Automatically extracting surfaces of reinforced concrete bridges from terrestrial laser scanning point clouds

Truong-Hong, Linh; Lindenbergh, Roderik

DOI
10.1016/j.autcon.2021.104127

Publication date
2022

Document Version
Final published version

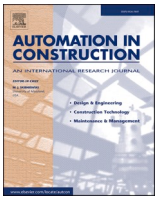
Published in
Automation in Construction

Automation in Construction 135 (2022) 104127

Contents lists available at ScienceDirect

Automation in Construction

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



Automatically extracting surfaces of reinforced concrete bridges from terrestrial laser scanning point clouds

Linh Truong-Hong *, Roderik Lindenbergh

Delft University of Technology, 2628 CN Delft, the Netherlands

ARTICLE INFO

Keywords:
Point cloud
Segmentation
Cell-based segmentation
Bridge deficiencies
Bridge inspection
Bridge modelling
Bridge components
Slab/box bridge

ABSTRACT

Three-dimensional (3D) geometric bridge models play an important role in bridge inspection, assessment, and management. Laser scanning nowadays offers a cost-efficient method to capture dense, accurate 3D topographic data of surfaces of the bridge. However, given the typical complexity of bridges, current workflows using commercial software to construct a bridge model still require intensive labour work. This paper presents a new approach to automatically extract the point cloud of surfaces of structural components of box and slab-beam bridges. The proposed method consists of 3 Parts: (1) point-to-surface, (2) superstructure and (3) substructure extraction. The method uses both spatial point clouds and contextual knowledge to extract point cloud subsets corresponding to surfaces of individual bridge components in a consecutive order from superstructure to substructure. For each bridge component, two levels of extraction are (1) coarse extraction to separate candidate points of the component from the full data set and (2) fine filtering to obtain final 3D points of individual surfaces using cell- or voxel-based region growing (CRG or VRG), followed by a connected surface component (CSC) method. An experimental test on one box-girder and one slab-beam bridges shows that the proposed method successfully extracts all surfaces of bridge components with the lowest *F1-score* of 0.93 based on a point-based evaluation. Moreover, a shape similarity evaluation also shows that discrepancies between extracted surfaces and ground truth ones are no larger than 0.82 for the area overlap ratio and 0.59 degrees for the angular deviation. The proposed method contributes to the automatic generation of 3D geometric bridge models and to give point clouds of individual surface for damage identification.

1. Introduction

Most bridges around the world were built shortly after the second world war and are now in the second half of their service life. These bridges are subjected to deterioration due to excessive usage, overloading, aging, and environmental impact. A recent survey on bridges from 6 European nations [1] showed that most bridges were built in the period 1945–1965 and undergo deterioration, while the American Society of Civil Engineers reported that about 7.5% of 617,000 national bridges suffer from deficiencies while 42% of the bridges is more than 50-year-old [2]. Similar trends were also found in the UK [3] and Japan [4]. Bridge deficiencies have negative impact on daily operation and are costly. To prevent catastrophic collapse, changes in the bridge structures must be monitored, predicted, and reported timely, which can be achieved through bridge inspection and assessment. In those activities, an as-service three-dimensional (3D) bridge model plays an important role because it can be used, for example, to identify structural deficiencies (e.g., deformation or cracks), to track damage propagation, to assess bridge behavior through structural analysis, and to integrate bridge information for bridge management. However, as-service 3D bridge models are rarely available. Such model can be produced from surveying measurements of locations and dimensions of structural elements. A classical surveying campaign using tapes and/or a total station to measure primary dimensions of structural elements is time-consuming and expensive because of the large scale, complex bridge structure. Moreover, as details of surfaces of the structural components are not captured

Abbreviations: 3D, Three dimensional; CAD, Computer aided drawing; mBB, Minimum bounding box; UAV, Unmanned aerial vehicle; TLS, Terrestrial laser scanning; KDE, Kernel density estimation; PDS, Probability density shape; PCA, Principal component analysis; 4NCS, 4-neighbouring cell searching; CRG, Cell-based region growing; CpRG, Cell-patch region growing segmentation; CSC, Connected surface component; VRG, Voxel-based region growing; PM, Proposed method; GT, Ground truth; TP, True positive; FP, False positive; FN, False negative; Comp., Completeness; Corr., Correctness.

* Corresponding author.
E-mail addresses: l.truong@tudelft.nl (L. Truong-Hong), r.c.lindenbergh@tudelft.nl (R. Lindenbergh).

https://doi.org/10.1016/j.autcon.2021.104127
Received 23 August 2021; Received in revised form 29 October 2021; Accepted 31 December 2021
Available online 8 January 2022

sufficiently by classical single point-based surveying instruments, status of structural components cannot be fully assessed, which leads to limited applicability of bridge models based on classical surveying.

Recent development of laser scanning and photogrammetry provides alternative approaches to replace or complement classical surveying methods in capturing bridge geometry, in which laser scanning from a tripod or unmanned aerial vehicle (UAV) are often used [5]. Such mapping systems provide highly detailed and accurate point clouds of component surfaces, particularly in case of terrestrial laser scanning (TLS). Additionally, UAV-based image acquisition is relatively cheap and quick, and a survey of a medium bridge can be completed in hours. However, it requires considerable time to generate a 3D point cloud from overlapping images using structure from motion techniques. Moreover, the quality of photogrammetric point clouds depends on image quality, which is subjected to camera quality, flight planning and environmental conditions. Notably, point cloud-based images have relatively low accuracy with an error budget in the order of centimetres [6], and have a large amount of noise [7]. As a consequence, pre-processing requires a crucial step to remove this noise to make point cloud-based images usable for bridge engineering applications [8]. As such, point cloud-based images are not preferable to use to create an as-service bridge model.

In a current practical pipeline, standalone commercial products, and computer aided drawing (CAD) plug-ins (e.g., Leica Cloudworx, Trimble RealWork and Revit) assist users to process raw point clouds. The point cloud of each member is segmented using built-in tools to separate and fit geometric primitives (e.g., planes, cylinders, and spheres or even irregular shapes) for 3D shape representation. However, these programs target building and facilities sections. For example, ClearEdge3D [9] is specialized to process point clouds of buildings and industrial plants. The software provides tools allowing users to select and extract point clouds of columns or beams by picking 2 end points of the element and then 3D models are fitted. Additionally, Lu and Brilakis [10] reported a processing time for 10 bridges of about 0.59 h per million points when Autodesk Revit was used to create an as-is bridge model. Similarly, Barazzetti [11] addressed large time consumption and costs using the current practice pipeline. Moreover, for large and complex projects like bridges, only experienced practitioners have the ability to handle and process massive point clouds efficiently. This shows that an automatic method should be developed to improve the current work-flow.

Recently, the research community has attempted to develop methods to process point clouds for bridge engineering applications including surface-based damage identification and bridge modelling. For example, Rivero et al. [12] and Truong-Hong and Lindenbergh [13] estimated vertical clearance of a bridge from a point cloud, and Truong-Hong et al. [14] identified surface damage of a bridge deck. Teza et al. [15] estimated the crust of bridge members, Truong-Hong and Laefer [16] estimated surface loss of steel members and Kim et al. [17] and Balado et al. [18] estimated volume loss of masonry bridges. Additionally, in a work on point cloud processing for bridge modelling, Rivero et al. [8] combined a heuristic method with image processing to segment masonry arch bridges. Lu et al. [19] relied on heavy engineering parameters derived from design specifications, and geometric bridge information to extract concrete bridge components (e.g., slab, girders, and piers). Truong-Hong and Lindenbergh [20] preliminarily used spatial information of a point cloud and contextual knowledge to extract components of box bridges, while Yan and Hajjar [21] extracted structural members of steel bridges from a classified point cloud using clustering techniques (e.g., Euclidean and Mean-shift clustering) and RANSAC-based line segmentation.

A common problem of these existing methods is that they are unable to automatically extract point cloud of individual surfaces of each bridge component directly from a point cloud of a complete bridge. This is to be a barrier to automatic point cloud analysis for surface-based damage identification and bridge modelling. Indeed, when point clouds of surface is available, surface damage can be identified directly, while the complexity of the 3D modelling process is reduced significantly because it can be done at component level. Moreover, although some segmentation methods (e.g., [22,23]) can extract point clouds of individual surfaces from single bridge components, it is a non-trivial task to parameterize these methods to obtain desired results [23], particularly most bridge structures are composed of different types, shapes and sizes of components.

To overcome this problem, this paper proposes a new method exploring both spatial point clouds and contextual knowledge to extract point clouds of surfaces of individual bridge components, which can be subsequently used in surface damage identification and 3D geometric modelling. The method consists of 3 Parts: *Part 1: point-to-surface*, *Part 2: superstructure extraction* and *Part 3: substructure extraction*. Moreover, for each bridge component, surfaces are extracted in two levels: (1) coarse extraction to obtain candidate points of the component and (2) fine filtering to determine final points for the surface. As such, as only subsets are processed to obtain surfaces of each bridge component, data complexity and intensive computations are reduced significantly, which makes the proposed method feasible to process a massive point cloud of the bridge, which is typically acquired to the point clouds of the bridge.

2. Related work

Given the complexity of bridge structures, recognizing point clouds of surfaces of individual structural components is a key step toward automatically inspecting surface deficiencies and generating a 3D geometric bridge model. Many methods based on region growing [6], Hough transform [24], and RANSAC [25] have been developed to extract subsets of a point cloud corresponding to the surface of an object. Dimitrov and Golparvar-Fard [26] used region growing based on surface roughness to extract building objects, where the surface roughness of each point was computed from multi-scale neighbourhood points. The method still had some drawbacks, for example over-segmentation and intensive computations. To reduce exhausted computations, Vo et al. [23] introduced octree-based region growing to extract planar surfaces in an urban scene, in which an adaptive octree was used to generate voxels representing objects, while salient features (e.g., normal vectors and residuals) estimated from the points occupied by the voxels were used in a region growing process. Ochmann et al. [27] and Thomson and Boehm [28] adopted a RANSAC framework developed by Schnabel et al. [29] to extract planar surfaces (e.g., walls and floors) for indoor building reconstruction. These methods are efficient to extract building structures but may not directly apply to bridge structures with their complex geometry in terms of shape, dimension, and orientation. As such, this section is restricted to methods aiming at extracting bridge components from a point cloud, while extensive surveys on segmentation and geometric modelling can be found in [30–32].

