COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING

Computer-Aided Civil and Infrastructure Engineering **0** (2018) 1–22

Detection of Structural Components in Point Clouds of Existing RC Bridges

Ruodan Lu, Ioannis Brilakis* & Campbell R. Middleton

Laing O'Rourke Center, Department of Engineering, University of Cambridge, Cambridge, UK

Abstract: *The cost and effort of modeling existing bridges from point clouds currently outweighs the perceived benefits of the resulting model. There is a pressing need to automate this process. Previous research has achieved the automatic generation of surface primitives combined with rule-based classification to create labeled cuboids and cylinders from point clouds. Although these methods work well in synthetic data sets or idealized cases, they encounter huge challenges when dealing with real-world bridge point clouds, which are often unevenly distributed and suffer from occlusions. In addition, real bridge geometries are complicated. In this article, we propose a novel top-down method to tackle these challenges for detecting slab, pier, pier cap, and girder components in reinforced concrete bridges. This method uses a slicing algorithm to separate the deck assembly from pier assemblies. It then detects and segments pier caps using their surface normal, and girders using oriented bounding boxes and density histograms. Finally, our method merges oversegments into individually labeled point clusters. The results of 10 real-world bridge point cloud experiments indicate that our method achieves very high detection performance. This is the first method of its kind to achieve robust detection performance for the four component types in reinforced concrete bridges and to directly produce labeled point clusters. Our work provides a solid foundation for future work in generating rich Industry Foundation Classes models from the labeled point clusters.*

1 INTRODUCTION

The global infrastructure market is poised for an explosive adoption of bridge information modeling (BrIM), which provides a shared knowledge resource for information exchange to support a reliable basis for decision making during a bridge's life cycle (Fanning et al., 2014). The adoption of BrIM in the United States and the United Kingdom has increased by 30% from 2015 to 2017 (Dodge Data & Analytics, 2017). However, the produced models are mainly as-designed models of new structures. The generation of as-is BrIM models for existing bridges is very limited, despite the widespread adoption of laser-scanning for faster and better data collection (Park et al., 2007; Park et al., 2015). This is because the automatic generation of as-is models from point cloud data (PCD) remains an unsolved problem. The time required to manually create an as-is three-dimensional (3D) solid model in a point cloud using cutting edge modeling software typically is of the order of 10 times greater than that required to obtain the point cloud (Trimble, 2014; Lu and Brilakis, 2017).

There is a pressing need to lower the cost and effort required for modeling existing bridges. This is particularly true for the highway infrastructure sector. There are more than 600,000 highway bridges in the United States and more than one in nine is classified as structurally deficient (ASCE, 2017). According to an in-house report (Network Rail, 2015), Network Rail and other bridge owners manage more than 30,000 bridges on the United Kingdom's motorways and major A-roads. Based on a 2-year inspection cycle, there is a need for at least 315,000 bridge inspections per annum across the United States and the United Kingdom. This explains why there is a huge market demand for less labor-intensive bridge documentation techniques that can efficiently boost bridge management productivity.

In general, the *from-PCD-to-BrIM* model process consists of two steps: (1) detecting bridge components in point clouds in the form of labeled point clusters and (2)

*To whom correspondence should be addressed. E-mail: *ib340@cam.ac.uk*.

© 2018 *Computer-Aided Civil and Infrastructure Engineering*.
DOI: 10.1111/mice.12407

generating a geometric model by fitting Industry Foundation Classes (IFC) entities and spatial relationships in labeled point clusters. This study intends to automate Step 1, that is, bridge-component detection in point clouds. This step is currently largely achieved manually using modeling software.

Major vendors such as Autodesk, Bentley, and ClearEdge3D provide the most advanced software solutions for Building Information Modeling. ClearEdge3D can semiautomatically (a few clicks and adjustments are required) fit geometric shapes embedded in the subsets of a point cloud without segmenting beforehand. However, ClearEdge3D is tailored for building and industrial environments and only standardized shapes, such as rectangular walls, pipes, steel beams, and so forth, can be recognized (ClearEdge3D, 2017). For most of the other commercial applications, the shape fitting is largely assisted by manually segmenting the point cloud in advance. These solutions demand a significant amount of attention when segmenting the target objects. Modelers need to repeatedly rotate the point cloud to various views and try to select regions of interest using clipping polygons. Lu and Brilakis (2017) report that, on average, 1.52 hours is needed to complete just the bridge-component detection task for processing a typical reinforced concrete (RC) highway-bridge point cloud.

In this article, we propose a novel top-down method for the abovementioned object detection problem. The novelty of this method lies in the fact that it directly extracts the key components of RC bridges without generating low-level shape primitives.

We discuss the current state of research in Section 2 and we outline the proposed method in Section 3. We then elaborate on the research methodology and experimental results in Section 4. Finally, we conclude in Section 5.

2 BACKGROUND

Existing software does not provide a fully automatic solution for concrete object detection in point clouds. Much effort has been devoted to automating this process. We define "detection" in this context as the combination of clustering (point cloud-to-point clusters) and classification (labeling the clusters). Current methods of point cloud clustering generally follow a "bottom-up" approach, which goes from points to surfaces or patches followed by semantic labeling to derive objects. Most point cloud classification methods follow a "top-down" approach, which employs human visual perception such as relationships, contexts, and so forth to detect specific instances embedded in a point cloud or to 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 complex systems in successively higher levels until a top-level system is formed (Borenstein and Ullman, 2008). The higher level features are typically the surface normal (Sampath, 2010), meshes (Marton et al., 2009), patches (Zhang et al., 2015), and nonuniform B-Spline surfaces (Dimitrov et al., 2016).

A large body of literature has been devoted to generating surface-based primitives, especially planar surfaces (Pătrăucean et al., 2015). Zhang et al. (2015) present a sparsity-inducing optimization-based method to detect parametric planar patches and define their boundaries from noisy bridge point clouds. However, this method can work only with planar-surface objects and cannot detect pier patches when point densities of these regions are low. Walsh et al. (2013) present a region growing (RG) algorithm to detect objects in a point cloud. RG starts with an initial set of small iteratively merged areas, followed by choosing a specific seed and adding in neighboring points based on similarity until an edge is reached. However, this method cannot detect the edge between a pier cap and a pier in a small portion of a bridge point cloud as shown in their experiment. The segmentation was finally achieved after manually choosing key points. Likewise, Dimitrov and Golparvar-Fard (2015) suggest an upgraded RG method through which the seed is found adaptively. This method can deal with curved surfaces and excels when the input point cloud does not suffer from substantive occlusions. However, it oversegments objects when occlusions are present. The persistent occlusion problem in real point clouds was addressed by Xiong et al. (2013) through a learning-paradigm that detects occluded planar surfaces in building point clouds. However, their method cannot be applied in bridge settings because the occluded surfaces in a bridge point cloud do not follow a specific pattern as in a building point cloud. Specifically, their algorithm detects rectangularly shaped openings, such as windows and doorways, assuming there are many identical openings on a wall and the rectangle is the predominant shape in most buildings. Similarly, Laefer and Truong-Hong (2017) develop a kernel density estimation–based method for modeling steel members by simulating several possible occlusions. In contrast, the occluded regions do not have such a repeated pattern in bridge point clouds. Most of the occlusions are due to on-site vegetation and long-distance scanning. Thus, the occlusions are in arbitrary locations and shapes. Schnabel et al. (2007) detect basic shapes (e.g., spheres, cylinders) using Random Sample Consensus (RANSAC) by random sampling of minimal sets in a point cloud. Yet, given the computationally 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 with synthetic data, but are not ready to tackle the real bridge components whose as-constructed and as-weathered shapes further increase the as-designed complexity. To reduce computational time, Xu et al. (2018) suggest an octree-based probabilistic segmentation model for construction sites. The authors partitioned the scene into voxels. However, the segmentation accuracy of this method is quite sensitive to the voxel size. This problem is discussed by Vo et al. (2015) who propose an octree RG-based algorithm for surface patch segmentation in urban environments. Although their method can automatically adjust the voxel size through an adaptive octree, it faces the difficulty of patch generation for low point density regions.

2.2 Top-down detection

We contend that bottom-up detection is rarely suitable for point cloud classification. Classification through surfaces is insufficient; local surfaces or patches can be labeled as such, but it is difficult to determine whether they belong to the same instance. The intervention of object-level information is required to overcome such challenges (Pinheiro et al., 2016). The top-down approach is typically a heuristic approach for object detection, which begins with a broad-picture view, and then is broken into compositional sub-problems that are easier to solve (Kokkinos et al., 2006). It usually combines a set of engineering criteria and classifies objects in the point cloud that meet the criteria. Prior studies show that knowledge-based classification methods are robust, as domain-specific information such as object classes (Dore and Murphy, 2014), topological relationships (Koppula et al., 2011), and known parameters (e.g., diameter) or constraints (e.g., direction) (Ahmed et al., 2014) are invariant to factors such as pose and appearance. Recent research relies on existing as-designed documents to inform the top-down modeling strategy. This can simplify point cloud clustering and classification tasks (Liu et al., 2012). Likewise, Belsky et al. (2016) encapsulate domain expert knowledge in the form of rule sets to enrich semantics for a building model.

However, the methods developed in these studies are tailored for buildings and indoor and industrial objects, and not tailored for use in bridge settings, as the geometric properties of bridge components are quite different than those objects. What is more, there are few as-built or as-is models for existing bridges. Recently, some studies have started to employ top-down strategies to detect bridge components in point clouds. Riveiro et al. (2016) use specific constraints to segment masonry bridge point clouds into surfaces. However, their algorithm is based on histograms that largely depend on data quality. It is difficult to generalize this algorithm to large RC bridges, as the real point clouds usually suffer from occlusions and nonuniformly distributed points. Ma et al. (2017) leverage relationship knowledge and shape features to classify bridge 3D solid objects. First, the input of this method needs to be a solid bridge model (i.e., not a bridge point cloud). Second, the method assumes that the 3D solid model is developed in a grid system, in ideal geometries and that the pairwise relationship between two 3D solid objects is well defined. These assumptions are quite restrictive and make the method less feasible in 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. 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 multiclass classifier, which assigns bridge-component labels to surface primitives. As in the previous cases, this method is designed for simplified bridge designs that do not contain skews, irregularities, and complicated objects, which are often the cases of real bridge components.

Numerous volumetric convolutional neural network and deep learning frameworks are proposed by transforming 3D points into voxel grids (Qi et al., 2017; Tatarchenko et al., 2017) for object detection. The major restrictions to apply these data-driven machine learning schemes to bridge-component detection tasks include (1) the lack of a sufficient number of labeled large-scale real bridge point clouds to train a good classifier and (2) the high computational burden.

2.4 Gaps in knowledge

BrIM models are mainly represented using volumetric primitives (Tang et al., 2010), whereas most existing works focus on clustering point clouds through generating surfaces. These methods are not robust with regard to occlusions and sparseness and are difficult to transfer to the problem of object detection in a real bridge point cloud because bridge components are often skewed. In addition, none of the existing bottom-up methods can directly output labeled bridge point clusters. The few bridge-related studies that work well have restrictive constraints. We therefore contend that the problem of detecting bridge objects in the form of labeled point clusters from real bridge point clouds has yet to be solved.

2.5 Objectives and research questions

We aim to (1) automatically segment an RC bridge point cloud into mutually disjoint point clusters corresponding to the components making up the bridge and (2) automatically assign correct semantic labels to these point clusters. This will be done by answering the following research questions: (1) how to effectively segment a real RC bridge point cloud and (2) how to efficiently and robustly classify major RC bridge-component point clusters.

2.6 Hypothesis

A simple bottom-up method that generates low-level primitives has not, to date, managed to solve the above-mentioned research questions. In contrast, human expertise can facilitate the search for a specific object in a scene because our guesses for the embedded objects are best when we know what to expect in the point cloud. Hence, we will investigate a brand-new top-down method which directly extracts key components of RC bridges without generating low-level shape primitives.

The hypothesis of this work is that the top-down bridge-component detection approach is efficient and reliable and there is no significant difference in detection performance for different RC bridges. We elaborate this hypothesis in the following.

