m(p+1)}$ (c) pier cap parts

Fig. 4. Extract pier cap from D_{PC_M}

Step 3.3 – Pier cap extraction from deck assembly

We extract the pier cap parts from D_{PC_M} if the pier caps are detected in Step 3.2. First, the points of D_{PC_m} are projected onto the YZ-plane followed by generating a density histogram along Y-axis. Then, the bins are clustered using the gaps between them (Fig. 4 (a)). We denote the segments as $\{\gamma_{m(p+1)}\}$. Then, we perform slicing along the X-axis of $\{\gamma_{m(p+1)}\}$. For $\gamma_{m(p+1)}$, the pier cap area is detected if $range_{j\langle z\rangle} > \rho_{3b}|\max\{z_i|\gamma_{m(p+1)}\} - \min\{z_i|\gamma_{m(p+1)}\}|$ (Fig. 4 (b)), where $\rho_{3b} = \frac{\rho_1}{\rho_2}$. Next, the procedure is similar to Step 3.1 and Step 3.2. The pier cap parts $\{Pc|D_{PC_M}\}$ are finally acquired (Fig. 4 (c)). In the end, we merge both pier cap parts output from Step 3.2 and Step 3.3.

Step 4 – Girder detection

Step 4.1 – Segment the whole deck assembly into several segments

To begin with, we conduct a merging process to build up a whole deck assembly cluster. This involves piecing up all point clusters classified as deck assembly in the previous steps.

For a beam-slab bridge, the length of the girder depends on the span. We need to split the whole merged deck assembly into several segments in order to find the appropriate length of span. The best cutting planes depend on the orientation of the expansion joints. This is because two adjacent deck assembly segments must be interconnected by the expansion joints. Pier clusters and pier caps are then oriented based on the joints. We employ 3D oriented bounding box of a pier cluster to capture the orientation, through which, the entire deck assembly is segmented into multiple segments $\{deck_\omega\}$ (Fig. 5).

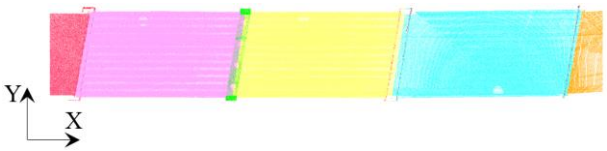


Fig 5. Deck segments $\{deck_\omega\}$

Step 4.2 – Girder detection in the deck assembly segment

We detect girders in each deck assembly segment. We start by rotating $deck_\omega$ around its Y-axis until $deck_\omega$ reaches the best projection view. Rotation is conducted through a grid search in a range of angles $\{\xi\}$= [-3°, 3°]. A density histogram $\mathcal{H}_Z$ along the Z-axis is employed for evaluating if a best rotation is reached. Specifically, the best rotation angle is acquired when the maximum standard deviation of $\mathcal{H}_Z$ for $deck_\omega$ returns.

Next, we used the bottom ρ_4 (%) points of $deck_{\omega(\xi)}$ (denoted as $b_{deck_{\omega(\xi)}}$) for girder detection, where $\rho_4 = \frac{\rho_1 - \rho_{3a}}{\rho_1}$. A density histogram $\mathcal{H}_Y$ is drawn along Y-axis of $b_{deck_{\omega(\xi)}}$. The density probability is uniformly distributed with significantly lower variance when there is no girder (i.e. slab bridge) while significant peaks can be observed in the distribution with non-trivial variance when girders exist (i.e. beam-slab bridge). We segment the girders if they exist and the girder section type (e.g. Y, U or SY beams) can be inferred from the bottom flange width and the girder depth along the best project view. The 4-step top-down recursive detection method terminates. All the over-segments from Step 1 to Step 4 are merged as per their class labels.

4. RESEARCH METHODOLOGY

4.1 Data & Methods

We used a FARO Focus 3D X330 laser scanner to collect point cloud data of 10 highway bridges around Cambridgeshire, UK. Our analyses consist of two parts. The first part is to experimentally define the optimal values of the two hyper-parameters (ρ_1 and ρ_2) at the level of individual point clusters in Steps 1 and 2 respectively. Then, we derived the optimal values of the other three hyper-parameters (ρ_{3a}, ρ_{3b} and ρ_4). The second part is to assess the optimized proposed method on the level of bridge structural components using both bounding box-wise and point-wise performance metrics.