To investigate the application of point clouds for structural analysis, Stull and Earls [33] used the software Geomagic to create a parasolid model of a three-span, continuous steel I-girder bridge from a terrestrial laser scanning (TLS) point cloud for finite element analysis (FEA). Moreover, in attempting to generate a finite element model of a historic metal bridge, Gyetvai et al. [34] introduced non-parametric regression kernel density estimation (KDE) to determine overall dimensions including width, length and height of a historic metal bridge as well as the distance between lateral beams and struts. However, cross-sections of members were manually created within AutoCAD with support of a plug-in of Leica Cloudworx [35] because of the high complexity of the cross-sections, and limited quality and quantity of the point cloud. Yan et al. [36] employed a voxel grid to subdivide a point cloud into uniform voxels, and the real boundary of the structure was then modified using a grid-based mesh generation method proposed by Schneiders [37]. Moreover, nodes, edges and faces between adjacent elements were adjusted to ensure a continuous mesh.

On extracting structural components of historic bridges, a highlight is the work of Riveiro et al. [8], which used the normal vector of each

point to separate a point cloud into vertical and non-vertical elements. Separation was done by analysing the histogram of elevation angles of the normal vectors in a spherical coordinate system. Subsequently, a voxelization model was employed to cluster points of spandrel walls using voxel connectivity, and to extract non-vertical elements like arches, pathway and parapets based on minimum/maximum elevations of the voxels in each row and column. In this work, the voxelization model was also used to filter noisy points and subsequently to cluster points of piers based on voxel connectivity. Although the method showed efficiency to extract masonry bridge components, the result heavily depended on the voxel size. Pan et al. [38] proposed a super-voxel structure and global graph optimization to segment a point cloud derived from UAV images of historic bridges. Rule-based classification based on dimensions and orientations of segments, was used to label the segments as the components of the bridge. The proposed method could recognize decks, fences, and bases with an accuracy of 0.83 in terms of the number of segments, and the authors addressed those results of the surface extraction rely on the density of the point cloud. However, the results are generally insufficient to fully describe 3D bridge information.

To create a 3D model of a pier cap, Walsh et al. [39] used a region-growing technique to extract point clouds of individual surfaces, and a complete 3D model was obtained by modifying fitting planes to their intersections. However, this work cannot detect edges or small surfaces, and the data points of the pier cap were manually extracted from one of the bridges considered. Based on the principles of region growing segmentation, Truong-Hong et al. [14] introduced cell-based region growing (CRG) segmentation to extract bridge decks (or a road surface on the bridge) to identify surface damage. In this work, the point cloud of the bridge was decomposed into 2D horizontal cells and kernel density estimation based on the z-coordinates of the data points in each cell was used to extract points of local planes of bridge components. The point clouds of the road surface and sidewalks were then obtained through CRG. Although this work reported high accuracy in extracting the components in an efficient way, it was limited to extracting bridge deck and footpaths. Zhang et al. [40] detected planar patches of structural elements of a bridge involving three main steps: (1) determining linear dependency between points, (2) clustering points based on their linear subspace and (3) extracting planar points through singular value decomposition. This method is computationally intensive because features of all points were required to compute. The resulting accuracy of planar extraction accuracy is about 78% (31 planes extracted from a total of 40 planes) with normal deviations of 0.11 ± 0.11 degree, which is still on the low side for application in practice.

Recently, Lu et al. [19] sliced a point cloud of a bridge in longitudinal direction and classified the slices into deck and pier assemblies based on slice height. For the pier assembly, it was sliced in transverse direction to detect pier areas and deck assembly, and subsequently the surfaces of the pier cap or the bottom of the deck were extracted based on normal vectors having a near vertical orientation. These surfaces were used to filter points of the deck and pier. For the desk points, a histogram was applied to extract girder points after aligning a bounding box such that it edges were parallel to the axes of the coordinate system. Although a high accuracy of component extraction was reported, the method relied on a wide range of input parameters including bridge engineering parameters and bridge design rules. The method may give false detection results when irrelevant objects (e.g., trees) around the bridge cause incorrect slice classification. The proposed method was suitable for a straight bridge, but it is required that the input data set is oriented such that it is parallel to the direction of one of the global coordinate system axes. For bridge with the vertical curve, the accuracy of point classification was reduced by about 10%. Similarly, Yan and Hajjar [21] proposed a heuristic method to extract point clouds of components of steel girder bridge. In this method, the point clouds were classified as points above the bridge and the ground, in which the position of the scan stations were priority known. The data points of the bridge deck and side barriers/parapets were extracted from the data points above the bridge. Subsequently, RANSAC-based line segmentation was used to cluster girder points and a mean-shift algorithm was used to split cross-frames. Connected component-based clustering were clustered the substructure assembly (e.g., abutments and piers) from the remaining points after removing data points of the superstructures. Although points of the bridge components were extracted with an accuracy of 93.3%, the proposed method required that known scanning station positions was cooperated to the point cloud for the point cloud classification task, and the input data points must be down-sampled with even distribution.

Recently, the use of a deep learning framework was investigated to extract and reconstruct bridge models. Narazaki et al. [41] used a multiscale convolutional neural network (CNN) architecture to recognize bridge components including columns, beams and slabs from images. However, the detection accuracy was still low, for example, the precision of column extraction was about 65%, and results were limited to recognize only parts of the bridge. Additionally, Hu et al. [42] proposed a deep learning framework to construct a 3D model of a cable bridge from a photogrammetric point cloud extracted from multiview UAV images and photogrammetric point clouds. Although the method was successful in generating a bridge model, the distance from mesh to cloud up to 1.78 m implies the model may not be suitable for engineering purposes. To leverage UAV-based images for bridge inspection, Perry et al. [43] used a Gaussian Mixture Model and Agglomerative Clustering to cluster and segment photogrammetric point clouds of bridges based on normal vectors of the point clouds, while defects of the bridge surfaces were identified from images using a Black Hat Transform and Canny Edge Detector. Kim et al. [44] divided a point cloud of a bridge into multiple overlapping subsets as input for a PointNet framework for extracting bridge deck and pier. The authors reported that the classification performance of the framework can achieve an accuracy of 95%. Similarly, to extract decks, piers, and abutments of concrete bridges from TLS data, Lee et al. [45] modified an existing dynamic graph CNN approach [46] by using the absolute distance and mean EdgeConv operator to tune the impact of neighbourhood selection. Although bridge decks were well recognized with a precision of 0.98, performance of other parts were still relatively low with a precision of 0.82 and 0.90 for abutments and piers, respectively. Interestingly, Saovana et al. [47] classified 3D point clouds generated from images by using 3D masks back-projected from 2D mask images using exterior camera orientation, in which the images were classified by deep CNN. Although the images were classified with a F1-score of 0.96, the highest F1-score for point cloud classification was 0.82 for the abutment component while the lowest F1-score was only 0.672 for cross-girders. Generally, methods based on machine learning, particularly deep learning, have one or more downsides: high computation time for labelling and training data, required user input to assign segments to components (e.g., a bridge deck or a pier), inconsistent or relatively low accuracy in object detection, and imbalanced learning. Importantly, the above methods all extracted only parts of bridges instead of surfaces for each single structural element, or individual components of the bridge.

In summary, in converting raw point clouds of a bridge into meaningful information for bridge assessment application, point clouds of individual surfaces of each bridge component play a crucial role in both surface-based damage identification and 3D geometric modelling [21]. However, no existing method is able to directly and automatically extract the data points of all structural surfaces from an entire point cloud of a modern bridge. Although existing studies have been working on developing methods to automatically extract single surface information from a point cloud of an entire bridge, these methods have still several downsides: (1) applicability for limited number of bridge types [19,21], (2) the requirement of even distribution and orientation of data points [21], (3) the requirement of prior knowledge on the input point cloud and heavy engineering parameters for individual bridges [19,21], (4) extraction of only a limited number of parts or components [19,21,45], (5) inconsistent or low accuracy results in the extraction of bridge components [40,45,47], and (6) computational efforts

[19,40,44,45]. Thus, in an effort to resolve those problems, this paper proposes an efficient and robust method to automatically extracting individual surfaces of structural components of a bridge. The proposed method applies to point clouds of slab, slab-beam, or beam-box bridges without requiring any knowledge on the point cloud and data pre-processing.

3. Scope of work

According to the US National Bridge Inventory [48], slab and box beam or girder bridges account for 23.5% of the total bridge population, and is the second largest bridge type on the road network. However, this bridge type accounts for about 17.6% of the total bridge deficiencies slightly more than 13.2% for stringer, multi-beam or girder bridges, which is the largest bridge type. Moreover, in the UK, 43% of the highway bridges are slab bridges and 86% of future bridges will be either a slab or slab-beam bridge [49]. Therefore, as the slab, slab-beam or beam-box bridges are prominent, this study focuses on extracting components of this bridge type from laser scanning point clouds.

Components of a bridge can be classified as superstructure and substructure (Fig. 1). The superstructure consists of components (e.g., girders, slabs, parapets) above the bearings, while the substructure is components (e.g., piers and abutments) below the bearings. The superstructure elements are often distributed in a horizontal plane along the longitudinal direction of the bridge, while the substructure components are distributed in lateral and vertical direction. When using a laser scanning to acquire bridge geometry, only visible exterior surfaces of the bridge components can be captured. Interior surfaces can be collected by internal scanning, but this is out of a scope of this study. In order to extract the point cloud of surfaces of the bridge components, the following main contextual knowledge is used in this study.

Property 1. *A bridge cross-section often consists of a roadway, sidewalk, and railing. A superstructure is bounded: (i) top surfaces of the superstructure ($S_{supstr.top}$), (ii) bottom surfaces of the superstructure ($S_{supstr.bot}$), and (iii) intermediate surfaces ($S_{supstr.int}$), as shown in Fig. 2. $S_{supstr.top}$ consist of surfaces of a roadway (S_{rw}), road curbs (S_{rc}) and sidewalk (S_{sw}). The roadway, S_{rw}, is the largest surface in terms of area (or size), while S_{sw} are smaller surfaces along the road. The elevation of S_{sw} is often either equal or higher than that of S_{rw}.*

Property 2. The superstructure connects to the substructure indirectly via bearings or directly as in integral bridges. In the former case, bearing stones are often small and cannot be captured due to obstruction of the beams/girders and abutments or piers, but a void space in the vertical direction between the superstructure and substructure is available. In the latter case, the bottom surfaces of the beams/girders directly connect to the vertical surface of the abutments and piers. That implies the $S_{supstr.bot}$ can be used to distinguish the superstructure and substructure because its orientation completely differs from adjacent surfaces of the substructure.