3 PROPOSED SOLUTION

3.1 Scope

This research only focuses on typical RC slab and beam-slab bridges on motorways and major A-roads in the United Kingdom, because 73% of the existing highway bridges and 86% of the planned future bridges in England are RC slab and beam-slab bridges (Kim et al., 2016). These two types of bridges reflect the fundamental bridge-design rules that allow for natural introduction of geometric constraints, describing the relationships that should hold between various components of a bridge. The key components of RC slab bridges include slabs, piers, and pier caps, and for RC beam-slab bridges, there are slabs, girders, piers, and pier caps (Kim et al., 2016). Thus, this work deals with four very important and highly detectable structural components of a highway bridge, that is, slabs, piers, pier caps, and girders, in line with Kedar's (2016) first element identification evaluation category.

3.2 Overview

The general thrust behind our top-down approach is to use the fundamental bridge design rules such as bridge topological constraints, given the low level of variance of the topological layout of RC slab and beam-slab bridges.

Our method bypasses the stage of surface generation altogether and directly obtains segmented and labeled point clusters. It breaks down a large bridge point cloud into sub-data sets through a recursive slicing algorithm. That is, the method chops the point cloud by means of a "virtual parallel scalpel" with a specified equal thickness. This algorithm is repeatedly used with sub-data sets until target objects are found and all small detection problems are solved. The key insight behind this method is to formulate the geometric feature search to explore shortcuts so that the target components can be quickly located in the point cloud.

The workflow of the proposed method is illustrated in Figure 1. Dashed frames refer to ambiguous components that may or may not exist in a bridge point cloud. Acronyms are used to present inputs, intermediate outputs, and final outputs of each step. For example, {B|A} represents B is a subset of A. A is derived from a previous step and is the superset of B. {B|A} may also represent B is a property set of A (e.g., $\{x_i|\mathrm{A}\}$ means the x values of A). More precisely, "deck assembly" refers to the areas which contain slab and girders (if they exist), "pier assembly" refers to the areas which contain a transverse strip of slab, pier caps (if they exist) and piers and "pier areas" refers to the subsets of pier assembly which contain a small part of the slab strip, part of the pier cap and individual pier.

The first two steps in our detection method are recursive. The first step segments a whole aligned bridge point cloud (i.e., D_N) into pier assembly (denoted α_M) and deck assembly (denoted ${\alpha_M}^C$). The second and third detect pier areas (denoted β_{mp}) and pier caps (denoted P_C) in pier assembly and deck assembly. The last step detects girders (denoted *girders*) and slab (denoted *slab*) in a merged deck assembly. Note that pier caps and girders may not exist in some bridge point clouds.

3.3 Step 1—Pier assembly and deck assembly detection

A bridge point cloud is given at an arbitrary position and orientation. The pose of a bridge should be

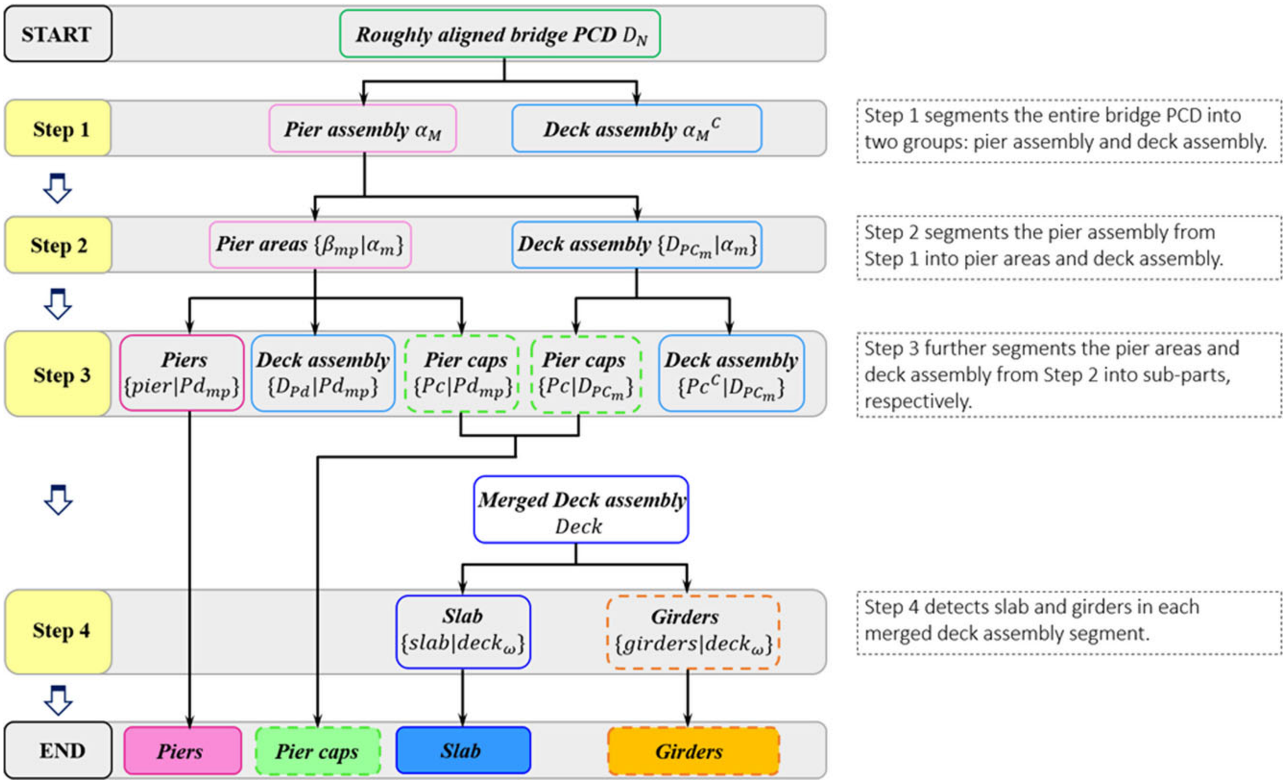

START
Roughly aligned bridge PCD D_N
Step 1
Pier assembly α_M
Deck assembly $\alpha_M{}^C$
Step 1 segments the entire bridge PCD into two groups: pier assembly and deck assembly.
Step 2
Pier areas $\{\beta_{mp}|\alpha_m\}$
Deck assembly $\{D_{PC_m}|\alpha_m\}$
Step 2 segments the pier assembly from Step 1 into pier areas and deck assembly.
Step 3
Piers $\{pier|Pd_{mp}\}$
Deck assembly $\{D_{Pd}|Pd_{mp}\}$
Pier caps $\{Pc|Pd_{mp}\}$
Pier caps $\{Pc|D_{PC_m}\}$
Deck assembly $\{Pc^C|D_{PC_m}\}$
Step 3 further segments the pier areas and deck assembly from Step 2 into sub-parts, respectively.
Merged Deck assembly
Deck
Step 4
Slab $\{slab|deck_\omega\}$
Girders $\{girders|deck_\omega\}$
Step 4 detects slab and girders in each merged deck assembly segment.
END
Piers
Pier caps
Slab
Girders

Fig. 1. Workflow of the proposed method.

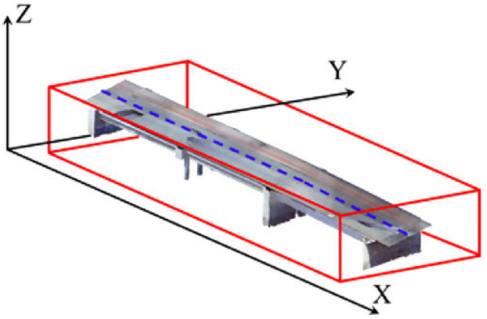


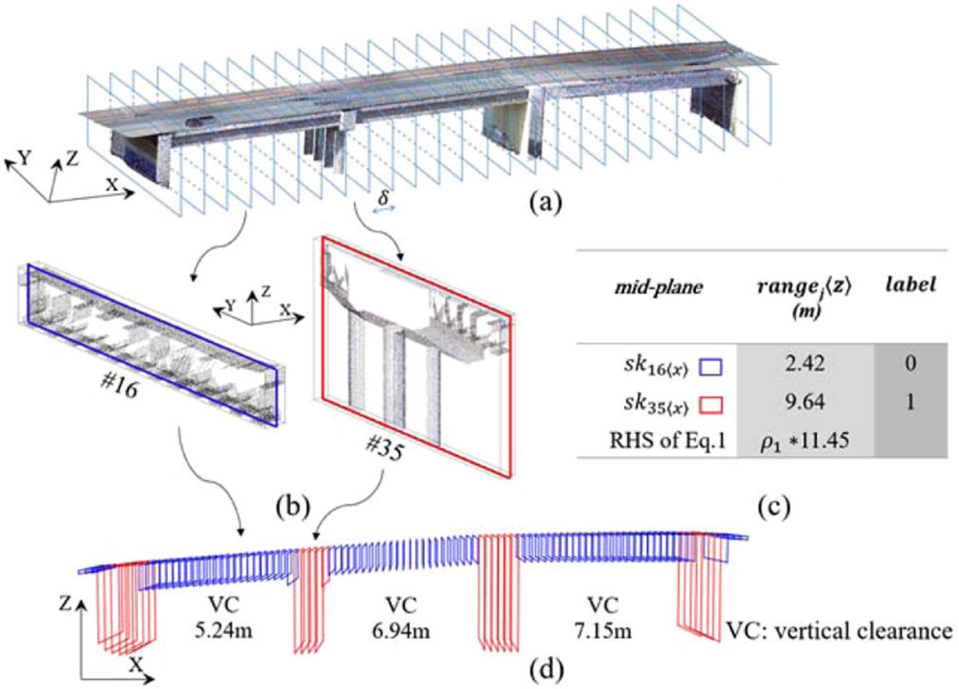
Fig. 2. Aligned bridge point cloud.

normalized in advance as all features extracted in further steps are in a canonical coordinate frame. 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 (Figure 2). Approximate alignment at an early stage makes it possible to reformulate features employed for recursive segmentation in the following steps. Therefore, the input of Step 1 is a roughly aligned bridge point cloud $D_N = \{p_i : i = 1, 2, \dots, N\}$, where each point is defined as $p_i = (x_i, y_i, z_i)^T$.

We aim to classify all the bridge points into two groups: pier assembly group $\alpha_M = \{\alpha_1, \alpha_2 \dots, \alpha_m\}$, where m is the number of the pier assembly and deck assembly group $\alpha_M{}^C$.

We chop D_N into multiple slices along the X-axis (Figure 3a). Let J be the number of slices, then we obtain slices $S_X = \{S_{jx} : j = 1, 2, \dots, J\}$, where x refers to the axis of slicing. Define $D_j = \{p_{ji}\} = \{p_i | S_{jx}\}$ to be the point set in slice S_{jx} so that we must have $\sum_{j=1}^{J} \sum_{i=1}^{D_j} p_{ji} = D_N$, where p_{ji} is the ith point in the jth slice. Note that slicing might lead to an empty slice where local points are missing or a slice with just one single point. In this case, the geometric features cannot be computed. Our slicing method can prevent such situations from happening. By assuming the slice thickness δ is constant, the initialized number of slices J is proportional to the length of the bridge (i.e., $J \propto |\max\{x_i|D_N\} - \min\{x_i|D_N\}|$). Here, δ is set to be 0.5 m. When the "virtual scalpel" encounters an empty or a single-point slice, the method will infer the geometric feature from the nearest "sound" slice: if $|D_j| = \emptyset$ or $|D_j| = 1$, then $S_{jx} \cong S_{j-\varphi x}$, where φ is the count of slices between S_{jx} and the closest nonempty and non–single-point slice. This approximation is not perfect but provides immunity to locally incomplete data.

Then, we need a feature detector that can distinguish the pier assembly from the deck assembly.



mid-plane	*$range_j(z)$ (m)*	*label*
$sk_{16(x)}$ □	2.42	0
$sk_{35(x)}$ □	9.64	1
RHS of Eq.1	$\rho_1 * 11.45$	

Fig. 3. (a) Slicing along X-axis; (b) deck assembly slice (blue) and pier assembly slice (red); (c) comparison between a deck assembly slice and a pier assembly slice; (d) midplanes $\{sk_{jx}\}$.