4.2 Estimation of hyper-parameters

We developed a user-defined bounding box functionality to manually remove the irrelevant points, such as on-site traffic, vegetation and trees. After down sampling, we aligned the cropped bridge PCD using PCA. Then, we estimated the two hyper-parameters parameters ρ_1 and ρ_2. We grid searched various values of ρ_1 and ρ_2 over the value space (0, 1) and identified the optimal ρ_1^* and ρ_2^* through computing the empirical receiver operating characteristic. Denote "S" as a specific point cluster, where $S \in \{\alpha_M, {\alpha_M}^C\}$ in Step 1 and $S \in \{D_{PC_M}, \beta_{MP}\}$ in Step 2. We defined the following point-wise performance metrics Precision (Pr), Recall (R) and F1-score (F1) as:

$$\mathrm{Pr_s} = \frac{\mathrm{TP_s}}{\mathrm{TP_s} + \mathrm{FP_s}} = \frac{\#\text{ of correctly labeled points in cluster s}}{\text{total \# of points in cluster s}} \tag{4}$$

$$\mathrm{R_s} = \frac{\mathrm{TP_s}}{\mathrm{TP_s} + \mathrm{FN_s}} = \frac{\#\text{ of correctly labeled points in cluster s}}{\text{total \# of points in ground truth cluster s}} \tag{5}$$

$$\mathrm{F1_s} = 2 * \frac{\mathrm{Pr_s} * \mathrm{R_s}}{\mathrm{Pr_s} + \mathrm{R_s}} \tag{6}$$

Then, ρ_{3b} can be computed based on ρ_1^* and ρ_2^*. The value of ρ_{3a} is derived using probability of point density for $\{\beta_{mp}\}$ so that ρ_4 is also acquired.

4.3 System Validation and Results

The detection results were shown in Fig. 7. The method generated a bounding box for each segmented point cluster (AutoBBox) and assign a semantic instance label to each point. They are compared with the manually segmented point clusters in their bounding boxes (GTBBox) and the cluster points in their instance labels.

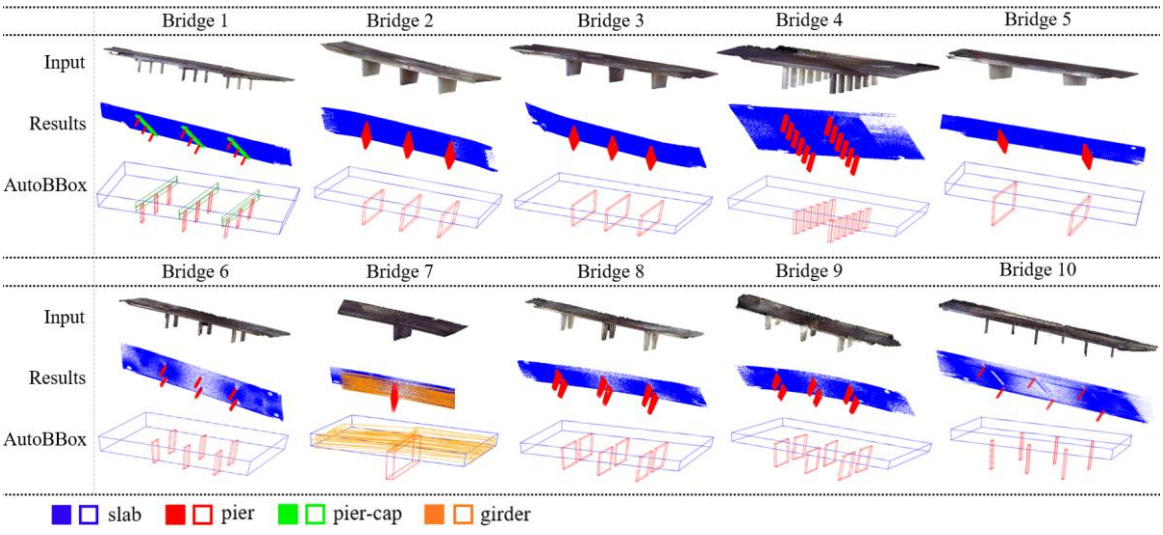


Fig. 7. Detection results and AutoBBoxes for point clusters

For a specific point cluster generated from the solution, let C_{auto} and C_{gt} be the centers of its AutoBBox and its GTBBox (if it exists), respectively, and $d(C_{auto}, C_{gt})$ be the Euclidean distance between C_{auto} and C_{gt}.
C1. GTBBox of the specific point cluster exists;

C2. C_{auto} is inside the corresponding GTBBox;

C3. $\varepsilon = \frac{d(C_{auto}, C_{gt})}{min(l_{gt}, w_{gt}, h_{gt})}$ <50%, where l_{gt}, w_{gt}, h_{gt} are the length, width and height of the GTBBox of the point cluster, respectively.