Property 3. *Bridge railings may consist of traffic (P_{tr}) and pedestrian (P_{pr}) railings. The railings are distributed along edges of S_{rw} and S_{sw}, and above S_{sw} (Fig. 1). The distance from the railings to the central line of the bridge is a constant because of constant widths of traffic lanes, sidewalk, and other elements along the bridge. Heights of railings are no lower than* 810 *mm for P_{tr} and* 1000 *mm for P_{pr}* [50]. Notably, although the railings are extracted in this study, details of the railing components are not a main objective of this study. The railings may still contain irrelevant points from adjacent objects, for example a traffic sign.

Property 4. *A bridge substructure includes piers and abutments, and these elements are distributed in transverse and vertical directions (Fig. 2). The pier often consists of a pile cap (P_{plcap}), pier column (P_{prcol}), and pier cap (P_{prcap}) (Fig. 2a). However, P_{plcap} is often below a ground or water level, and the point cloud of this component may not be available. Moreover, P_{prcap} may not explicitly appear in some types of piers (Fig. 2a). P_{prcap} is mainly oriented in the transverse direction while P_{prcol} is oriented in the vertical direction. In the abutment, both ballast and breast walls (A_{baw} and A_{brw}) are vertical surfaces and their lengths ($A_{baw}.L$ and $A_{baw}.L$) are often equal to the bridge width ($B_{br}.W$) (Fig. 2b). Moreover, wing walls (A_{ww}) are usually perpendicular to A_{baw} and A_{brw}.* In general, relative relationships between the abutments and pier and the bridge can be expressed in Eq. (1).

$$\begin{cases} P_{prcol}.L \le P_{prcap}.L \le B_{br}.W \\ P_{prcol}.L \le P_{plcap}.L \\ P_{prcol}.W \le P_{prcap}.W \\ P_{prcap}.H \le P_{prcol}.H \\ A_{baw}.L, A_{brw}.L \le B_{br}.W \end{cases} \quad (1)$$

where suffixes *".L,.W,* and *.H*" are respectively the length, width and height of the components. For the abutment and pier, their dimensions (length, width, and height) are dimensions along a transverse, longitudinal direction of the bridge, and a vertical direction.

Property 5. *Minimum geometric parameters of bridge components as shown in Table 1 are used in filtering real components of the bridge. Beside several values are obtained directly from a design specification and practices, the remain values are selected as following reasons. For example, although different minimum heights of railings, the smallest value as* 0.81 *m was selected to use in this study. In a bridge design, the minimum vertical clearance is* 0.6 *m for passing a waterway and* 4.3 *m for passing a structure* [51]. *Moreover, to protect the bearing, the breast wall must be* 0.5 *m higher than the top of a slope of an embankment ground. As such, $H_{substr,min}$ = 0.6 m is selected to extract the candidate points of the substructure. A minimum component thickness ($S_{cs}.T_0$) is selected based on the minimum slab thickness of* 0.178 *m* [52] *plus a pavement thickness. Finally, as a roadway lateral slope (g_{rw}) often varies* 1.5% - 2.0% [53], *and the sidewalk slope (g_{sw})* is often 1.0% [53]. To overcome critical circumstance, the lateral slopes of both is set of 2.0%.

4. Proposed method

The proposed method for extracting point clouds corresponding to surfaces of bridge components has 3 main parts (Fig. 3). Goal of the

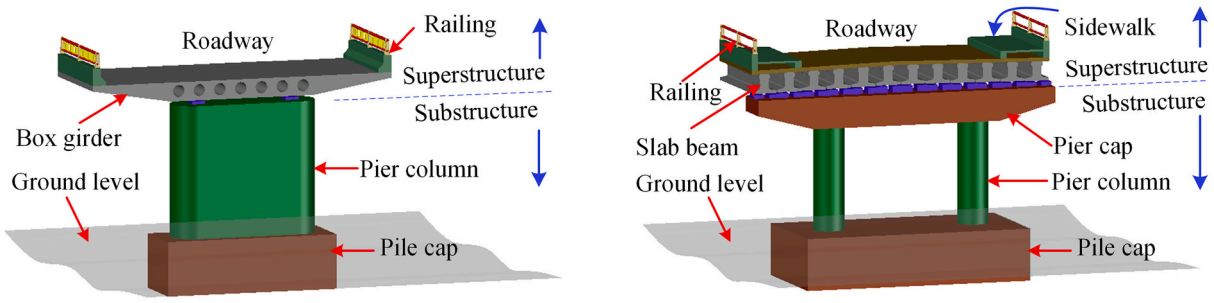


Fig. 1. Typical cross-section of slab-beam or box bridges.

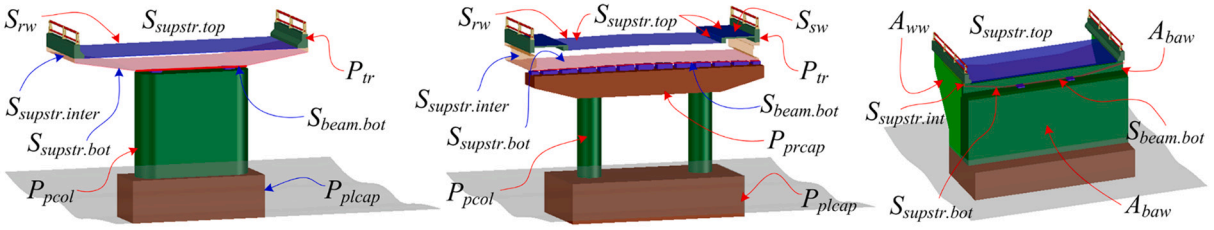


Fig. 2. Surface components of a superstructure.

Table 1
Minimum dimensions of bridge components.

Name	Notation	Values	References
Minimum span length (m)	$L_{sp}.L_{min}$	9.0	[54]
Minimum sidewalk width (m)	$S_{sw}.W_{min}$	1.5	[50]
Minimum railing height (m)	$P_{rl}.H_{min}$	0.81	[50]
Minimum road curb height (m)	$S_{rc}.H_{min}$	0.157	[50]
Minimum substructure height (m)	$H_{substr,min}$	0.6	[51]
Minimum cross-section size(m)	$S_{cs}.D_0$	0.4	Design practice
Minimum component thickness (m)	$S_{cs}.T_0$	0.2	[52]
A slope of a cross-section of a roadway/ sidewalk (%)	g_{rw}/g_{sw}	2.0%/ 1.0	[53]

point-to-surface (Part 1) is to group raw point clouds to clusters representing surfaces of bridge components, which include two steps: (i) coarse and (ii) fine extraction for each type of structures. Coarse extraction is to extract candidate points of structures using point cloud-based features and contextual knowledge, while in the fine extraction, segmentation methods are implemented to obtain final points of individual surfaces of structural components. Subsequently, in Part 2 and 3, superstructure and substructure extraction aim to assign labels to corresponding surfaces. In this method, spatial point clouds and contextual knowledge of bridge components are used to identify surfaces of the bridge component in a sequential order from the superstructure to substructure. Details of the proposed method are presented below.

4.1. Part 1: Point-to-surface

Goal of this part is to extract subsets representing surfaces of individual bridge component from a massive x-, y- and z- point clouds of a bridge. A bridge structure is highly complex regarding the variation in sizes and orientations of the different bridge components. In addition, bridge point cloud is typically massive. Therefore, point-based methods are both time consuming and challenging, given the difficulty in tuning parameters that need to satisfy both local and global geometric variations. Moreover, point clouds of the road surface, sidewalks, bottom surfaces of a superstructure, and of the ground occupy a large portion of the entire data set, and these surfaces are all nearly horizontal. As such, methods based on 2D cells in the xy plane are given priority to process, because the corresponding 2D processing is much simple than 3D voxel-based methods. The process starts with coarse extraction of candidate points by decomposing the point cloud of the bridge into two dimensional (2D) cells in the xy plane and analysing distribution of data points within the cells in a vertical direction. Subsequently, in fine filtering step, cell- and voxel-based region growing segmentation (CRG and VRG) methods are used to obtain the final point cloud for each surface, while connected surface component (CSC) method is implemented to eliminate unreal surfaces of the structures. Details of these methods are presented below.

4.1.1. Quadtree representation

As mentioned in *Property 1*, a superstructure of a box/slab-beam bridge is often bounded by a set of surfaces including top ($S_{supstr.top}$), bottom ($S_{supstr.bot}$) and intermediate ($S_{supstr.inter}$) surfaces (Fig. 2). $S_{supstr.top}$ and $S_{supstr.bot}$, are nearly horizontal planes, while $S_{supstr.inter}$ are mostly inclined or vertical planes. Moreover, surfaces of substructure (abutments and piers) mostly vertical surfaces. To reduce complexity of the entire point cloud of a bridge, particularly for extracting surfaces of a superstructure, a quadtree is employed to recursively subdivide an initial 2D bounding box enclosing the bridge data points ($\boldsymbol{P}_{br} = \{p_i = (x_i, y_i, z_i) \in R^3, i = [1, N_{br}]\}$) into smaller 2D cells ($\boldsymbol{C} = \{c_i\}, i = [1, N_{br}]$) in the horizontal plane (or the xy plane of the global coordinate system) until a termination criterion is reached (Fig. 4). In this study, as a certain required level of details of structural components is the objective, cell size (ce_0) is considered as the terminated criterion, which translates to

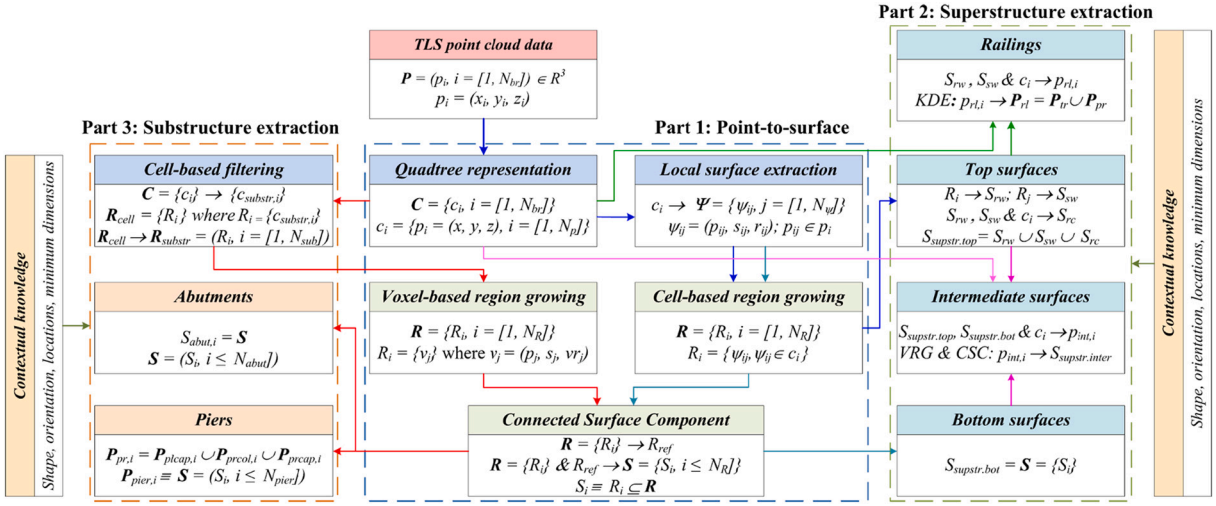


Fig. 3. A workflow of the proposed method.