Each slice S_{jx} is bounded by a 3D axis-aligned-bounding-box and a 2D skeleton sk_{jx} is drawn for each slice using the midplane of its bounding-box (Figure 3b). According to bridge engineering knowledge, piers support the deck against gravity, so they should start from the ground. Therefore, the height of a pier assembly slice should obviously be much larger than that of a deck assembly slice (Figure 3b). Thus, in each slice S_{jx}, we extract the geometric feature $range_{jz}$ which is the height of S_{jx}. We classify S_{jx} as a pier assembly slice if Equation (1) is satisfied; otherwise, S_{jx} is considered a deck assembly slice:

$$\text{if} \quad range_{jz} > \rho_1 \left|\max\{z_i | D_N\} - \min\{z_i | D_N\}\right| \qquad (1)$$

then,

$$S_{jx} \leftarrow \text{pier assembly}$$

where ρ_1 is a discrimination parameter that refers to the thickness ratio of the deck assembly relative to the height of the bridge, which should not be affected by the varying elevation (Figure 3c). This assumption (see Section 4.1, A1) will be experimentally justified in Section 4.3.2. The adjacent slices with the same assembly property are merged into a cluster (Figure 3c). Finally, pier assembly α_M and deck assembly $\alpha_M{}^C$ (Figure 4a) are acquired.

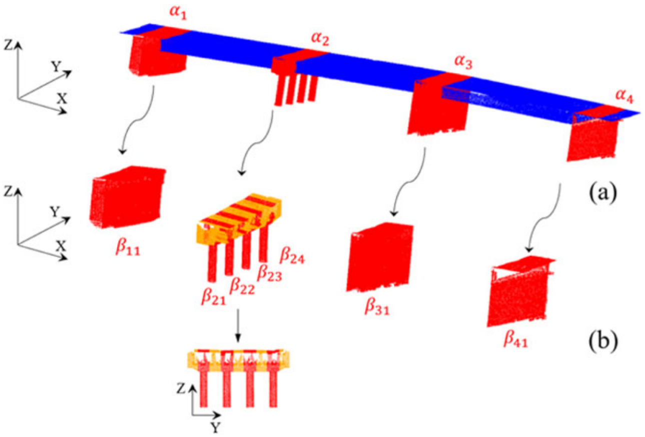


Fig. 4. (a) Deck assembly $\alpha_M{}^C$ (blue) and pier assemblies α_M (red); (b) deck assembly D_{PC_M} (orange) and pier areas β_{MP} (red).

3.4 Step 2—Pier area detection in pier assembly

The inputs of Step 2 are the pier assemblies α_M output from Step 1. The outputs are deck assemblies $D_{PC_M} = \{D_{PC_m} | \alpha_m\}$ and pier area(s) $\beta_{MP} = \{\beta_{mp} | \alpha_m\}$, where $\alpha_M = D_{PC_M} \cup \beta_{MP}$ (Figure 4). Each pier assembly α_m can be considered a smaller scale of a bridge point cloud so that Step 2 follows the same procedure as Step 1, except that the slicing is performed along the Y-axis of α_m to

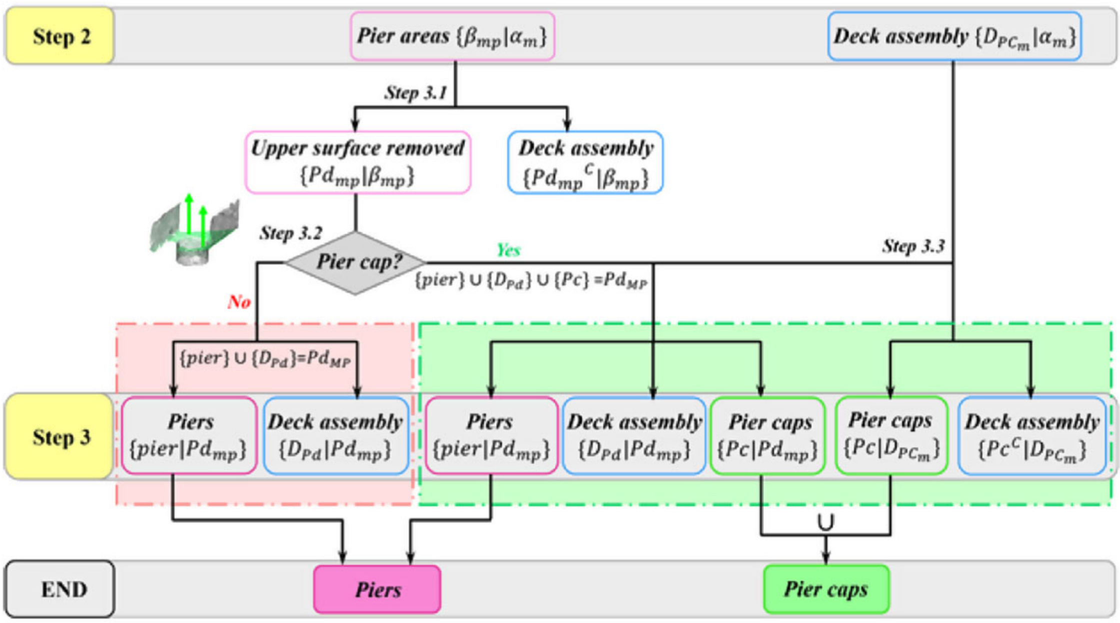

Step 2
Pier areas $\{\beta_{mp}|\alpha_m\}$
Deck assembly $\{D_{PC_m}|\alpha_m\}$
Step 3.1
Upper surface removed $\{Pd_{mp}|\beta_{mp}\}$
Deck assembly $\{Pd_{mp}{}^C|\beta_{mp}\}$
Step 3.2
Pier cap?
Yes
No
Step 3.3
$\{pier\} \cup \{D_{Pd}\} \cup \{Pc\} = Pd_{MP}$
$\{pier\} \cup \{D_{Pd}\} = Pd_{MP}$
Step 3
Piers $\{pier|Pd_{mp}\}$
Deck assembly $\{D_{Pd}|Pd_{mp}\}$
Piers $\{pier|Pd_{mp}\}$
Deck assembly $\{D_{Pd}|Pd_{mp}\}$
Pier caps $\{Pc|Pd_{mp}\}$
Pier caps $\{Pc|D_{PC_m}\}$
Deck assembly $\{Pc^C|D_{PC_m}\}$
$\cup$
END
Piers
Pier caps

Fig. 5. Input/output flowchart of Step 3.

obtain slices $S_Y = \{S_{jy}|\alpha_m\}$. Again, the value of $range_{jz}$ for each slice S_{jy} is extracted. The method classifies S_{jy} as a pier area slice if the Equation (2) is satisfied; otherwise, S_{jy} is considered a deck assembly slice:

$$\text{if} \quad range_{jz} > \rho_2 \left|\max\{z_i|\alpha_m\} - \min\{z_i|\alpha_m\}\right| \qquad (2)$$

then,

$$S_{jy} \leftarrow \text{pier area}$$

ρ_2 is another discrimination parameter that is used to separate the pier area from the rest in α_m. For a pier assembly without pier cap, $\rho_2 = \rho_1$; otherwise, $\rho_2 > \rho_1$. This assumption (see Section 4.1, A2) will be experimentally justified in Section 4.3.3.

3.5 Step 3—Pier cap detection

We illustrate the workflow of this step in Figure 5. We attempt to detect pier caps using surface normal through triangulation in the upper part of the pier area. Triangulation can be computed efficiently for a relatively small region without noticeably affecting either computation time or memory overhead.

3.5.1 Step 3.1—Remove upper deck slab surface. In this step, we aim to remove the upper slab surface points from the pier area(s) $\{\beta_{mp}\}$ output from Step 2. The void space between the upper and bottom slab surfaces is used as a discriminator. This blank part is consistent because the laser scanner can project laser beams only onto an object's external surface. According to the Design Manual for Roads and Bridges (Highways England, 2017), the general transverse maximum gradient is defined to be 5% (1/20) so that the lower bound of upper slab points is $\lambda_{min} = 5\% W_{\beta_{mp}}$, where $W_{\beta_{mp}}$ is the width of β_{mp} (Figure 6) 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. There should be $5\% W_{\beta_{mp}} < \Delta\lambda < \rho_{3a} H_{\beta_{mp}} < \rho_1 H_{\beta_{mp}}$, where ρ_{3a} is the slab thickness ratio estimation (see Section 4.3.3). The points in $\Delta\lambda$ are then removed and the remaining points in pier area(s) are denoted as $\{Pd_{mp}\}$ (Figure 6 upright).

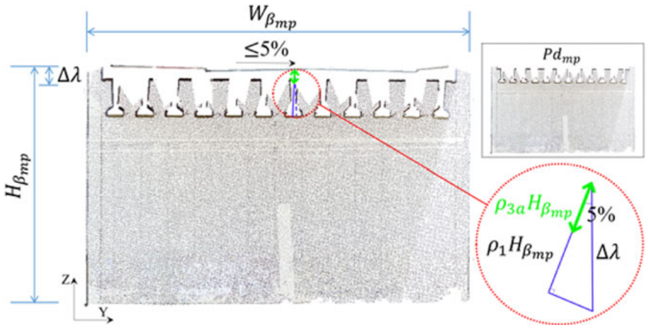


Fig. 6. Upper slab surface range $\Delta\lambda$ in β_{mp}.

3.5.2 Step 3.2—Pier cap detection at top of piers. This step aims to detect pier caps $\{Pc|Pd_{mp}\}$. The input is the refined pier areas $\{Pd_{mp}\}$ output from Step 3.1. Pier caps are underneath the slab, playing an important role in

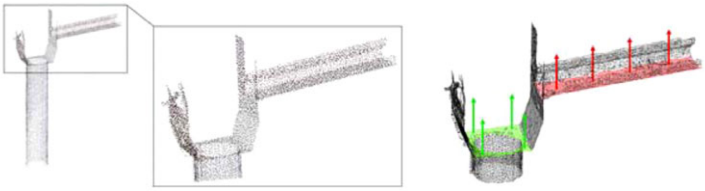

(a) Pd_{mp} (b) $upper_{Pd_{mp}}$ (c) Normal detection

Fig. 7. Pier cap detection.

distributing concentrated loads from the superstructure evenly over the area of the piers. For each pier assembly α_m:

Scenario 1. A single pier area is detected in the pier assembly where it belongs to Equation (3), and the pier area extends almost the full width of the pier assembly (Equation 4):

$$\text{if}\begin{cases}\beta_M = 1 & (3)\\ \text{and} & \\ W_{\beta_{mp}} \cong W_{\alpha_m} & (4)\end{cases}$$

where $\beta_M = \{\beta_{m1}, \beta_{m2} \ldots \beta_{mp}\}$ (see Figure 4b). Then, we consider α_m is a wall-type-pier assembly as the single wall-type-pier supports the whole deck assembly above. As a result, a pier cap does not exist (see Section 4.1, A3a).

Scenario 2. For a pier assembly with 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, the pier assembly α_m may or may not have a pier cap (e.g., multiple columns without cap). This scenario is more complex and requires further detection.

Given that pier caps are located on the top of the pier, the upper part of Pd_{mp} (i.e., the top ρ_2) is used to detect the pier cap (Figure 7a). We denote this part as $upper_{Pd_{mp}}$, which contains the deck assembly's bottom surface, the pier cap (if it exists) and a small part of the pier (Figure 7b). Then, we generate the mesh for $upper_{Pd_{mp}}$ and compute the normal of each triangular surface. Estimating a normal per given triangle is completed by the cross product of two vectors on this triangle. Define $\{\vec{n}_t\ (n_t^x,\ n_t^y,\ n_t^z) : t = 1, 2, \ldots, T\}$ to be the normal vectors of the triangles. Normal indicates the surface orientation. If a cluster of surface normal is revealed in $upper_{Pd_{mp}}$ whose orientations are quasi-parallel to Z-axis (i.e., downward- or upward-oriented normal), where

$$-90^\circ < \text{actan}\,(\theta_t) < -85^\circ$$

or

$$85^\circ < \text{actan}\,(\theta_t) < 90^\circ,\ \text{where } \theta_t = \frac{n_t^z}{\sqrt{n_t^{x2} + n_t^{y2}}} \quad (5)$$

and if those normal are found around the level ρ_1 $(\max\{z_i|\beta_{mp}\} - \min\{z_i|\beta_{mp}\})$ (Figure 7c, red), then, the points (i.e., feature points) constituting these surfaces together with the points in $upper_{Pd_{mp}}$ that above the feature points are classified as deck assembly. Otherwise, the pier cap feature points are detected if a cluster of downward- or upward-oriented normal is found around the level ρ_2 $(\max\{z_i|\beta_{mp}\} - \min\{z_i|\beta_{mp}\})$ (i.e., A3b, Section 4.1) (Figure 7c, green). The method iterates through $\{Pd_{mp}\}$ using the same procedure and the pier caps $\{Pc|Pd_{mp}\}$ and the piers $\{pier|Pd_{mp}\}$ are acquired.