The point cluster is correctly detected by the AutoBBox and we assigned one to True Positive (TP) if all the above three conditions are satisfied; one to False Positive (FP) if C1 is false but an AutoBBox is generated; one to False Negative (FN) if C1 is true but at least one of C2 and C3 is not satisfied. The precision, recall and F1-score were generated using the values of TP, FN and FP (Eq. (7) – Eq. (9)). Table 1 summarizes the results of bounding-box-wise performance evaluation.

$$\mathrm{Pr}_{\mathrm{bridge\_j}} = \frac{\#\text{ of correctly detected point clusters}}{\text{total \# of AutoBBoxes for a bridge PCD}} \tag{7}$$

$$\mathrm{R}_{\mathrm{bridge\_j}} = \frac{\#\text{ of correctly detected point clusters}}{\text{total \# of GTBBoxes for a bridge PCD}} \tag{8}$$

$$\mathrm{F1}_{\mathrm{bridge\_j}} = 2 * \frac{\mathrm{Pr}_{\mathrm{bridge\_j}} * \mathrm{R}_{\mathrm{bridge\_j}}}{\mathrm{Pr}_{\mathrm{bridge\_j}} + \mathrm{R}_{\mathrm{bridge\_j}}} \tag{9}$$

Table 1. Bounding-box-wise component detection performance

Bridge	1	2	3	4	5	6	7	8	9	10	**Avg**
FN	0	0	0	0	0	0	0	0	1	0	
FP	0	0	0	0	0	0	0	0	0	0	
TP	13	4	4	13	3	7	20	7	6	7	
Precision	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	**100%**
Recall	100%	100%	100%	100%	100%	100%	100%	100%	85.7%	100%	**98.6%**
F1-score	100%	100%	100%	100%	100%	100%	100%	100%	92.3%	100%	**99.2%**

We reconducted the system evaluation with point-wise metrics, i.e. Eq. (4) – Eq. (6). Herein, the "S" refers to any specific final point cluster generated from our proposed solution. For a specific bridge PCD, we computed the micro-average and the macro-average scores. In micro-average, we computed the statistics:

$$\mathrm{Pr}_{\mathrm{micro}} = \frac{\sum_{s=1}^{|S|} \mathrm{TP_s}}{\sum_{s=1}^{|S|} \mathrm{TP_s} + \sum_{s=1}^{|S|} \mathrm{FP_s}} \tag{10}$$

$$\mathrm{R}_{\mathrm{micro}} = \frac{\sum_{s=1}^{|S|} \mathrm{TP_s}}{\sum_{s=1}^{|S|} \mathrm{TP_s} + \sum_{s=1}^{|S|} \mathrm{FN_s}} \tag{11}$$

where |s| is the number of generated point clusters in this given bridge PCD. The micro-average F1-score is simply the harmonic mean of $\mathrm{Pr}_{\mathrm{micro}}$ and $\mathrm{R}_{\mathrm{micro}}$. In macro-average, we took the average of the Precision and Recall of all point clusters (Eq. (12) and Eq. (13)). Likewise, the macro-average F1-score is the harmonic mean of $\mathrm{Pr}_{\mathrm{macro}}$ and $\mathrm{R}_{\mathrm{macro}}$. We summarized the point-wise the micro-average and the macro-average evaluation results in Table 2.

$$\mathrm{Pr}_{\mathrm{macro}} = \frac{\sum_{s=1}^{|S|} \mathrm{Pr_s}}{|S|} \tag{12}$$

$$\mathrm{R}_{\mathrm{macro}} = \frac{\sum_{s=1}^{|S|} \mathrm{R_s}}{|S|} \tag{13}$$

5. CONCLUSIONS

Our proposed method achieved remarkable performance: the micro-average of Pr/R/F1 are all 98.8% (for multi-class case, the micro-average option yields result in a mathematically equivalent definition for Pr and R, thus equivalent F1); the macro-average ones are 99.2%, 98.3% and 98.7%, respectively. However, small girder spacing decreases the detection performance (i.e. Bridge 7). This performance degradation is mainly due to a limited line of sight to the surface points between those girders. The results indicated typical highway bridges can be supported

using the proposed solution, which can significantly reduce the modeling cost of BrIM.