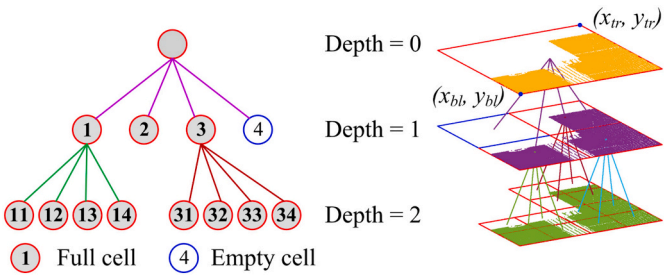


Fig. 4. Quadtree subdivision.

lengths of cells' edges on leaf nodes of the quadtree no smaller than the cell size ce_0. Additionally, each cell gets the property as "*empty*", if the cell contains the number of the points less than a predefined threshold (c_{min_ptc}); otherwise, it is classified as "*full*". Only full cells are considered in next steps of the data processing. Each full cell c_i on the leaf node can be described as a tuple including two opposite corners (x_{bl}, y_{bl}, x_{tr}, y_{tr}), and a list of point indices ($c_i = \{p_i\}, i = [1, N_p]$) where N_p is the number of the points within the c_i.

4.1.2. Local surface extraction

As the 3D data points of the bridge are decomposed into 2D cells ($\boldsymbol{C} = \{c_i\}, i = [1, N_{br}]$) in the xy plane, points in a cell ($p_i \in c_i$ where $p_i = (x, y, z)$, $i = [1, N_p]$) may belong to multiple local surfaces or patches in the vertical direction (Fig. 5a). Moreover, as surfaces $S_{supstr.top}$ and $S_{supstr.bot}$ are nearly horizontal, $p_i \in c_i$ are expected to be distributed in different groups in a vertical direction (or along a z axis). Similar to Armeni et al. [55] who used a density histogram signal to distinguish indoor building components based on void space between surfaces, kernel density estimation is used to identify approximate locations of surfaces, which correspond to local maxima of al probability density shape (PDS), compare also Fig. 5a [56]. In this study, PDS of KDE is accumulated the density at predefined points, in which a so-called Epanechnikov kernel [57] is employed to estimate the density based on the z-coordinates of each point $p_i \in c_i$, and a bandwidth (bw). Given a PDS of KDE, points potentially belonging to a local surface are extracted based on a valley-peak-valley pattern (Eq. (2)), in which the valleys and peaks of the PDS are determined through a second derivative of the PDS (Fig. 5b). By using this simple assumption, a patch ψ_{ij} may possess points of adjacent surfaces, for example, the patches in Fig. 5c still occupy the points of vertical surfaces. Subsequently, indices of the patch are ordered according to the patch elevation (Fig. 5c).

$$\psi_{ij} = \left(p_{ij} \mid PDS.z_{valey,k} \le p_{ij}.z \le PDS.z_{valey,k+1}\right) \text{ where } p_{ij} \subseteq p_i \in c_i \tag{2}$$

where $PDS.z_{valey,k}$ and $PDS.z_{valey\ k+1}$ are coordinates of two consecutive valleys of the PDS.

4.1.3. Cell-based region growing (CRG)

Goal of CRG is to extract point clouds corresponding to surfaces $S_{supstr.top}$ and $S_{supstr.bot}$ from patches within cells in *Section 4.1.1* and *4.1.2*. The method consists of 5 steps (Fig. 6): fit a plane to a patch (*Step 1*), cell-patch region growing segmentation (CpRG) (*Step 2*), patch filtering (*Step 3*), patch-point region growing (*Step 4*) and merging-planar region growing (*Step 5*). *Steps 3* and *4* are to avoid that the patches possess points of adjacent surfaces, while goal of *Step 5* is to resolve under-segmentation due to missing or sparse or noisy data.

Step 1: Plane fitting.

As each patch $\psi_{ij} \in c_i$ is assumed to represent a planar surface, a plane is fitted to points $p_{ij} \in \psi_{ij}$ using principal component analysis (PCA) based on a covariance matrix $cov_{\psi ij}$ (Eqs. (3) and (4)) [58], in which the normal vector n_{ij} of the fitting plane (s_{ij}) is the eigenvector corresponding to the smallest eigenvalue of matrix $cov_{\psi ij}$.

$$cov_{\psi ij} = \sum_{i=1}^{N} \left(p_{ij} - p_{ij,0}\right)\left(p_{ij} - p_{ij,0}\right)^T \tag{3}$$

$$p_{ij,0} = \frac{1}{|p_{ij}|} \sum p_{ij} \tag{4}$$

where $p_{ij,0} = (x_{ij,0}, y_{ij,0}, z_{ij,0})$ is the centroid of the points $p_{ij} \in \psi_{ij}$, and $|\,|$ is the cardinality of a set, which denotes the number of points p_{ij}.

Moreover, a residual value (cr_{ij}) defined as the root mean square distances d_{ij} of the points p_{ij} to the fitting plane, ($s_{ij} = (p_{ij,0}, n_{ij})$ given in Eqs. (5) and (6) is computed, which is considered as a salient feature of the patch. Finally, a patch is described by the tuple $\psi_{ij} = (p_{ij}, s_{ij}, cr_{ij})$.

$$d_{ij} = \frac{\left(x - x_{ij,0}\right)n_{ij}.x + \left(y - y_{ij,0}\right)n_{ij}.y + \left(z - z_{ij,0}\right)n_{ij}.z}{\sqrt{n_{ij}^2.x + n_{ij}^2.y + n_{ij}^2.z}} \tag{5}$$

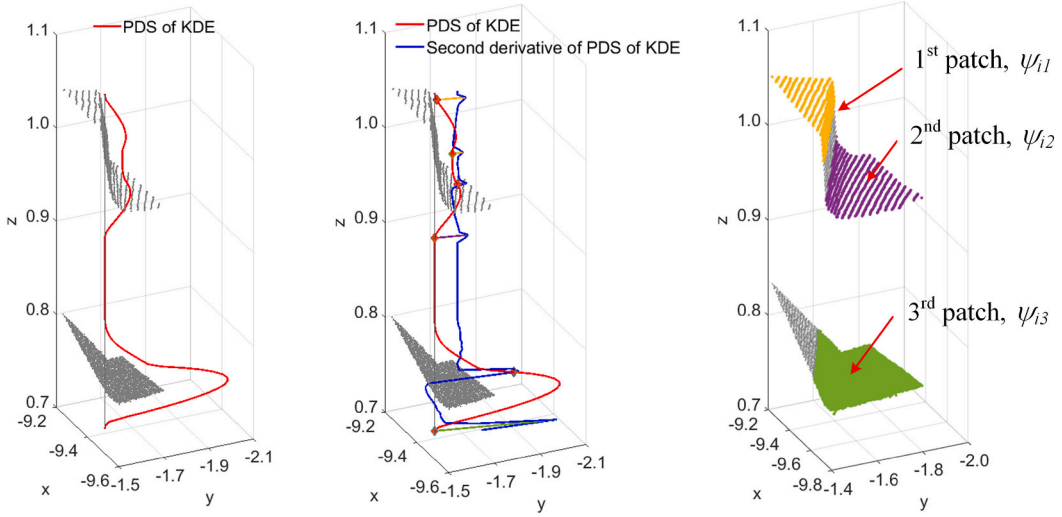


Fig. 5. Extracting data points of patches within a 2D cell.

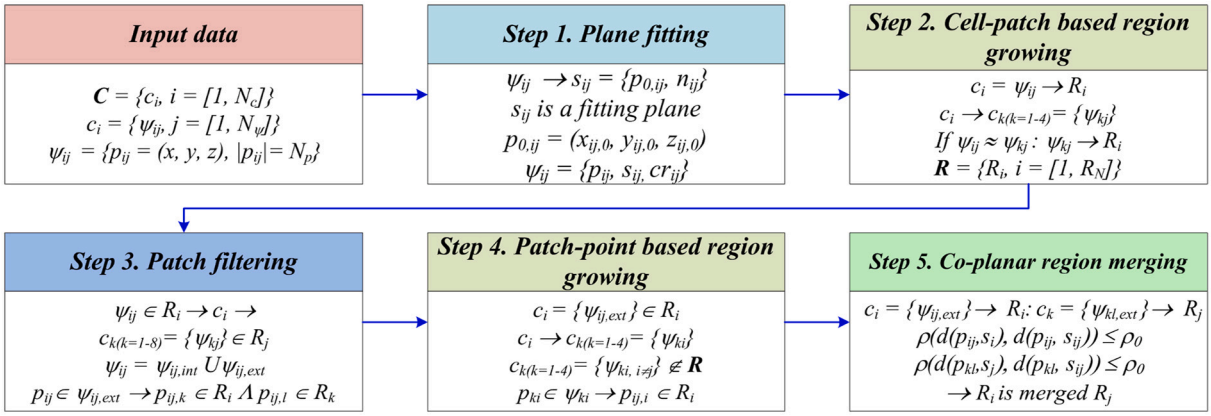


Fig. 6. A workflow of CRG.

$$cr_{ij} = \sqrt{\frac{1}{|d_{ij}|}\sum_{i=1}^{N} d_{ij}^2} \tag{6}$$

Step 2: Cell-patch -based region growing segmentation (CpRG).