3.5.3 *Step 3.3—Pier cap extraction from deck assembly.*

The detected pier caps in Step 3.2 imply that they should also be present in D_{PC_M}, which is the deck assembly output from Step 2. The pier cap parts from D_{PC_M} are extracted in the following way, where $D_{PC_M} = \{D_{PC_m}\}$ (Figure 8a).

First, the points of D_{PC_m} are projected onto the YZ-plane followed by generating density histograms along the Y-axis through which the number of points is tallied within multi-equal-width bins. The width of bin is determined using the square-root choice:

$$bin_w = \frac{|\max\{y_i|Cluster\} - \min\{y_i|Cluster\}|}{\sqrt{n}} \quad (6)$$

where $Cluster$ represents D_{PC_m} and n is its point count. Then, the bins are clustered using the gaps between them (Figure 8b), so that D_{PC_m} is segmented (Figure 8c). Denote the segments as $\{\gamma_{m(p+1)}\}$ and then a slicing procedure along the X-axis of $\{\gamma_{m(p+1)}\}$ is performed. For $\gamma_{m(p+1)}$, the pier cap area is detected if $range_{jz} > \rho_{3b}|\max\{z_i|\gamma_{m(p+1)}\} - \min\{z_i|\gamma_{m(p+1)}\}|$ (Figure 8d), where $\rho_{3b} = \frac{\rho_1}{\rho_2}$ with ρ_1 and ρ_2 being determined in Steps 1 and 2, respectively (see Section 4.1, A3c). Next, the procedure is similar to Step 3.1 and Step 3.2. The extracted pier cap area is considered a smaller scale

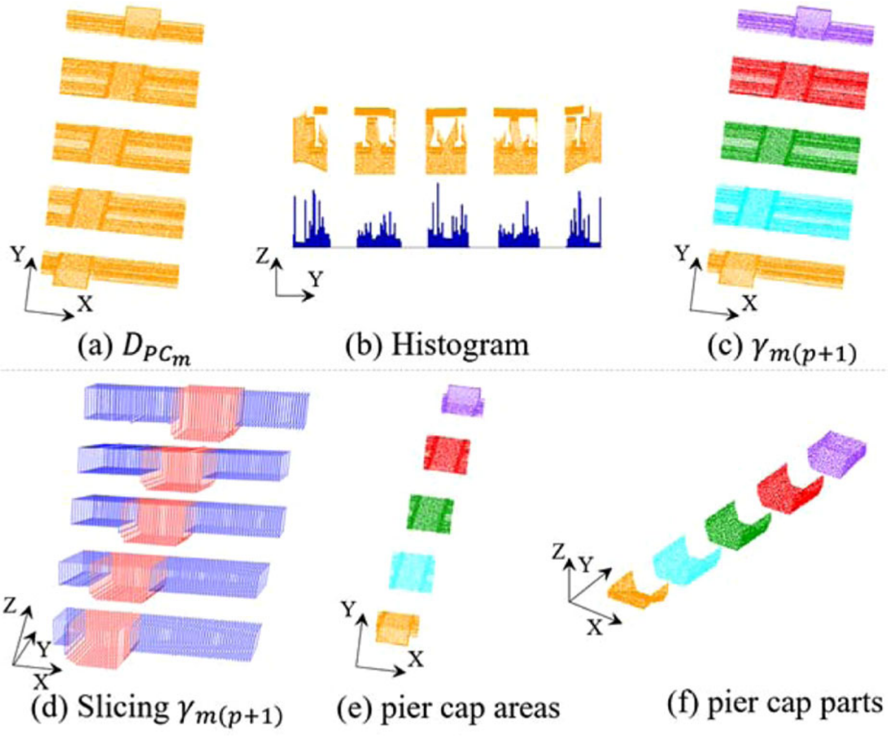


Fig. 8. Extract pier cap from D_{PC_M}.

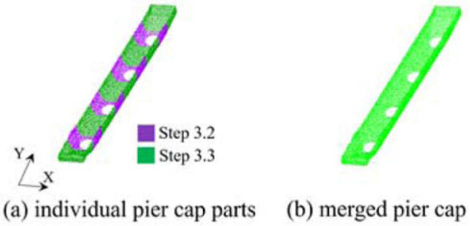


Fig. 9. Merge pier cap parts.

of β_{mp}. The upper slab surface points are removed and classified as deck assembly, followed by triangulation to detect and classify the deck assembly's bottom surface points. The pier cap parts $\{Pc|D_{PC_m}\}$ are finally acquired (Figure 8f). In the end, both pier cap parts output from Step 3.2 and Step 3.3 are merged (Figure 9).

3.6 Step 4—Girder detection

Step 4 aims to detect girders in the deck assembly. This is achieved in two substeps: (1) segment the deck assembly into several segments $\{deck_\omega\}$ and (2) detect girders in each segment $\{girders|deck_\omega\}$.

3.6.1 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, which is composed of slab and girders (if they exist). This involves piecing together all point clusters classified as deck assembly in the previous steps (Figure 10).

For a beam-slab bridge, the length of the girder (i.e., beam) depends on the span, which is the distance between the two intermediate supports (Wai-Fah and Lian, 2014). We need to split the whole merged deck assembly into several segments to find the appropriate length of span. The best cutting planes are not necessarily orthogonal to the horizontal alignment of the bridge (i.e., X-axis of the deck assembly), but rather depend on the orientation of the expansion joints. This is because two adjacent deck assembly segments must be interconnected by the expansion joints as per the Highway Agency's BD 33 design code (Highways England, 1994). The choice of joint depends on many factors, including imposed loadings, anticipated movement, temperature range, deck shortening, and deck rotation. Pier clusters and pier caps are then oriented based on the joints. Hence, the problem of finding the best cutting planes is transformed into orientation determination of the pier clusters or pier caps (Figure 11). We employ a 3D oriented-bounding-box (OBB) to capture the orientation property. OBB is the tightest oriented box depicting a given 3D point set. All bridges have piers, but not all of them have pier caps. So, a pier cluster

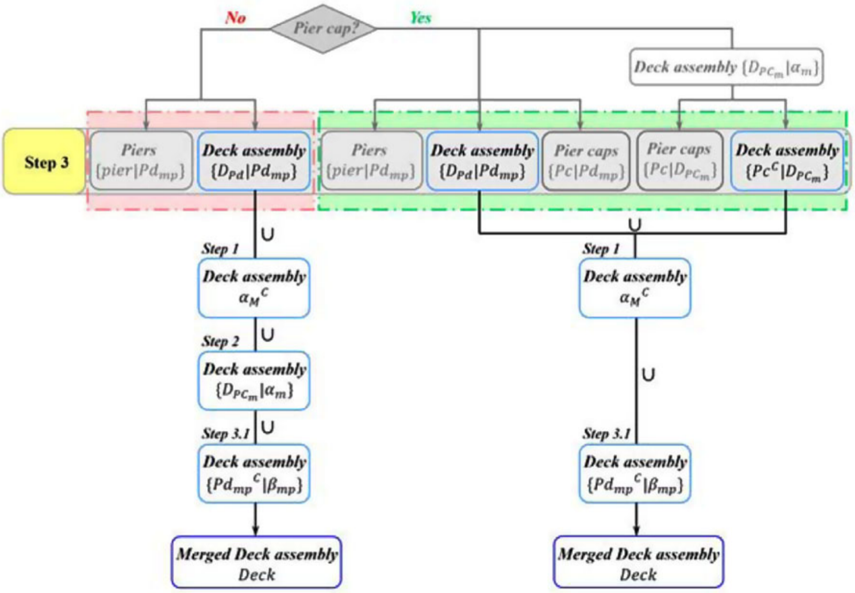

No
Pier cap?
Yes
Deck assembly $\{D_{PC_m}|\alpha_m\}$
Step 3
Piers $\{pier|Pd_{mp}\}$
Deck assembly $\{D_{Pd}|Pd_{mp}\}$
Piers $\{pier|Pd_{mp}\}$
Deck assembly $\{D_{Pd}|Pd_{mp}\}$
Pier caps $\{Pc|Pd_{mp}\}$
Pier caps $\{Pc|D_{PC_m}\}$
Deck assembly $\{Pc^C|D_{PC_m}\}$
∪
Step 1
Deck assembly $\alpha_M{}^C$
∪
Step 2
Deck assembly $\{D_{PC_m}|\alpha_m\}$
∪
Step 3.1
Deck assembly $\{Pd_{mp}{}^C|\beta_{mp}\}$
Merged Deck assembly Deck
∪
Step 1
Deck assembly $\alpha_M{}^C$
∪
Step 3.1
Deck assembly $\{Pd_{mp}{}^C|\beta_{mp}\}$
Merged Deck assembly Deck

Fig. 10. Merging to acquire the whole deck assembly.

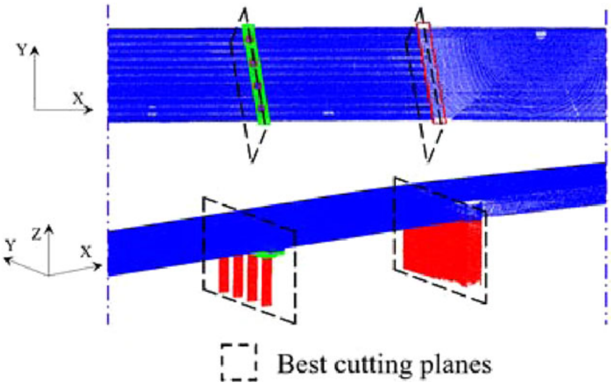


Fig. 11. Best cutting planes.

is used to create an OBB. Only its bottom-half is used to avoid retained boundary points from the pier cap as those points might lead to a shifted OBB.

OBB replaces a point set with a parallelepiped of 12 edges and 8 vertices. Let $\{d_1, d_2, d_3, d_4\}$ and $\{p_1, p_2, p_3, p_4\}$ be the four upper vertices of the merged deck assembly *Deck* and the bottom-half part of a pier cluster, respectively (Figure 12). Let M1, M2 be the midpoints of $line_{p1p2}$ and $line_{p3p4}$. MP1 and MP2 are two orthogonally projected points of M1 and M2 onto the plane pl_{d1-d3}. Then, d_5 and d_6 are two intersection points of $line_{d1d2}$ & $line_{MP1MP2}$ and $line_{d3d4}$ & $line_{MP1MP2}$, respectively. *Deck* is cut along the plane consisting of d_5, d_6, M1, and M2 so that the plane pl_{MP1-M2} is deemed the best cutting plane. Next, the method examines whether each point of the deck assembly is inside the $ConvexHull_{d1d4d6d5}$:

$$\text{if} \quad \forall\, p_i \in \left\{p_{ix,y}|Deck\ _{x,y}\right\}$$

is inside $ConvexHull_{d1d4d6d5}$: $\{p_i|Deck\} \in deck_1$; otherwise,

$$\{p_i|Deck\} \in deck_1^{\mathrm{C}}$$

This process is recursively performed until the entire deck assembly is segmented into multiple segments $\{deck_\omega : \omega = 1, 2, \ldots (m+1)\}$, where m is the number of pier clusters (equals to number of pier assemblies) (Figure 13).