Table 2. Point-wise performance evaluation results in micro- and macro-average

Bridge ID	TP#	FP#	$Pr/R/F1_{micro}$	Pr_{macro}	R_{macro}	$F1_{macro}$
1	486010	2443	99.5%	98.5%	99.3%	98.9%
2	499155	845	99.8%	99.9%	99.4%	99.7%
3	498713	1287	99.7%	99.9%	98.9%	99.4%
4	498356	223	100%	99.9%	99.9%	99.9%
5	498950	1050	99.8%	99.9%	99.2%	99.5%
6	499435	565	99.9%	100%	99.4%	99.7%
7	405279	49706	89.1%	94.4%	88.9%	90.8%
8	499338	662	99.9%	100%	99.6%	99.8%
9	499470	530	99.9%	100%	99.7%	99.8%
10	874806	230	100%	100%	99.1%	99.5%
Avg			**98.8%**	**99.2%**	**98.3%**	**98.7%**

ACKNOWLEDGMENTS

This material is based in part upon work supported by EPSRC and the Infravation SeeBridge project under Grant Number No. 31109806.0007. The authors would like to thank their supports.

REFERENCES

Ahmed, M. F., Haas, C. T., & Haas, R. (2014). Automatic Detection of Cylindrical Objects in Built Facilities. Journal of Computing in Civil Engineering, 28(3), 1–11.

Armeni, I., Sener, O., Zamir, A., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. (2016). 3D Semantic Parsing of Large-Scale Indoor Spaces. *CVPR*, 1534–1543.

Belsky, M., Sacks, C.V. and Brilakis, I. (2014). A Semantic Enrichment Engine for Building Information Modelling. Journal of Computer-Aided Civil and Infrastructure Engineering.

Dimitrov, A., & Golparvar-Fard, M. (2015). Segmentation of building point cloud models including detailed architectural/structural features and MEP systems. Automation in Construction, 51(C), 32–45.

Dimitrov, A., Gu, R., & Golparvar-Fard, M. (2016). Non-Uniform B-Spline Surface Fitting from Unordered 3D Point Clouds for As-Built Modeling. Computer-Aided Civil and Infrastructure Engineering, 31(7), 483–498.

FHWA. (2013). Federal Highway Administration - 2013 Status of the Nation's Highways, Bridges, and Transit: Conditions & Performance.

Koppula, H. S., Anand, A., Joachims, T., & Saxena, A. (2011). Semantic Labeling of 3D Point Clouds for Indoor Scenes. Neural Information Processing Systems, 1–9.

Lu, R., & Brilakis, I. (2017). Recursive Segmentation for as-is Bridge Information Modelling. In Lean & Computing in Construction Congress (LC3). Heraklion, Greece.

Ma, L., Sacks, R., Kattel, U., & Bloch, T. (2017). 3D Object Classification Using Geometric Features and Pairwise Relationships. Computer-Aided Civil and Infrastructure Engineering, 0, 1–13.

McGraw Hill Construction. (2014). The Business Value of BIM for Construction in Major Global Markets: How contractors around the world are driving innovation with Building Information Modeling.

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In CVPR 2017. Honolulu, Hawaii.

Riveiro, B., DeJong, M. J., & Conde, B. (2016). Automated processing of large point clouds for structural health monitoring of masonry arch bridges. Automation in Construction.

Schnabel, R., Wahl, R., & Klein, R. (2007). Efficient RANSAC for point-cloud shape detection. Computer Graphics Forum, 26(2), 214–226.

Walsh, S. B., Borello, D. J., Guldur, B., & Hajjar, J. F. (2013). Data processing of point clouds for object detection for structural engineering applications. Computer-Aided Civil and Infrastructure Engineering, 28(7), 495–508.

Xiong, X., Adan, A., Akinci, B., & Huber, D. (2013). Automatic creation of semantically rich 3D building models from laser scanner data. Automation in Construction, 31, 325–337.

Zhang, G., Vela, P.A. and Brilakis, I. (2014). Automatic Generation of as-built Geometric Civil Infrastructure Models from Point Cloud Data. In Proceedings of the 15th ICCCBE joint with the 2014 CIB W78 Conference on IT in Construction (pp. 406–413). 23-25 June 2014, Orlando, FL.

Zhang, G., Vela, P. A., Karasev, P., & Brilakis, I. (2015). A sparsity-inducing optimization-based algorithm for planar patches extraction from noisy point-cloud data. Computer-Aided Civil and Infrastructure Engineering, 30(2), 85–102.