CpRG is similar to the region growing segmentation proposed by Rabbani et al. [22], but this implementation uses features of the patch ψ_{ij} instead of single points. The segmentation process starts with an initial seeding patch $\psi_{ij} \in c_i$ ($\psi_{ij} \rightarrow$ a region R_i) having the smallest residual value ($cr_{ij} \rightarrow$ min). Next, adjacent cells $c_{k(k=1\div 4)}$ of c_i are retrieved based on a 4-neighbouring cell searching (*4-NCS*), where the neighbouring cells c_k share an edge with the cell c_i. In this work, only patches $\psi_{kj} \in c_k$ the same index order to $\psi_{ij} \in c_i$ are retrieved, where indices of patches within the cell can refer to Fig. 5c. A patch ψ_{kj} is added to a region R_i if deviations between the features of ψ_{ij} and ψ_{kj} satisfy Eq. (7). Moreover, the patch ψ_{kj} is only considered as a seeding patch for a next iteration if its residual cr_{kj} is no larger than a predefined residual threshold (cr_0). The region R_i is incrementally growing until no more patch can be added. The segmentation process iteratively groups patches into the same region until all input patches are examined. A result of the segmentation is a set of regions $\boldsymbol{R} = (R_i, i = [1, N_R])$ where $R_i = (\psi_{ij} \in c_i)$. Notably, although a cell c_i can occupy multiple patches ψ_{ij}, only one patch of each cell is segmented at this step.

$$\begin{cases} n_{ij}, n_{kj} \leq \alpha_{c0} \\ d\left(p_{kj,0}, s_{ij}\right) \leq d_{c0} \end{cases} \tag{7}$$

where $\angle n_{ij}, n_{kj}$ is the angle between the normal vectors of s_{ij} and s_{kj}, $d(p_{kj,0}, s_{ij})$ is the Euclidean distance from $p_{kj,0} \in s_{kj}$ to s_{ij}, and α_{c0} and d_{c0} are respectively a predefined angle and distance threshold.

Step 3: Patch filtering.

As the 2D cells represent the point cloud of a bridge, data points within the cell or patch may belong to multiple adjoined surfaces. That can cause over- or under-segmentation in results from CpRG (Fig. 7a and b), in which patches on a region boundary occupy points of adjacent surfaces (Fig. 7c). To solve this, patch filtering starts with patches on the

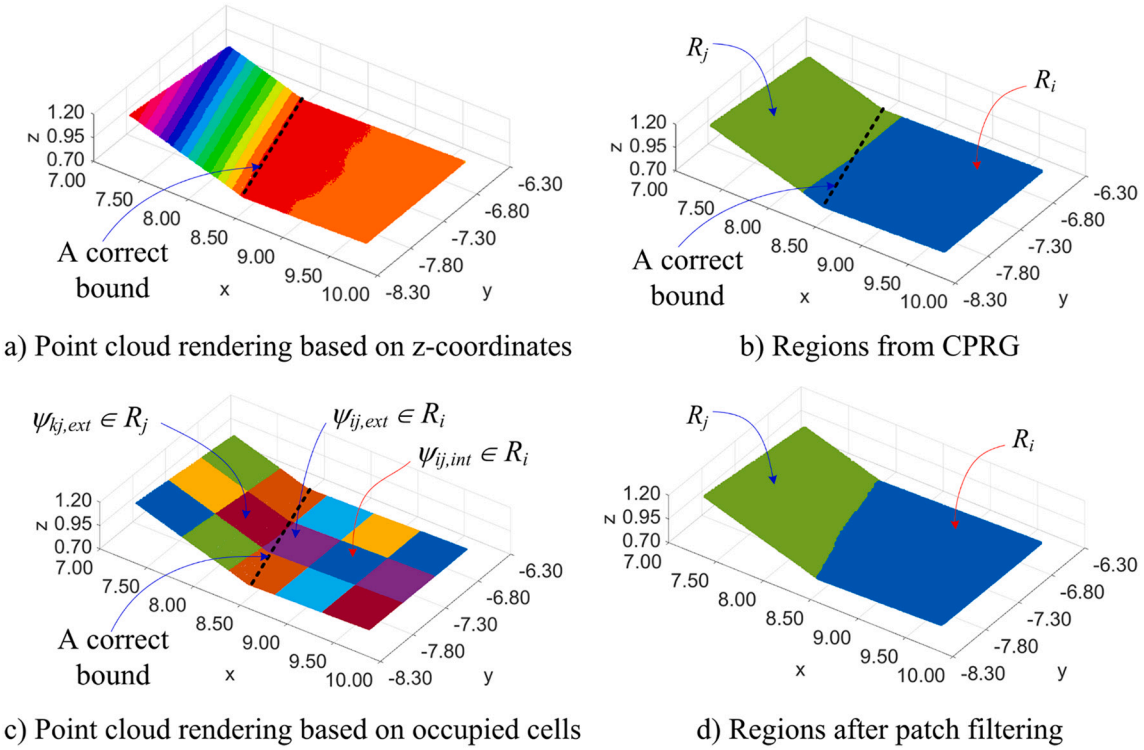


Fig. 7. Illustration of the patch filtering algorithm.

boundary of a specific region and data points of each patch are then re-assigned to any region connected to the patch, if the distances from the points to the region are no larger than the distance threshold d_{c0} [14]. If multiple regions get connected to the patch, the assignment ordered in priority from large to small regions in term of the number of patches in a region. The distance threshold d_{c0} dominates the quality of patch filtering.

First, a patch $\psi_{ij} \in R_i$ is classified as an interior ($\psi_{ij,int}$) and exterior ($\psi_{ij,ext}$) patch, in which $\psi_{ij,int}$ has all neighbouring patches derived based on *4-NCS (Step 2)* belonging to the region R_i. Second, for the exterior patch $\psi_{ij,ext}$, the neighbouring patches ψ_{kj} are subsequently divided into groups: (i) $\psi_{kj,int} \in R_i$, and (ii) $\psi_{kj,ext} \in R_i$ and/or R_j, in which R_j is an adjoined region of R_i (Fig. 7c). Third, if $\psi_{kj.ext} \in R_j$ exist, *4-NCS* (*Step 2)* is used to retrieve neighbour interior patches $\psi_{lj,int} \in R_j$. Next, local surfaces s_k and s_l of the regions R_i and R_j are respectively estimated from the points $p_{kj} \in \psi_{kj,int}$ and $p_{lj} \in \psi_{lj,int}$ using Eqs. (3)–(4). Finally, the points $p_{ij} \in \psi_{ij,ext}$ are classified into two sub-groups: $p_{ij,i} \in R_i$ and $p_{ij,j} \in R_j$ by two consecutive conditions in Eq. (8). Once the points within $\psi_{ij,ext}$ are completely classified, the fitting plane of $\psi_{ij,ext}$ is updated. Notably, if $\psi_{kj.ext} \in R_j$ are not available, there is no need to assign points to the region R_j.

$$\begin{cases} p_{ij} \rightarrow p_{ij,i}\ \epsilon\ R_i\ if\ d(p_{ij}, s_k) \le d_{c0} \\ p'_{ij} = p_{ij} \backslash p_{ij,i} \rightarrow p_{ij,j}\ \epsilon\ R_j\ if\ d\left(p'_{ij}, s_l\right) \le d_{c0} \end{cases} \quad (8)$$

where $p_{ij,\,i}$ and $p_{ij,\,j}$ are points of the regions R_i and R_j after filtering, $d(p_{ij}, s_k)$ and $d(p_{ij}', s_l)$ denote Euclidean distances from the points p_{ij} to s_k, and the points p'_{ij} to s_l, and d_{c0} is the distance threshold.

Step 4: *Patch-point-based region growing.*

In *Steps 2* and *3*, the processing is done based on a specific patch $\psi_{ij} \in c_i$ where j is a constant. However, in practice, because of the complexity of the structure and occlusions, a surface of a structure element may consist of different patches $\psi_{ik} \in c_i$ with different values of k. Therefore, a region R_i may still miss points $p_{ik} \in \psi_{ik(k \neq j)}$ that have not been examined in *Step 2*. To overcome this issue, the cell c_i occupied the $\psi_{ij,ext} \in R_i$ search its neighbouring cells $c_{k(k=1\div4)}$ by using *4-NCS*, in which all patches of the cells c_k are not assigned to regions **R**. Next, a subset of points $p'_{ki} \subseteqq p_{ki} \in \psi_{ki}$ is assigned to R_i if distances $d(p'_{ki}, s_i)$ satisfy the first condition in Eq. (8). Moreover, ψ_{ki} is considered as a new exterior patch of the region R_i if the area of the 2D convexhull of the points p'_{ki} projected on its fitting plane is more than 75% of the cell area (ce_0 x ce_0). This threshold is empirically selected. For each cell c_k, the process is iteratively checking all $\psi_{ki} \in c_k$ but is terminated if any ψ_{jk} is added to *the region* because no co-planar surface is available in the cell. Finally, patch-point-based region growing is iteratively searching new points of R_i until all $\psi_{ij,ext} \in R_i$ are checked.

Step 5: Co-planar region merging.