3.6.2 Step 4.2—Girder detection in the deck assembly segment. We now 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, because the original projection results of $deck_\omega$ might be "muddy" due to a curved bridge elevation. Rotation is conducted through a grid search in a range of angles $\{\xi\}$, where $\{\xi\}$ = [−3.4°, 3.4°], deduced from the general longitudinal maximum gradient (6%) (Highways England, 2017). A density histogram $\mathcal{H}_Z$ along the Z-axis is employed for evaluating if a best rotation is reached. The best rotation angle is determined using:

$$\breve{\xi} = \operatorname{argmax}\{std_{\mathcal{H}_Z(deck_\omega)}\} \qquad (7)$$

where $std_{\mathcal{H}_Z(deck_\omega)}$ is the standard deviation of the point counts in the bins. Empirical studies revealed that the best projection determination is not sensitive to the bin

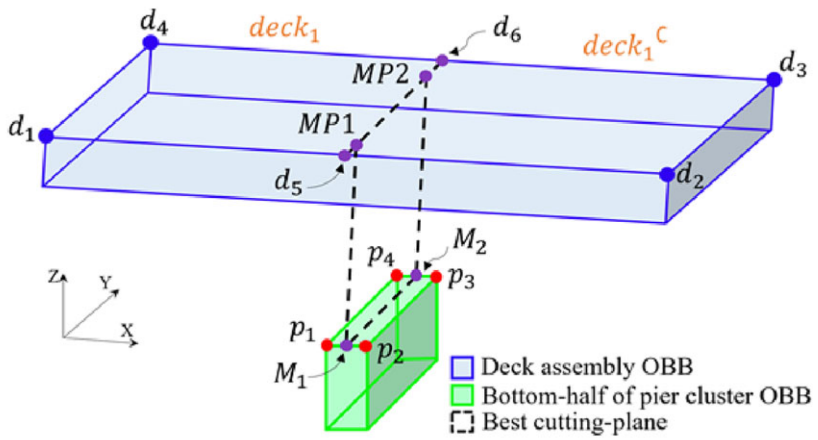


Fig. 12. Find the best cutting plane.

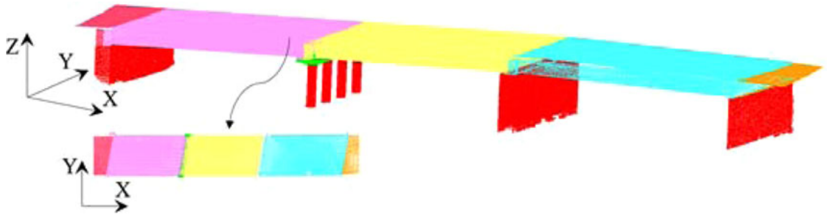


Fig. 13. Split the whole deck assembly into $\{deck_\omega\}$.

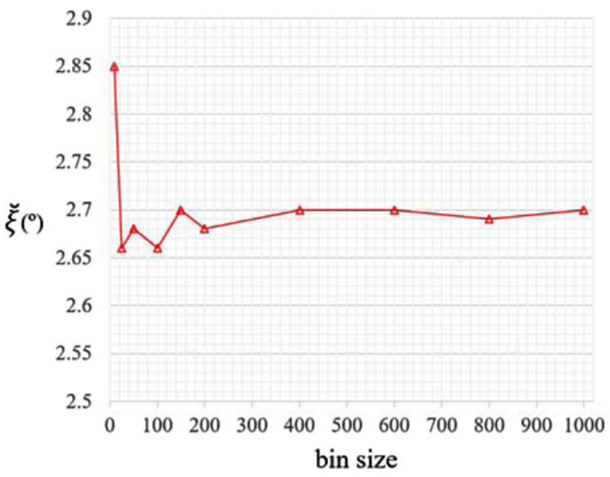


Fig. 14. Best rotation found vs. different bin size.

size (varies from #bin = 10 to #bin = 1000, μ = 2.70°, σ = 0.05°) (Figure 14). The bin size is then derived using Equation (6). $std_{\mathcal{H}_Z(deck_\omega)}$ is a stronger indicator than simply the maximum point count bin, because the elevation of girder depends on that of the slab. The best projection may not necessarily be given by the bin with the maximum point count resulting possibly from a concentration of unevenly distributed points. The best projection view can be found once the standard deviation of histogram bins on the Z-axis reaches its maximum. Figure 15 demonstrates an example where the best rotation ($\breve{\xi} = 2.7^\circ$) is obtained when the biggest $std_{\mathcal{H}_Z(deck_\omega)}$ (1,178) returns.

Next, only the bottom ρ_4 (%) points of $deck_{\omega(\breve{\xi})}$ (denoted $b_{deck_{\omega(\breve{\xi})}}$) are used for girder detection, where $\rho_4 = \frac{\rho_1 - \rho_{3a}}{\rho_1}$, with the thickness of deck assembly as well as that of the slab being estimated to be roughly ρ_1 (obtained from Step 1) and ρ_{3a} (obtained from Step 3.1), respectively. The removal of the deck assembly's upper part (top $(1 - \rho_4)$) is crucial as many more points are captured from deck external surface, overpowering the girder points and make the geometric features uninformative. The extremities of $deck_{\omega(\breve{\xi})}$ are also excluded to avoid noise from bridge accessory components. Density histograms $\mathcal{H}_Y$ are drawn along the Y-axis of $b_{deck_{\omega(\breve{\xi})}}$ using Equation (6) ($Cluster \leftarrow b_{deck_{\omega(\breve{\xi})}}$) followed by generating the normalized probability of the point density (Figure 16). The density probability is uniformly distributed with significantly lower variance when there is no girder (i.e., slab bridge) whereas significant peaks can be observed in the distribution with non-trivial variance when girders exist (i.e., beam-slab bridge). For the latter, a binary list (0, 1) is created after thresholding out all small counts (small-count bin ← 0,

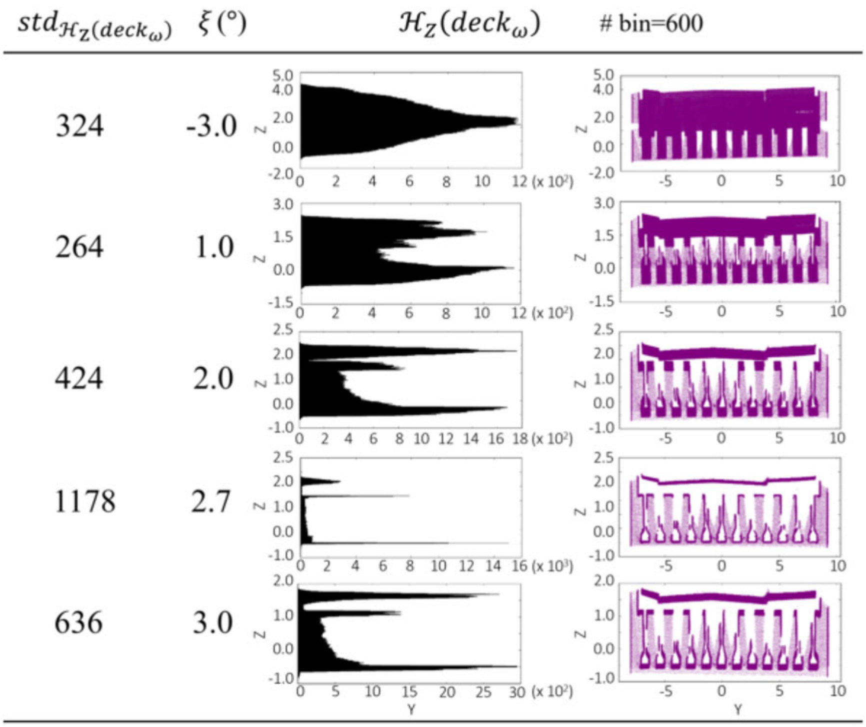


Fig. 15. Best rotation search in $deck_\omega$.

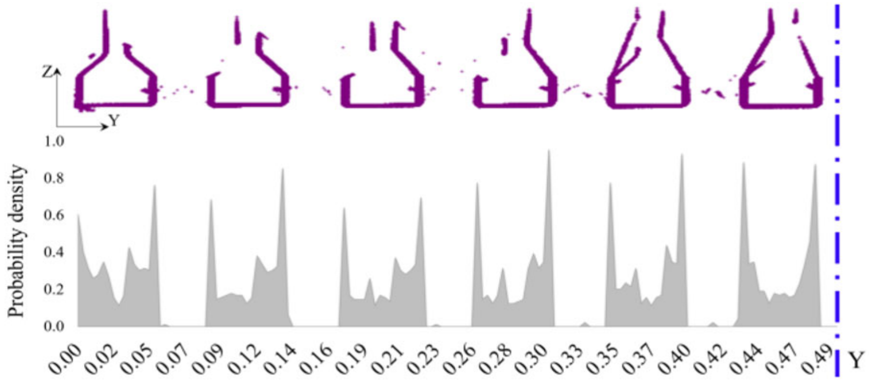


Fig. 16. Girder segmentation.

big-count bin ← 1). This list is further denoised through a simple k-NN filter, which works as a voting scheme. It checks the label of neighboring bins, and then assigns a candidate label to the investigated bin. This process is iteratively performed until optimal clusters are returned, meaning the "1" chunks have similar length because the girder section type is identical in a specific span (see Section 4.1, A4). The bottom flange can infer a collection of possible girder section types (e.g., Y, U, or SY beams). We then use the web depth (i.e., girder's height) extracted along the best projection view to decide a specific girder type so that the girders can be separated accordingly from the slab.

All the over-segments from Step 1 to Step 4 are merged as per their class labels. The 4-step top-down recursive detection method then terminates.

4 RESEARCH METHODOLOGY

4.1 Assumptions

According to national standards (Highways England, 2017), the proposed method is feasible in the context of RC bridge modeling under the following conditions, which are also confirmed in our experiments.

A1. Pier assembly and deck assembly can be separated using the ratio ρ_1.

A2. Pier area and deck assembly can be separated using the ratio ρ_2.

A3a. A pier assembly does not contain a pier cap if a single pier area is detected in the pier assembly and the width of the pier is almost the same as that of the pier assembly.

A3b. Surface normals are distinct features that can be used to distinguish a pier cap part apart from a pier.

A3c. Pier cap parts can be extracted from the deck assembly using the ratio $\rho_{3b} = \frac{\rho_1}{\rho_2}$.

A4. The density histograms along the best view can be used to segment the girders in the deck assembly segment.

In particular, A1 and A2 are validated experimentally in Section 4.3.3 whereas A3a–c and A4 are validated in Section 4.4. We also assume an RC bridge satisfies the following conditions:

A5. The piers are quasi-vertical.

A6. The on-site clutters and irrelevant points are properly removed manually.

4.2 Data and methods

To test our hypothesis, we used a FARO Focus 3D X330 Terrestrial Laser Scanner (ranging error ±2 mm at 10 m, self-leveling: accuracy 0.015° (range ±5°)) to collect PCD of 10 RC highway bridges around Cambridgeshire, United Kingdom (Figure 17). The locations (GPS), vertical clearance (denoted VC), and other bridge data information are given in Table 1. The large size of the excluded data (i.e., nondata) is mainly due to the manual removal of the on-site traffic noise, trees, large ground surfaces, ramps, and abutments.

The samples we used to test our detection method are bridge components. The sample size depends on the standard normal deviation, error limit, and category proportion. As the calculated sample size decreases, the margin of error grows, so theoretically, more bridge data ($\approx$20) is needed to achieve a good confidence level (CL = 90%) with a relatively small error limit (EL = 0.2). However, the cost and risk of data collection is extremely high, as researchers must operate a laser scanner next to a live carriageway and face significant traffic hazards. We therefore consider this proof of concept study validated if we achieve a low performance variance over the 10 bridge data sets (CL = 90%, EL = 0.3). To our best knowledge, this study has the largest collection of real-world RC bridge point clouds.

We took an average of 17 scans per bridge. The distance between the captured scan points was set to be in 7.67 mm over a scan distance of 10 m (except for inaccessible standpoints). We registered all raw scans using the FARO Scene software. On average, the registration time was 10.6 hours per bridge. Note that several factors may affect the measuring accuracy, such as low temperatures, which may condense elements inside the scanner. As the data-collection work was conducted during the cold winter season, we warmed the scanner up before every task until its internal temperature stabilized. We also used the built-in inclinometer to store the inclination of the leveled scanner.

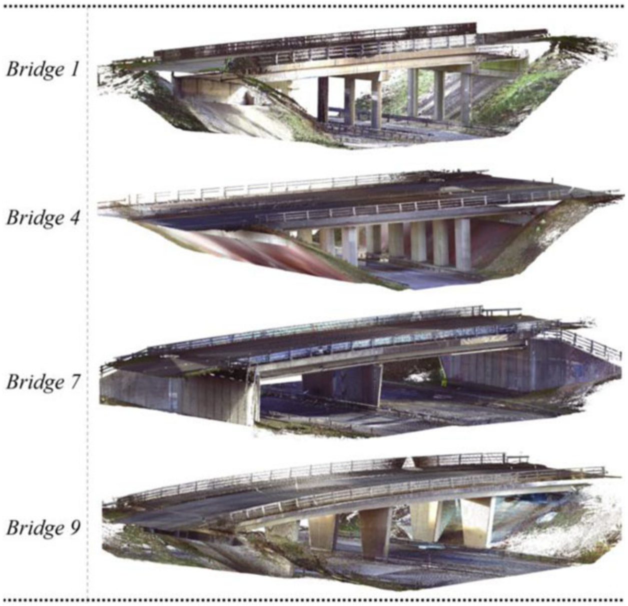


Fig. 17. Four examples of RC bridge point clouds.
Note: The on-site traffic, trees, and vegetation are removed.