In practice, due to missing data and/or noisy data, and/or surface defects, a surface of a structural element can be segmented into multiple regions, which need to be merged to obtain a complete surface. In reality, the surface of a component is continuous and homogeneous, which implies two adjacent regions are considered in a merging process, if an adjoined area of two regions has deviations of points to a fitting plane similar to ones of the adjoined regions. Co-planar region merging starts with a region R_i and checks if an adjacent region R_j can be merged (Figs. 8a and 9a). The algorithm retrieves boundary patches $\psi_{ij,ext} \in R_i$ and $\psi_{kl,ext} \in R_j$, and neighbouring patches $\psi_{kl} \in R_j$ for $\psi_{ij,ext} \in R_i$ and $\psi_{ij} \in R_i$ for $\psi_{kl,ext} \in R_j$ are respectively extracted using a window search (*sw*), where distances between centers of cells containing these patches in x and y directions are no larger than *sw* (Figs. 8b and 9b). In this implementation, $sw = 3ce_0$ (ce_0 is a cell size) is used, which allows the gap between R_i and R_j to be up to $2ce_0$. Notably, this value can be adjusted based on quality and quantity of the data. Next, the local planes s_{ij}, s_{kl} and s_{ik} are respectively fitted to the points $p_{ij} \in \psi_{ij,ext}$ and $p_{kl} \in \psi_{kl,ext}$, and $p_{ij} \cup p_{kl}$ using PCA (Eqs. (3) and (4)). The regions R_i and R_j are considered as parts of the surface of the same bridge component, if distributions of distances $d(p_{ij}, s_{ij})$ and $d(p_{ij}, s_{ik})$, and $d(p_{kl}, s_{kl})$ and $d(p_{kl}, s_{ik})$ are similar. In this implementation, correlation between pairs of distances is used to measure similarity. The regions R_i and R_j are merged if their similarity is larger than a predefined similarity threshold (ρ_0) of 0.9. This value was empirically selected in this study, compare also Figs. 8c and d, and 9c and d.

4.1.4. Voxel-based region growing segmentation (VRG)

In fact, orientations of surfaces of bridge components are arbitrary in 3D space, and a simple technique, for example KDE, is only efficiently to extract candidate points of local surfaces, where these surfaces are nearly parallel to the xy, yz or xz plane. To handle the complexity of such data sets, voxel-based region growing segmentation (VRG) is introduced to extract surfaces of bridge components based on 3D voxels, in which the hypothesis of VRG is similar to that of CRG. VRG uses an octree to decompose an initial bounding box enclosing a point cloud into smaller voxels, similar to octree-region growing segmentation as proposed by Vo et al. [23]. However, differing from Vo et al. [23] who used an adaptive octree to decompose a point cloud, this method uses a pure octree [59], where voxels on leaf nodes are all the same size, and have edges' length equal to a predefined voxel size threshold (ve_0). A voxel is classifed as "*full*" if it contains at least a predified number of points ($v_{\min\_ptc}$); otherwise, a voxel is "*empty*". Only "*full*" voxels are used in the further process.

The points within a full voxel on a leaf node are assumed to represent to a plane. This plane s_i is fitted using PCA (Eqs. (3) and (4)) and a residual vr_i is computed according to Eqs. (3)–(6). This allows to describe

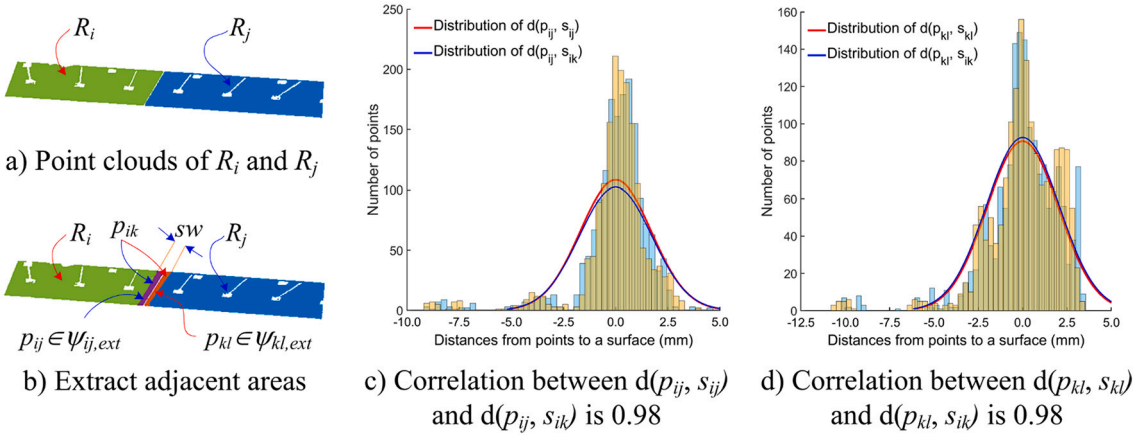


Fig. 8. Two adjacent regions has high correlation between residuals of fitting planes.

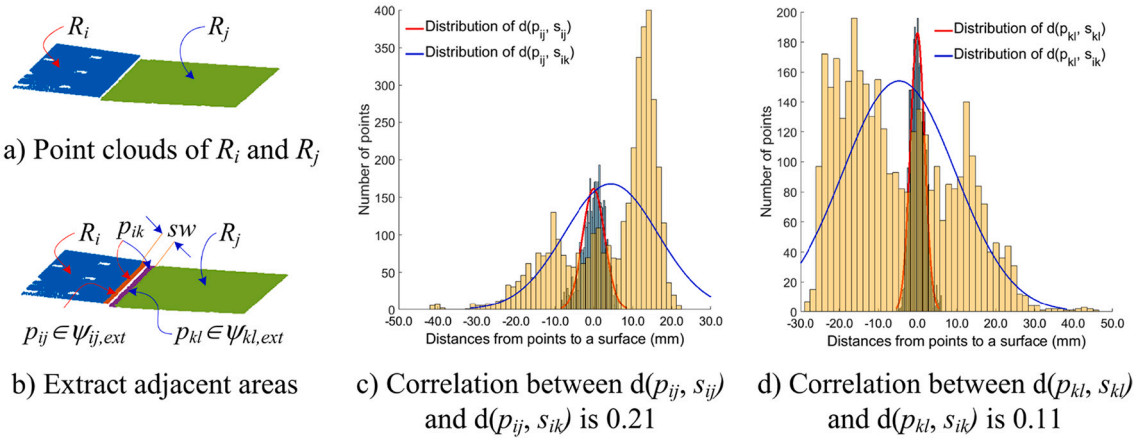


Fig. 9. Two adjacent regions has low correlation between residuals of fitting planes.

each full voxel by a tuple: $v_i = (p_i, s_i, vr_i)$, where $p_i = (x, y, z) \in R^3$ with $i = [1, N_{pi}]$ are points within a voxel v_i, and $s_i = (p_{i,0}, n_i)$ is the fitting plane defined by $p_{i,0} = (x_{0,i}, y_{0,0}, z_{0,i})$ and the normal vector n_i.

Once salient features $v_i = (p_i, s_i, vr_i)$ of voxels are computed, a region-growing mechanism is employed to incrementally group adjacent voxels having deviations of the salient features satisfying certain thresholds expressed by Eq. (7). The growing process is similar to *Step 3.2* in the CRG method, in which the voxel with the smallest residual value ($vr_i \rightarrow min$) is an initial seeding voxel. The resulting segmentation clusters input data $\boldsymbol{P} = (p_i \in R^3)$ into a set of regions $\boldsymbol{R} = (R_i \mid i = [1, N_R])$. A region R_i can be either a planar or curved surface. Similar to Vo et al. [23] or CRG, the voxels on a region boundary may contain points of adjacent regions. To solve this problem, in this implementation, a process similar to the patch filtering (*Step 3*) and patch-point-based region growing (*Step 4*) of CRG method presented in *Section 4.1.2* is adopted, in which voxels are used instead of cells in CRG. Finally, a merging process is also introduced to group adjoined, co-planar regions as shown in *Step 5* of CRG.

4.1.5. Connected surface component (CSC)

CRG and VRG respectively segment points within patches or voxels into a set of regions $\boldsymbol{R} = (R_i \mid i = [1, N_R])$ representing surfaces. However, unreal surfaces, which are not surfaces of bridge components, possibly exist in a set of regions **R**. A CSC method, based on a region growing principles, is introduced to group regions that are part of the structural component. CSC is based on the hypothesis that individual surfaces of a structural component are connected through an intersection edge. CSC starts with an initial reference region R_i known as a region representing a surface of the component and search for an adjacent region R_j that satisfies connection conditions. If the searching is successful, the region R_j is added to the cluster of the structural component. Subsequently, the region Rj is set as a reference surface for a next iteration. The process iteratively checks each pair of regions R_i and R_j to incrementally grow the cluster until no more region to be added. Notably, CSC does not need to check all regions $R_i \in \boldsymbol{R}$ and the initial reference is selected based on features of the component. The connection condition between a pair of regions R_i and R_j is determined as follows.

Starting with boundary patches $\psi_{ij,ext} \in R_i$ and $\psi_{kl,ext} \in R_j$ for the regions R_i and R_j from CRG (or neighbour voxels $v_{i,ext} \in R_i$ and $v_{j,ext} \in R_j$ for the regions R_i and R_j from VRG), neighbour patches $\psi'_{kl,ext} \subseteq \psi_{kl,ext}$ and $\psi'_{ij,ext} \subseteq \psi_{ij,ext}$ (or neighbour voxels $v'_{j,ext} \subseteq v_{j,ext}$ and $v'_{i,ext} \subseteq v_{i,ext}$) are extracted(Fig. 10a). Subsequently, three consecutive conditions are applied to check if R_i is adjoined to R_j. The first condition checks if an adjoined area is available (Eq. (9)).

$$\begin{cases} \psi'_{ij,ext} \neq \varnothing \\ \psi'_{kl,ext} \neq \varnothing \end{cases} \text{or} \begin{cases} v'_i \neq \varnothing \\ v'_j \neq \varnothing \end{cases} \tag{9}$$

The second condition checks if R_i intersects with R_j (Eq. (10)), which is done based on fitted planes $s_i \in R_i$ and $s_j \in R_j$ determined from $p_{ij} \in \psi'_{ij,ext}$ and $p_{kl} \in \psi'_{kl,ext}$, or $p_i \in v'_{i,ext}$ and $p_j \in v'_{j,ext}$ (Eqs. (2) and (3)). Additionally, the third condition checks if two adjoined regions share sufficient boundary (Eq. (11)).

$$\angle n_i, n_j \geq \alpha_0 \tag{10}$$

where n_i and n_j are normal vectors of the planes s_i and s_j, and α_0 is the angle threshold.

$$\begin{cases} L_i \geq L_0 \\ L_j \geq L_0 \\ L_{ij} \geq 0.5min(L_i, L_j) \end{cases} \tag{11}$$

where $L_i = |P_{i1}\ P_{i2}|$ and $L_j = |P_{j1}\ P_{j2}|$ are lengths of the line segment fitting though the projected points $p_i \in \psi'_{ij,ext}$ and $p_k \in \psi'_{kl,ext}$ (or $p_i \in v'_{i,ext}$ and $p_j \in v'_{j,ext}$) onto the intersection line L_{ij}, is the overlap length between L_i and L_j, and L_0 is a predefined length threshold (Fig. 10b). As

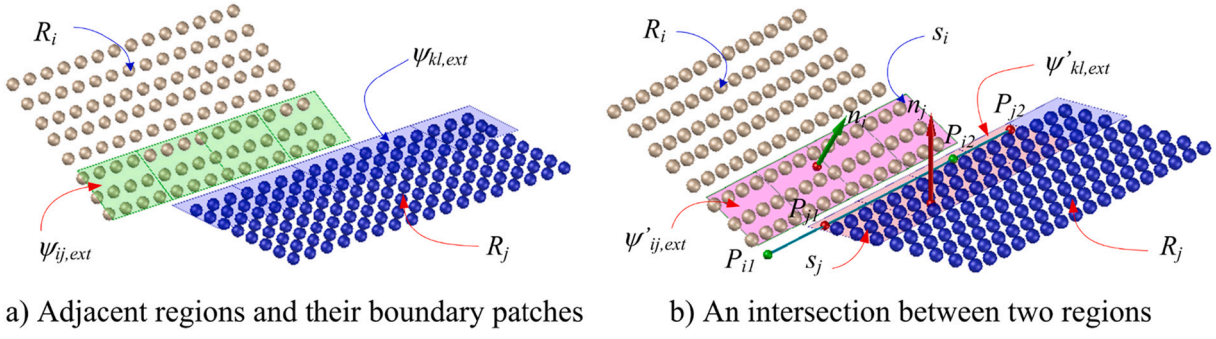


Fig. 10. Example of determining a connection between two adjacent regions based on cells for R_i and R_j derived from CRG.

shown in Fig. 10b, $L_i = |P_{i1}\,P_{i2}|$, $L_j = |P_{j1}\,P_{j2}|$, and $L_{ij} = |P_{j1}\,P_{i2}|$. Notably, L_0 is selected based on dimensions of the structure of interest.

4.2. Part 2: Superstructure extraction

As mentioned in *Property 1*, the superstructure of a box/slab-beam bridge is bounded by top ($S_{supstr.top}$), bottom ($S_{supstr.bot}$) and intermediate ($S_{supstr.inter}$) surfaces (Fig. 2). $S_{supstr.top}$ and $S_{supstr.bot}$, are nearly horizontal planes, while $S_{supstr.inter}$ are mostly inclined or vertical planes. Additionally, $S_{supstr.top}$ involves surfaces of a roadway, road curbs and sidewalks. As such in Part 2, these surfaces are extracted in a sequential order $S_{supstr.top}$, $S_{supstr.bot}$ and $S_{supstr.inter}$ (Fig. 3), in which $S_{supstr.top}$ and $S_{supstr.bot}$ are obtained using CRG segmentation of patches within 2D cells, while VRG segmentation is applied to points between $S_{supstr.top}$ and $S_{supstr.bot}$ to obtain data points of $S_{supstr.inter}$.

4.2.1. Top surfaces of the superstructure

4.2.1.1. Roadway and sidewalk. Based on *Property 1* and when observing the patch distribution within a cell, point clouds of the roadway and sidewalk are mostly located within first patches $\psi_{ij(j=1)} \in c_i$, and these patches are used as input for *Step 2* and *3* of CRG instead of all patches. Moreover, the remaining patches $\psi_{ij(j>1)} \in c_i$ are used as input in *Step 4* of CRG. After merging co-planar regions in *Step 5* of CRG, the resulting segmentation consists of a set of regions $(\boldsymbol{R} = (R_i) \mid i = [1, N_R])$ as shown in Fig. 11a and b.

Additionally, to assign semantic labels to the regions and eliminate regions representing terrains or non-bridge structures, primary features of a region R_i are computed from points $p_i \in R_i$, which include (1) a fitting plane, $S_i = (p_{i,0}, n_i)$, determined by Eqs. (3) and (4), and (2) dimensions (width – $R_i.W$, length – $R_i.L$, and area – $R_i.A$) based on a *2D mBB* derived from projected points of p_i onto S_i. Based on *Feature 1*, the roadway S_{rw} is recognized as the largest region (Eq. (11)), and subsequently a bridge central line (L_{bCL}) is estimated. For a horizontal straight bridge, a parametric line $L_{bCL}(P_0, t_{tr})$ is determined, where the point P_0 is a centroid of the *2D mBB* and the directional vector t_{tr} is the vector of the longest edge of the *2D mBB* because the bridge width is generally shorter than its length. However, for a horizontal curved bridge, the method proposed by Soilan et al. [60] can be used to determine L_{bCL}, represented by multi-line segments $L_{bCL} = (L_i \mid i = [1, N_L])$ where L_i is a line segment connecting two consecutive points of the bridge center.

$$R_i \rightarrow R_{rw} \equiv S_{rw} \text{ if } R_i.A \rightarrow max \tag{12}$$

Remaining regions ($R_j = \boldsymbol{R} \setminus R_i$) are assigned to the sidewalk (S_{sw}) if R_j satisfies Eq. (12) (Fig. 11c). Although the goal of these conditions is to recognize a region similar to a real sidewalk, these factors are empirically selected to compensate the data quality problems. However, a factor of 0.75 in the second condition is lower than others because the length of the sidewalk S_{rw} often includes an approach road, which is larger than the bridge length.

$$R_j = (\boldsymbol{R} \backslash R_i) \rightarrow S_{sw} \text{ if } \begin{cases} 0.9S_{sw}.W_{min} \le R_j.W \\ 0.75S_{rw}.L \le R_j.L \\ 0.9S_{rw}.z \le R_j.z \end{cases} \tag{13}$$

where $S_{sw}.W_{min}$ is the minimum width of the sidewalk, which is no less than 1.5 m (Table 1) [50], $R_j.L$ is the length of the region R_j joined to $R_{rw} \equiv S_{rw}$, $S_{rw}.z$ and $R_j.z$ are respectively the z-coordinates of the $s_{ij} \cong \psi_{ij} \in R_{rw}$ and $s_{kl} \cong \psi_{kl} \in R_j$, where the ψ_{ij} and ψ_{kl} are neighbouring patches derived from *4-NCS* (*Step 2* of *CRG*).

4.2.1.2. Road curb. A road curb (S_{rc}) is available when the elevation of the sidewalk S_{sw} is higher than one of the roadway S_{rw}. In this case, the road curb S_{rc} consists of both horizontal and vertical surfaces ($S_{rc.hor}$ and $S_{rc.vert}$). However, point clouds of the horizontal surfaces $S_{rc.hor}$ are on the same elevation of the sidewalk S_{sw} and the roadway S_{rc}, and include in segments of these surfaces (*Section 4.2.1.1*). In this step, only points of the vertical surface $S_{rc.vert} \in S_{rc}$ are extracted.

As the road curb vertical surface $S_{rc.vert}$ locates between the sidewalk S_{sw} and the roadway S_{rw}, a candidate cell c_i containing points of a road curb can be: (1) c_i contains $\psi_{ij} \in S_{rw}$ and $\psi_{ik} \in S_{sw}$ (for example Fig. 12a), (2) c_i contains $\psi_{ij} \in S_{rw}$ and has a neighbour cell c_j possessing $\psi_{jk} \in S_{sw}$, and (3) c_i occupies $\psi_{ij} \in S_{sw}$ and has a neighbour cell c_j possessing $\psi_{jk} \in S_{rc}$. Notably, neighbouring cells c_j of c_i are extracted using *4-NCS*. After extracting the candidate cell c_i, point-patch distance is used to extract candidate points $p_{rc,i} \in c_i$ (Eq. (14)) (Fig. 12b).

$$c_i \rightarrow (p_i) \rightarrow p_{rc,i} \text{ if } \begin{cases} sign\big(d(p_{rc,i}, s_{rw})\big)\,sign\big(d(p_{rc,i}, s_{sw})\big). < 0 \\ where\ cos(n_{rw}, n_{sw}). > 0 \\ p_{rc,i} \notin (\psi_{rw} \bigcup \psi_{sw}) \end{cases} \tag{14}$$

where *sign()* denotes signed distances from p_i to local planes $s_{rw} \in S_{rw}$ and $s_{sw} \in S_{sw}$, and n_{rw} and n_{sw} are normal vectors of s_{rw} and s_{sw}. The local planes s_{rw} and s_{sw} determine points of patches ψ_{rw} and ψ_{sw} respectively possessed points of the roadway and sidewalk. The patches ψ_{rw} and ψ_{sw} are selected as follows: Case 1: $\psi_{rw} = \psi_{ij}$ and $\psi_{sw} = \psi_{ik}$, Case 2: $\psi_{rw} = \psi_{ij}$ and $\psi_{sw} = \psi_{kl}$, and Case 3: $\psi_{rw} = \psi_{kl}$ and $\psi_{sw} = \psi_{ij}$. Notably, only active points within c_i are considered.

However, due to noise in the data or surface damage, $p_{rc.i}$ may include outlier points that are un-segmented points of S_{rw} and/or S_{sw} (Fig. 12a). To eliminate these points, robust PCA [56] is employed to remove points having distances to the fitting surface larger than the predefined threshold d_s. The final points of the road curb in the cell are shown in Fig. 12c.

4.2.2. Railing extraction

Based on *Property 3* considering locations of parapets, the algorithm starts to extract cells (c_i) on edges between the roadway (S_{rw}) and sidewalk (S_{sw}), and above S_{sw} (Fig. 13a). Candidate points, $p_{rl,j} \in c_i$ of railings above S_{sw}, are extracted according to Eq. (15) (Fig. 13b).