The relative accuracy achieved (denoted acc in Table 1) was estimated from the averaged fraction between pair reference-point distance in the registered scan data and the corresponding on-site pair-point distance using a measuring tape, independent of other error sources. The spatial completeness of the data sets was computed based on rough estimation of the occlusion-data ratio (Table 1).

Our analyses consist of two parts. The first part is to experimentally define the optimal values of the two 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 hyperparameters (ρ_{3a}, ρ_{3b}, and ρ_4). The second part is to assess the proposed method at the level of bridge structural components using both bounding-box-wise (inspired from the outline evaluation suggested in Truong-Hong and Laefer, 2015) and point-wise performance metrics.

Table 1
Metadata of 10 RC bridge point cloud data sets

	Bridge 1		*Bridge 2*		*Bridge 3*		*Bridge 4*		*Bridge 5*	
GPS	*52.216821, 0.243050*		*52.214149, 0.215174*		*52.214241, 0.216835*		*52.075854, 0.193352*		*52.229033, 0.179660*	
Scanning	3.5 hours	18 scans	3.3 hours	17 scans	3.2 hours	19 scans	4 hours	25 scans	3.2 hours	16 scans
Registration	14 hours	acc:0.98	8 hours	acc:0.96	12 hours	acc:0.97	14 hours	acc:0.96	6 hours	acc:0.97
Completeness	97.9%		98.0%		98.3%		95.0%		98.7%	
Original size	26,235,309		11,852,062		12,284,105		12,655,082		10,157,137	
Nondata ratio	23.7%		34.1%		43.5%		36.8%		55.1%	
VC (m)	Span 1	Span 2	Span 1	Span 2	Span 1	Span 2	Span 1	Span 1		
	5.00	5.20	5.21	5.30	4.96	5.15	5.90	5.36		
	Bridge 6		*Bridge 7*		*Bridge 8*		*Bridge 9*		*Bridge 10*	
GPS	*52.359763, −0.40601*		*52.311695, −0.12829*		*52.274820, 0.503124*		*52.276257, 0.491071*		*51.973192, −0.19567*	
Scanning	2.5 hours	17 scans	2 hours	14 scans	2.3 hours	16 scans	2.2 hours	16 scans	2 hours	16 scans
Registration	8 hours	acc:0.96	6 hours	acc:0.97	6 hours	acc:0.96	24 hours	acc:0.96	8 hours	acc:0.98
Completeness	98.2%		92.2%		97.8%		98.8%		98.5%	
Original size	79,787,269		53,708,475		81,610,875		80,915,621		77,688,416	
Nondata ratio	14.5%		42.8%		11.5%		14.2%		11.1%	
VC (m)	Span 1	Span 2	Span 1	Span 2	Span 1	Span 2	Span 1	Span 2	Span 1	Span 2
	5.89	5.60	5.20	5.40	5.68	6.05	5.32	5.60	5.27	5.15

4.3 Activities

4.3.1 Data preparation. We developed a user-defined bounding-box functionality to manually delete irrelevant points such as the on-site traffic, vegetation, ground surface, and so on. This is the only required manual work. The proposed four-step object-detection method is fully automatic without any human intervention, because it is easier for a human modeler to delete irrelevant points than a computer, because the latter requires a sophisticated algorithm to identify which points are irrelevant or are noise. On the contrary, it is difficult for a human modeler to precisely segment a point cloud of complex geometries on 2D computer monitor.

After randomly downsampling, an RC bridge point cloud with the key components contains less than 1 million points. The reason for downsampling is that the original size was not used for manual detection as the commercial software is difficult to handle large data sets. Next, we aligned the cropped bridge point cloud using PCA so that the major axes of the bridge are positioned roughly parallel to the global axes X-Y-Z (none of the bridges can be positioned exactly parallel to the axes due to their real-world skewed geometry).

We created three ground-truth (GT) data sets (optimal desired output to compare against) by manually conducting Step 1 (i.e., *GT A*), Step 2 (i.e., *GT B*), and the entire solution (i.e., *GT C*):

GT A: For a given bridge point cloud input, we segmented it into two clusters, namely, deck assembly and pier assembly and assigned them corresponding point-wise labels.

GT B: For a pier assembly point cloud input, we segmented it into two clusters, namely, deck assembly and pier area and assigned them corresponding point-wise labels.

GT C: For a given bridge point cloud input, we segmented it into individual point clusters as per their semantic class—that is, structural components, including slab, piers, pier caps (if they exist), and girders (if they exist). Each of these point clusters was bounded by an oriented-bounding-box (hereafter GTBBox). We also assigned the points in each cluster their corresponding point-wise labels. Both GTBBox and manually labeled points were served as reference for comparison.

4.3.2 Implementation. We implemented the solution into a robust software prototype Gygax (https://github.com/ph463/Gygax) as a proof of concept in a desktop computer (Intel Core i7-4790K 4.00GHz CPU, 32 GB RAM, Samsung 500GB SSD). Gygax is an open research platform in C#. It uses a sparse wrapper to allow the usage of PCL and other open libraries.

4.3.3 Estimation of hyperparameters. We estimated the two hyperparameters ρ_1 and ρ_2 and compared the results against *GT A* and *GT B*, respectively.

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:

$$\mathrm{Pr_s} = \frac{\mathrm{TP_s}}{\mathrm{TP_s} + \mathrm{FP_s}} = \frac{\text{\# of correctly labeled points in cluster s}}{\text{total \# of points in cluster s}} \tag{8}$$

$$\mathrm{R_s} = \frac{\mathrm{TP_s}}{\mathrm{TP_s} + \mathrm{FN_s}} = \frac{\text{\# of correctly labeled points in cluster s}}{\text{total \# of points in ground truth cluster s}} \tag{9}$$

$$\mathrm{F1_s} = 2 \times \frac{\mathrm{Pr_s} \times \mathrm{R_s}}{\mathrm{Pr_s} + \mathrm{R_s}} \tag{10}$$

The values of ρ_1 and ρ_2 vary for different bridge configurations. To learn how sensitive the outputs of the first and the second steps of our proposed method is to ρ_1 and ρ_2, we conducted grid-search over the entire range of values of ρ_1 as well as ρ_2 within the value space (0, 1), and computed the empirical receiver operating characteristic (ROC), which depicts the trade-off between true positive rate (TPR) and false positive rate (FPR). The TPR (Equation (11)) is known as sensitivity of detection—that is the equivalent of the Recall whereas the FPR (Equation (12)) is known as probability of false alarm (1-specificity), where:

$$\mathrm{TPR_s} = \text{sensitity} = \frac{\mathrm{TP_s}}{\mathrm{TP_s} + \mathrm{FN_s}} \tag{11}$$

$$\mathrm{FPR_s} = 1 - \text{specificity} = \frac{\mathrm{FP_s}}{\mathrm{FP_s} + \mathrm{TN_s}} \tag{12}$$

A too small or too big (close to 1) ρ_1 may lead the method to consider a whole RC bridge as either pier assembly or deck assembly. By intuition, a relative big ρ_1, for example, 0.5, can be used to extract pier assemblies in Step 1 because normally, the height of pier should be much larger than that of the deck. Yet, to keep a necessary vertical clearance, the thickness of the deck is almost impossible to be as thick as 0.5 times of the height of pier assembly. We therefore should find a reasonable value of ρ_1, which is both theoretically and realistically sound. We identified the optimal ρ_1^* and ρ_2^* when the distance to the perfect classification in the ROC (i.e., FPR = 0, TPR = 1) was minimized:

$$\rho_k^* = \operatorname{argmin}\left(d_{\rho_k}\right) \tag{13}$$

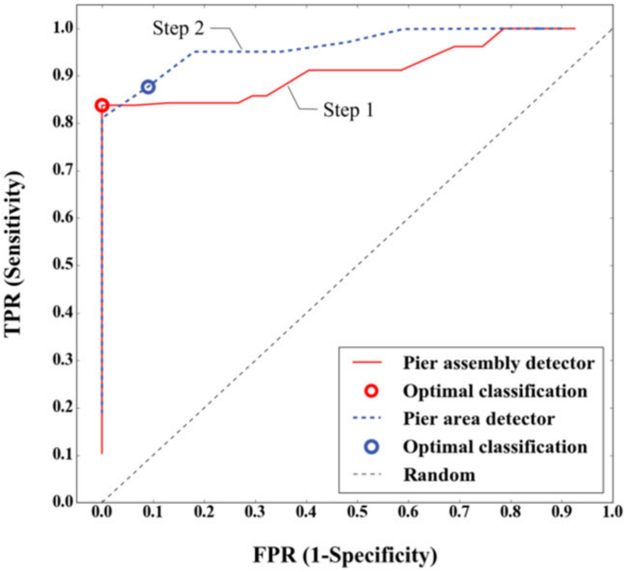


Fig. 18. ROC of detector $range_{jz}$ of pier assembly and pier area (example of *Bridge 1*).

Table 2
ρ_1^* determination

Bridge	ρ_1^*
1	0.31
2	0.25
3	0.25
4	0.25
5	0.25
6	0.25
7	0.31
8	0.20
9	0.30
10	0.30
$\overline{\rho_1^*}$	0.27
S	0.03

where $d_{\rho_k} = \sqrt{(1 - \text{sensitivity})^2 + (1 - \text{specificity})^2}$, for $k \in \{1, 2\}$, k represents the number of the step.

The optimal thickness ratio ρ_1 for each bridge was then found using Equation (13). Figure 18 shows an example of ROC curve of *Bridge 1*. Assume the samples follow a t-distribution with 9 degrees of freedom (i.e., $n - 1$, where $n = 10$), the 95% confidence interval (CI) critical value was derived from the t-table as 2.262 for calculating the true sampling distribution mean $\mu_{\overline{\rho_1^*}}$. The optimal ρ_1 was then computed as 0.27 ± 0.03, that is, $\overline{\rho_1^*} \pm t(s/\sqrt{n})$, where $\overline{\rho_1^*}$ is the sample mean and s is the sample standard deviation (Table 2). We also computed $\mu_{\overline{\rho_1^*}}$ using the bootstrapping technique which resamples the data by replacement with a same sample size of 10 followed by repeating 1,000 times. The 95% CL upper bound was estimated to be 0.29, which is in line with that of the *t*-statistic. We chose the upper bound of the *t*-statistic to set ρ_1^* as 0.30 rather than its lower bound because all indicators such as Pr, R, F1, FPR, and d_{ρ_1} had a good balance when $\rho_1 = 0.30$ (e.g., $\overline{\text{F1}}_{0.3} = 0.84$, $\overline{\text{F1}}_{0.24} = 0.74$). More importantly, setting a bigger ρ_1^* can avoid extracting too many false positives (e.g., $\overline{\text{FPR}}_{0.3} = 0$, $\overline{\text{FPR}}_{0.24} = 0.06$).

Then, we also grid searched the optimal ρ_2 in Equation (2) in the same way followed by plotting ROC (Figure 18) at various threshold settings ρ_2 for a pier assembly sample. The 95% CI of *t*-statistic for ρ_2 was derived to be 0.36 ± 0.03. Likewise, we chose the upper bound to set ρ_2 as 0.39. Once we obtained ρ_1^* and ρ_2^*, we calculated $\rho_{3b} = \frac{\rho_1^*}{\rho_2^*} \approx 0.80$.

Then, we estimated the value of ρ_{3a}, which is the slab thickness ratio estimation used to remove the upper slab surface points from the pier area(s) $\{\beta_{mp}\}$ (see Step 3.1).

The pier areas $\{\beta_{mp}\}$ are relatively small regions compared with an entire bridge. It is reasonable to consider the vertical elevation of β_{mp} constant so that we can present the probability of point density for $\{\beta_{mp}\}$. Suppose the distribution of the points along Z-axis of $\{\beta_{mp}\}$ is a collection of probabilities of locations on the pier area surface where the points are located in. The density estimations of pier area samples from slab bridges showed an obvious void space between the top surface and the bottom surface of a slab. The 95% CI of the normalized slab bottom level was 0.84 ± 0.06. Likewise, for all the pier area samples from beam-slab bridges. The 95% CI of ρ_{3a} was estimated to be 0.76 ± 0.04. These statistics suggested that for slab and beam-slab bridges, the top 10% β_{mp} points include the upper slab surface so as we can remove and classify them into deck assembly. This slab thickness ratio ρ_{3a} is especially used for beam-slab bridges in Step 4 for girder segmentation. We chose the slab bottom level estimation of beam-slab bridge to estimate the ρ_{3a} as 0.28 and 0.2 (i.e., $1 - (0.76 \pm 0.04)$). Taking into account the shallow girder and the effect of transverse gradient, we chose $\rho_{3a} = 0.2$ so that then, $\rho_4 = \frac{\rho_1^* - \rho_{3a}}{\rho_1^*} \approx 0.33$.

4.4 System validation and results

We evaluated the whole proposed method of the prototype on the level of structural components with the optimal hyperparameters identified in Section 4.3.3. Then we compared the results against *GT C*. The method generated an oriented-bounding-box for each segmented point cluster (hereafter AutoBBox) and assigned a semantic instance label to each point. The run-time for

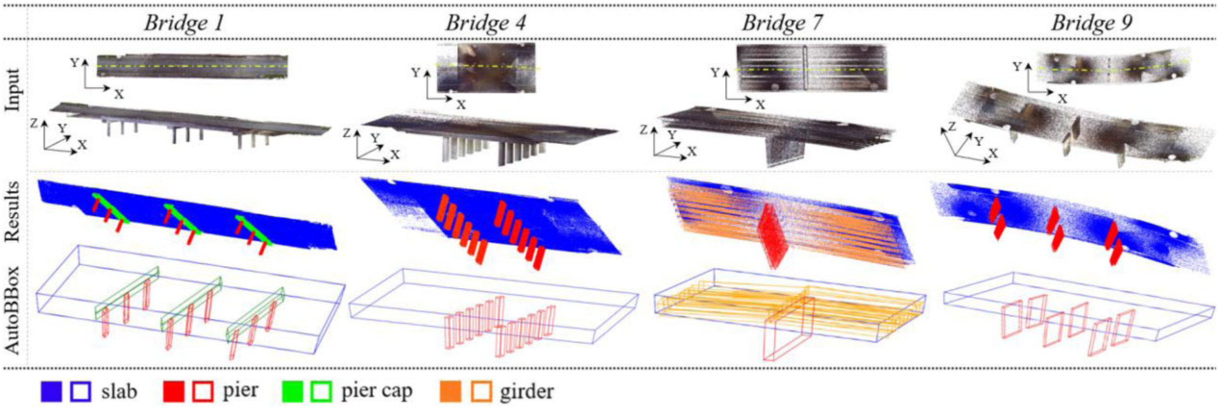


Fig. 19. Detection results and AutoBBoxes for point clusters of four example bridges.

each bridge point cloud was on average 8.02 ± 3.02 minutes (less than one million points), including all four major steps of the proposed method. This means a dramatic decrease of 90% compared to *GT C*.

We first compared AutoBBoxes against GTBBoxes and evaluated the proposed method's performance using the following conditions. For a specific point cluster generated from the proposed method, 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 Pr, R, and F1 in bounding-box-wise metrics for each bridge were generated using similar formulas as Equations (8)–(10). Specifically, the Pr is the number of correctly detected point clusters out of the total number of AutoBBoxes of a bridge data set, the R is the number of correctly detected point clusters out of the total number of GTBBoxes of a bridge data set, and the F1 is the harmonic mean of the Pr and R.

Figure 19 illustrates the point-wise and bounding-box-wise detection results. As shown, *Bridge 9* slab contains obvious skew. The average Pr, R, and F1 of the bounding-box-wise performance of 10 bridges were 100%, 98.5%, and 99.2% (Table 3), respectively. All of the components were correctly detected except pierCap1 ($\varepsilon = 71.9\%$), pierCap2 ($\varepsilon = 81.9\%$) of *Bridge 1*. Yet, few points in these clusters were detected as FP ($\text{FDR}_{\text{pierCap1}} = 4.4\%$, $\text{FDR}_{\text{pierCap2}} = 8.6\%$), where the false discovery rate (FDR) for each point cluster is

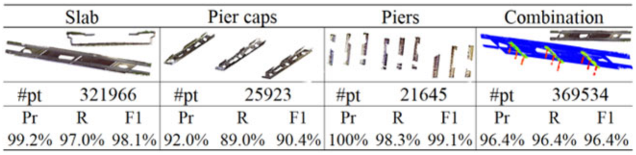
Table 3
Bounding-box-wise performance and point-wise performance in micro-average of 10 bridge data sets

	Bounding-box-wise				*Point-wise*	
Bridge	*#FN*	*#FP*	*#TP*	*F1*	*#FP*	*Pr/R/F1 micro*
1	2	0	11	91.7%	2493	99.5%
2	0	0	4	100%	4839	99.0%
3	0	0	4	100%	3486	99.3%
4	0	0	13	100%	220	100%
5	0	0	3	100%	3256	99.3%
6	0	0	7	100%	1509	99.7%
7	0	0	20	100%	49412	89.1%
8	0	0	7	100%	2351	99.5%
9	0	0	7	100%	2777	99.4%
10	0	0	7	100%	234	100%
Avg.				99.2%		

Table 4
Confusion-matrix of *Bridge 2*

		Auto label				
		slab	*pier11*	*pier12*	*pier13*	*Pr*
GT label	*slab*	388525	2639	574	1624	98.8%
	pier11	0	37935	0	0	100%
	pier12	0	0	31804	0	100%
	pier13	2	0	0	36897	100%
	R	100%	93.5%	98.2%	95.8%	

Table 5
Effect of large occlusions on *Bridge 1*



Slab			Pier caps			Piers			Combination		
#pt	321966		#pt	25923		#pt	21645		#pt	369534	
Pr	R	F1	Pr	R	F1	Pr	R	F1	Pr	R	F1
99.2%	97.0%	98.1%	92.0%	89.0%	90.4%	100%	98.3%	99.1%	96.4%	96.4%	96.4%

$$\mathrm{FDR_s} = \frac{\mathrm{FP_s}}{\mathrm{FP_s} + \mathrm{TP_s}} \tag{14}$$

Therefore, although bounding-box-wise metrics can give us a general picture of the performance, they are too sensitive to the locations of misclassified points, which largely affected the values of $d(C_{auto}, C_{gt})$.

We repeated the system evaluation with point-wise metrics, that is, Equations (8)–(10). Herein, the "*S*" in Equations (8)–(10) refers to any specific final point cluster generated from our proposed solution.

For a specific bridge point cloud, we computed the micro-average scores. In micro-average, we summed up individual TP, FP and FN from all point clusters to obtain the statistics:

$$\mathrm{Pr_{micro}} = \frac{\sum_{s=1}^{|S|} \mathrm{TP_s}}{\sum_{s=1}^{|S|} \mathrm{TP_s} + \sum_{s=1}^{|S|} \mathrm{FP_s}} \tag{15}$$

$$\mathrm{R_{micro}} = \frac{\sum_{s=1}^{|S|} \mathrm{TP_s}}{\sum_{s=1}^{|S|} \mathrm{TP_s} + \sum_{s=1}^{|S|} \mathrm{FN_s}} \tag{16}$$

where |S| is the number of generated point clusters in this given bridge point cloud. The micro-average F1-score is simply the harmonic mean of $\mathrm{Pr_{micro}}$ and $\mathrm{R_{micro}}$.

Assumptions A3a – A3c were justified as the method recognizes that there was no pier cap in the pier assemblies of *Bridges 2, 3, 5,* and *7* (wall-type-pier) and *Bridges 4, 6, 8, 9,* and *10* (multiple columns without cap). The method correctly identified the pier caps in *Bridge 1*.

Table 3 shows that the proposed method achieved remarkable performances: the highest micro-average of Pr/R/F1 was rounded up to 100% and the lowest was 89.1% (for multiclass case, the micro-averages yields result in a mathematically equivalent definition for precision and recall, thus equivalent F1-score).

4.5 Discussion

An example of confusion matrix of *Bridge 2* is given in Table 4. Although we achieved high detection rates with the PCD of all 10 RC bridges in both the bounding-box-wise and the micro-average assessments (Table 3 and Table 6), the FDR of some point clusters revealed that the proportion of the FP points is not insignificant. This is especially true for *Bridge 7*. There are a few components that reached very high detection precision, such as the pier (97.2%) and many girders, such as girder14 (100%), girder15 (100%), girder22 (100%), and girder23 (100%), among others. However, some points were not properly classified. Normally, a slab cluster should contain the most populated labels in a bridge point cloud (i.e., it has the most points). For *Bridge 7* (Table 6), we had a misclassification for the slab point cluster ($\mathrm{FP_{slab}} = 37764$, $\mathrm{FDR_{slab}} = 17.7\%$). The FDR was also not trivial for girder11 (21.1%), girder19 (27.9%), girder21 (23.3%), and girder29 (20.6%). The microaverage metrics adequately captured class imbalance issues and brought the overall precision average down to 89.1%.

There are two main reasons for the reduced classification performance in *Bridge 7*. First, the significant parabolic vertical alignment of the roadway in each deck segment of this bridge made the segmentation less accurate. Future work should develop a further deck-segment slicing procedure in Step 4 to alleviate the impact of parabolic curves. Second, the girders were placed so close to each other that the gaps between adjacent girders were difficult to be seen by a scan sensor. Around 8% of the surface of *Bridge 7* was occluded (Table 1) mainly due to the fact that the points on the webs of the girders were missing. As a result, these regions were quite ambiguous, making it difficult to detect an individual girder by the proposed method. The points between adjacent girders were misclassified as slab. To learn how many occlusions are exactly acceptable, we reconducted experiments using *Bridge 1* by creating arbitrary occlusions in slab, pier caps, and piers, respectively, while others remain unchanged. Then we combined all these occlusions. The occlusion level was estimated to be 30–40%. Table 5 shows that our method achieved high detection performance. However, it is not encouraging to process such a high occlusion level data in real applications. The experiment of the best projection search of *deck*$_2$ of *Bridge 8* also proved that our method is robust to very unevenly distributed points between the upper and lower slab (Figure 20).

Table 6
Point-wise performance of *Bridges 1, 4, 7,* and *9*

	Cluster	*#FN*	*#FP*	*FDR (%)*	*ε (%)*
Bridge 1	slab	1478	48	0.1	0
	pierCap1	239	572	4.4	71.9
	pierCap2	0	800	8.6	81.9
	pierCap3	209	673	5.5	17.6
	pier11	37	0	0	1.8
	pier12	52	0	0	2.9
	pier13	54	0	0	3.1
	pier21	33	0	0	2.1
	pier22	23	0	0	2.8
	pier23	55	0	0	2.9
	pier31	20	0	0	0.9
	pier32	248	0	0	15.7
	pier33	45	0	0	3.4
Bridge 4	slab	82	114	0.03	3.4
	pier11	2	23	0.29	8.1
	pier12	9	1	0.02	1.5
	pier13	3	4	0.04	0.1
	pier14	2	6	0.02	0.2
	pier15	11	2	0.02	1.1
	pier16	2	25	0.30	1.2
	pier21	2	27	0.30	10.5
	pier22	32	2	0.02	4.4
	pier23	7	2	0.02	0.9
	pier24	14	2	0.02	2.3
	pier25	52	2	0.03	2.9
	pier26	2	10	0.12	0
Bridge 7	slab	11000	37764	17.7	37.0
	pier11	2	573	2.75	3.7
	girder11	108	2061	21.1	30.5
	girder12	113	3	0.04	2.4
	girder13	3230	3	0.03	13.2
	girder14	4000	2	0.01	7.1
	girder15	4005	2	0.01	13.3
	girder16	2715	2	0.02	17.4
	girder17	1051	2	0.02	12.6
	girder18	32	3	0.04	12.3
	girder19	89	3468	27.9	22.3
	girder21	84	3269	23.3	21.3
	girder22	98	4	0.04	1.1
	girder23	3511	2	0.02	16.2
	girder24	5351	3	0.02	3.9
	girder25	8368	3	0.01	3.5
	girder26	4377	4	0.03	10.9
	girder27	1034	5	0.05	10.6
	girder28	95	9	0.13	7.1
	girder29	149	2230	20.6	27.9
Bridge 9	slab	5	2768	0.75	0.2
	pier11	181	1	0.01	16.3
	pier12	1161	1	0	8.1
	pier21	278	1	0.01	25.2
	pier22	258	1	0.01	25.0
	pier31	457	1	0	14.6
	pier32	437	4	0.02	20.3

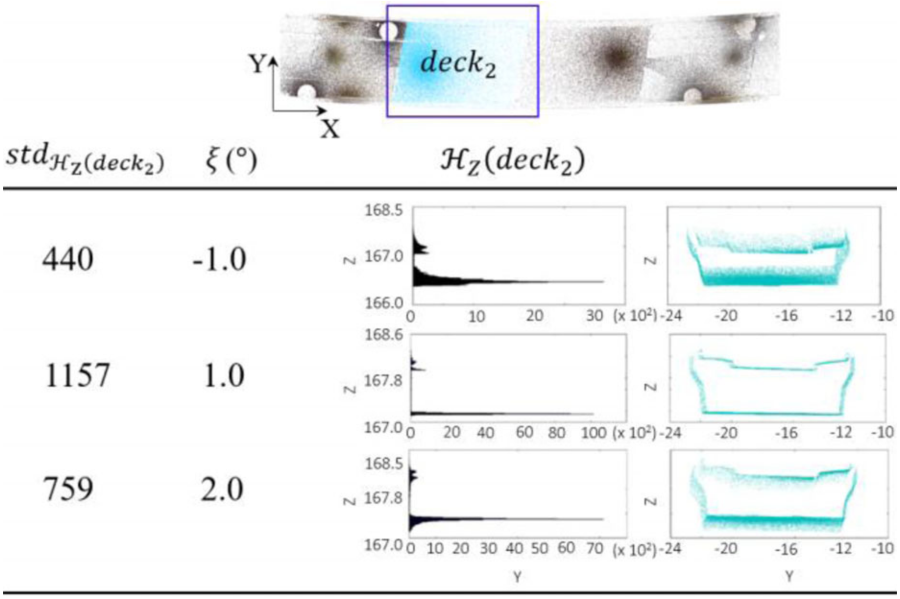


Fig. 20. Best projection search when points are very unevenly distributed (example of $deck_2$ of *Bridge 8*).

Indeed, there could be some extremely nonuniformly distributed points scenarios that our method may not accommodate. Yet, we believe that a carefully planned and elaborately designed scanning process could eliminate these cases. Specially targeted laser-scanning techniques or settings are required for these challenging regions. The method of Laefer and Truong-Hong (2017) can be considered in this endeavor.

It is worth noting that our method is efficient for a certain type of RC bridge, that is, the typical RC slab bridges and beam-slab bridges. Although it is too soon to claim that the proposed method will accommodate all needs, the experiments proved that our method fills some gaps in knowledge and is capable of dealing with some very common and important types of highway bridges.

This method could likely be scaled up for more complicated bridges. Additional procedures can be integrated into our method to detect other elements, such as abutments, bearings, handrails, and so forth. Future work can also be built on Step 4 to detect and segment delicate components in bridges with more complex superstructure geometries (e.g., grid-beams, cross-beams). The method developed for reconstructing gridded steel structures (Gyetvai et al., 2018; Laefer and Truong-Hong, 2017) can be integrated.

5 CONCLUSIONS

Object detection in bridge point clouds remains unsolved. In this work, we presented a novel top-down method for major bridge-component detection in point clouds and tested it on 10 sets of real RC highway-bridge PCD. The validation metrics showed that the method is very reliable, which supports our hypothesis. Given the high performance of our method on real-world bridge point clouds containing defects such as occlusions and sparseness, we contend that there is virtually no human intervention needed to check whether the points are correctly classified in each step.

The contributions of this research are the following:

1. Our method can deal with complex real-world bridge topologies, such as varying elevation and slightly curved horizontal alignment. The method also excelled on *Bridges 8* and *9*, which contain obvious curved horizontal alignments.
2. Our method can handle challenging scenarios, such as occlusions and locally variable densities of points. Although some input is very sparse (e.g., *Bridges 4, 6,* and *8*) due to long scan distances, our method still achieved quite good performance in these point clouds.
3. Our method dramatically reduces computational costs by breaking down a large set of point cloud into subsets. In this way, large-scale object detection efficiency can be significantly improved without sacrificing precision.

However, the proposed method does not intend to be a cure-all. More bridge data with various configurations especially that with different girder and pier cap types should be included and investigated in future studies. This can enhance the statistical power of the method with increased confidence level and reduced margin of error. Then, the method is not suitable for diaphragm bridges whose upstand diaphragms lay on the same level of the integrated lateral beams. It also cannot deal with concrete bridges with complex geometries or steel bridges (e.g., truss bridges). In addition, small girder spacing and severely unevenly distributed points may affect and decrease the detection performance of the proposed method for beam-slab bridges. These regions require additional very-high-resolution scanning. In short, this research indicated that the most important typical RC bridges can be supported using the proposed solution, which can significantly reduce the modeling cost and will accelerate the adoption of BrIM for existing RC bridges. Future planned works will focus on (1) the overcoming of the abovementioned limitations and addressing some of the assumptions; (2) upgrading the algorithm to scale up to more complex bridge systems and detection of more components; and (3) fitting IFC objects to the generated labeled point clusters.

ACKNOWLEDGMENTS

This work is funded by EPSRC, EU Infravation SeeBridge project under Grant No. 31109806.0007 and Trimble Research Fund. We thank them for their support.

DATA AVAILABILITY STATEMENT

The raw bridge PCD can be accessed by the following link https://doi.org/10.5281/zenodo.1233844.

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.

ASCE (2017), *2017 Report Card for America's Infrastructure, Bridges*, ASCE.

Belsky, M., Sacks, R. & Brilakis, I. (2016), Semantic enrichment for building information modelling, *Computer-Aided Civil and Infrastructure Engineering*, **31**(4), 261–74.

Borenstein, E. & Ullman, S. (2008), Combined top-down/bottom-up segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(12), 1–17.

ClearEdge3D (2017), *Structure Modelling Tools*. Available at: https://www.clearedge3d.com/, accessed July 2018.

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), 483–98.

Dodge Data & Analytics (2017), *The Business Value of BIM for Infrastructure 2017*, https://www2.deloitte.com/content/dam/Deloitte/us/Documents/finance/us-fas-bim-infrastructure.pdf.

Dore, C. & Murphy, M. (2014), Semi-automatic generation of as-built BIM façade geometry from laser and image data, *Journal of Information Technology in Construction*, **19**(January), 20–46.

Fanning, B., Clevenger, C. M., Ozbek, M. E. & Mahmoud, H. (2014), Implementing BIM on infrastructure: comparison of two bridge construction projects, *Practice Periodical on Structural Design and Construction*, **20**(4), 1–8.

Gyetvai, N., Truong-Hong, L. & Laefer, D. F. (2018), Laser scan-based structural assessment of wrought iron bridges: Guinness Bridge, Ireland, *Proceedings of the Institution of Civil Engineers: Engineering History and Heritage*, **171**(2), 76–89.

Highways England (1994), Expansion Joints for Use in Highway Bridge Decks. Available at: http://www.standardsforhighways.co.uk/ha/standards/dmrb/vol2/section3/bd3394.pdf, accessed July 2018.

Highways England (2017), Design Manual for Roads and Bridges. Available at: http://www.standardsforhighways.co.uk/ha/standards/dmrb/, accessed July 2018.

Kedar, A. (2016), *SeeBridge Project Document: Automated Compilation of Semantically Rich BIM Models of Bridges: Criteria for Evaluation of Complete SeeBridge System*, https://seebridge.net.technion.ac.il/reports/seebridge-deliverables/.

Kim, M.-K., McGovern, S., Belsky, M., Middleton, C. & Brilakis, I. (2016), A suitability analysis of precast components for standardized bridge construction in the United Kingdom, in *Creative Construction Conference 2016*, Budapest, Hungary, 47–52.

Kokkinos, I., Maragos, P. & Yuille, A. (2006), Bottom-up & top-down object detection using primal sketch features and graphical models, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**(2), 1893–900.

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, http://pr.cs.cornell.edu/sceneunderstanding/nips_2011.pdf.

Laefer, D. F. & Truong-Hong, L. (2017), Toward automatic generation of 3D steel structures for building information modelling, *Automation in Construction*, **74**, 66–77.

Liu, X., Eybpoosh, M. & Akinci, B. (2012), Developing As-built Building Information Model using construction process history captured by a laser scanner and a camera, in *Construction Research Congress*. West Lafayette, IN, United States.

Lu, R. & Brilakis, I. (2017), Recursive segmentation for as-is bridge information modelling, in *Lean & Computing in Construction Congress (LC3)*. Heraklion, Greece, pp. 209–17.

Ma, L., Sacks, R., Kattel, U. & Bloch, T. (2017), 3D object classification using geometric features and pairwise relationships, *Computer-Aided Civil and Infrastructure Engineering*, **33**(2), 152–64, https://doi.org/10.1111/mice.12336.

Marton, Z. C., Rusu, R. B. & Beetz, M. (2009), On fast surface reconstruction methods for large and noisy point clouds, *IEEE International Conference on Robotics and Automation*, **269**, 3218–23.

Network Rail (2015), Bridge List. Available at: https://www.networkrail.co.uk/, accessed July 2018.

Park, H. S., Lee, H. M., Adeli, H. & Lee, I. (2007), A new approach for health monitoring of structures: terrestrial laser scanning, *Computer-Aided Civil and Infrastructure Engineering*, **22**(1), 19–30.

Park, S. W., Park, H. S., Kim, J. H. & Adeli, H. (2015), 3D displacement measurement model for health monitoring of structures using a motion capture system, *Measurement*, **59**, 352–62.

Pătrăucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I. & Haas, C. (2015), State of research in automatic as-built modelling, *Advanced Engineering Informatics*, **29**, 162–71.

Pinheiro, P. O., Lin, T.-Y., Collobert, R. & Dollàr, P. (2016), Learning to refine object segments, *arXiv preprint arXiv:1603.08695*.

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*, **72**(3), 258–68.

Sampath, A. (2010), Segmentation and reconstruction of polyhedral buildings from aerial LiDAR point clouds, *IEEE Transactions on Geoscience and Remote Sensing*, **48**(3), 1554–67.

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

Tang, P., Huber, D., Akinci, B., Lipman, R. & Lytle, A. (2010), Automatic reconstruction of as-built building information models from laser-scanned point clouds: a review of related techniques, *Automation in Construction*, **19**(7), 829–43.

Tatarchenko, M., Dosovitskiy, A. & Brox, T. (2017), Octree generating networks: efficient convolutional architectures for high-resolution 3d outputs, *arXiv preprint arXiv:1703.09438*.

Trimble (2014), Creating a 3D CAD Model From a laser Scanned Point Cloud (Overview). Available at: https://constructible.trimble.com/construction-industry/how-to-create-a-3d-cad-model-from-a-laser-scanned-point-cloud-overview, accessed July 2018.

Truong-Hong, L. & Laefer, D. F. (2015), Quantitative evaluation strategies for urban 3D model generation from remote sensing data, *Computers & Graphics*, **49**, 82–91.

Vo, A. V., Truong-Hong, L., Laefer, D. F. & Bertolotto, M. (2015), Octree-based region growing for point cloud segmentation, *ISPRS Journal of Photogrammetry and Remote Sensing*, **104**, 88–100.

Wai-Fah, C. & Lian, D. (2014), *Bridge Engineering Handbook, Construction and Maintenance*, CRC Press, Boca Raton, FL.

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**, 325–37.

Xu, Y., Tuttas, S., Hoegner, L. & Stilla, U. (2018), Voxel-based segmentation of 3D point clouds from construction sites using a probabilistic connectivity model, *Pattern Recognition Letters*, **102**, 67–74.

Zhang, G., Vela, P. A. & 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*, June 23–25, 2014, Orlando, FL, 406–13.

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.