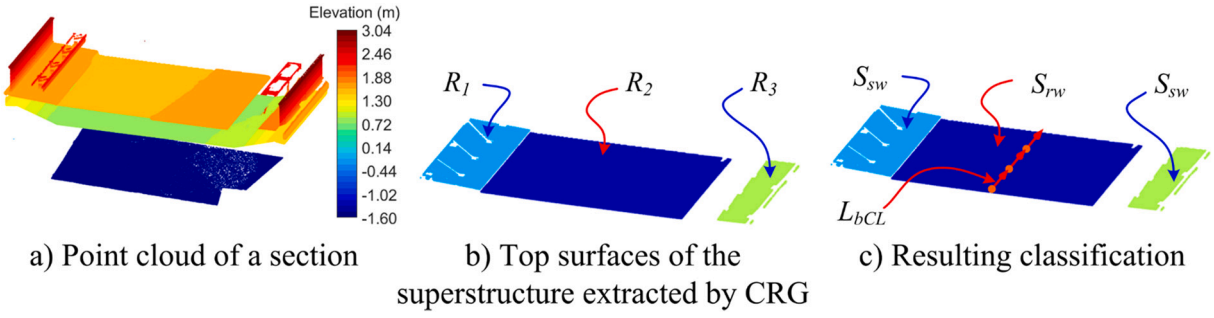


Fig. 11. Illustration of extracting top surfaces of a superstructure.

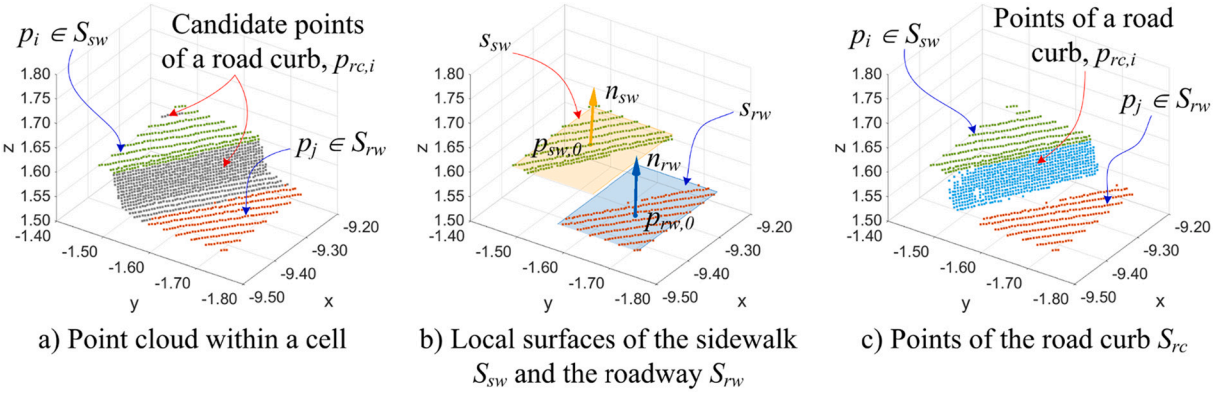


Fig. 12. Extracting data points representing the road curb.

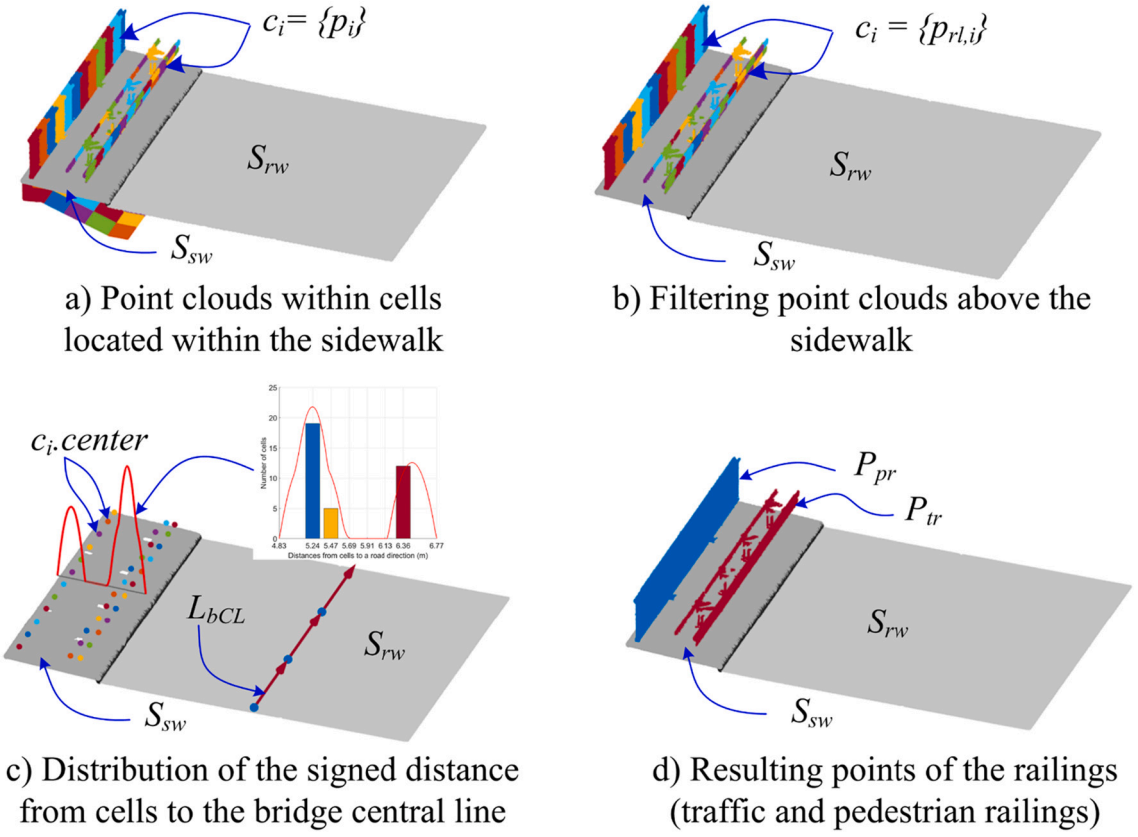


Fig. 13. Extracting point clouds of railings.

$$c_i = (p_i) \rightarrow p_{rl,i} \; if \begin{cases} 0 < d(p_{rl,i}, s_{ij}) \; where \; 0 < cos(n_{ij}, n_z) \\ P_{rl}.H_{min} \le max(p_{rl,i}.z) - p_{ij,0}.z \\ min\_ptc \le |p_{rl,i}| \end{cases} \quad (15)$$

where $s_{ij}(p_{ij,0}, n_{ij})$ is the fitting plane of $\psi_{ij} \in c_i \in S_{sw}$, n_z is a unit vector of the z axis, $P_{rl}.H_{min}$ denotes a minimum height of the railings derived from a design specification (Table 1), $p_{ij,0}.z$ is the elevation of s_{ij} or the sidewalk, and $min\_ptc = 10$ is empirically selected to define if the cell c_i contains the railing points. The first condition extracts points within the cell c_i above S_{sw}, while the second condition ensures that c_i contains railing points. Notably, if c_i does not possess any patch assigned as the sidewalk, the fitting plane s_{kl} of $\psi_{kl} \in c_k \in S_{sw}$ is used, in which c_k is the closest neighbour cell of c_i.

However, cells c_i may not contain points of real parapets, which are subsequently eliminated using clustering based on distances between a cell c_i and the bridge central line (L_{bCL}). For each cell c_i, the centroid of the cell in the xy plane is computed from $p_{par,j}$ (Eq. (16)).

$$c_i = (p_{rl,i}) \rightarrow c_i.center = 0.5([x_{min}, y_{min}] + [x_{max}, y_{max}]) \quad (16)$$

where $[x_{min}, y_{min}]$ and $[x_{max}, y_{max}]$ are respectively minimum and maximum coordinates of the points, $p_{rl,i}$.

Subsequently, signed distances $d(c_i, L_{bCL,2D})$ from $c_i.center$ to a projection of L_{rbCL} onto the xy plane are computed (Fig. 13c). Additionally, KDE generated from the signed distances $d(c_i, L_{bCL,2D})$ is employed to

group c_i into the cluster $\boldsymbol{C}_{rl,i} = (c_i)$ by using a valley-peak-valley pattern (Eq. (2)), in which a bandwidth equal to $1/4S_{sw}.W_{min}$ is used. Subsequently, a length-based filtering of the cluster is applied to eliminate unrealistic railing clusters due to humans or obstacles in the scene during data acquisition. The cluster length, $C_{rl,j}.L$ is the distance between the outmost projection of $c_j.center$ onto $L_{brcent2D}$. Finally, the cluster $C_{rl,j}$ is considered as the railing if $C_{rl,j}.L$ is no shorter than $0.75S_{rw}.L$ (Fig. 13d).

After points are assigned to labels as the roadway S_{rw}, sidewalk S_{sw}, railings P_{pr} and P_{tr}, which belong to top surfaces of the superstructure $S_{supstr.top} = S_{rw}$ U S_{sw} U P_{pr} U P_{tr}, these points are immediately deactivated. Moreover, points $p_i \in c_i$ above $S_{supstr.top}$ are also deactivated because these points do not represent any part of the bridge components (Eq. (17)).

$$c_i = (p_i) \rightarrow p_{deact,i} \text{ if } 0 < d\left(p_{deact,i}, s_{ij}\right) \text{ where } 0 < cosin\left(n_{ij}, n_z\right) \quad (17)$$

where s_{ij} is the plane fitted to points of a patch $\psi_{ij} \in S_{rw}$ or S_{sw}.

4.2.3. Bottom surfaces of a superstructure ($S_{supstr.bot}$)

Similarly, CRG is used to extract points of $S_{supstr.bot}$, in which only the second patches $\psi_{ij\ (j=2)} \in c_i$ are initially used in *Steps 2* and *3* of CRG while other patches $\psi_{ij(j>2)} \in c_i$ are used in *Step 4* of CRG. CRG groups patches ψ_{ij} into a set of regions $\boldsymbol{R} = (R_i) |\ i = [1, N_R]$ (Fig. 14a). However, these regions may include surfaces of other structural elements, for example, the top surface of a pier cap because the bottom surface of the superstructure above the pier cap is not available due to obstruction. CSC (*Section 4.1.4*) is subsequently employed to group regions that represent bottom surfaces of a superstructure. In this method, an initial reference surface is a key in recognizing the superstructure's regions successfully, which is selected as follow.

In the bridge structure, particularly for short and medium spans, heights of cross-sections of the box- or slab-beams are constant. This implies the distance between the bottom surfaces of the beams $S_{beam,bot}$ to the roadway S_{rw} is a nearly constant while distances from other surfaces to the roadway S_{rw} is abruptly change. Based on this feature, $S_{beam,bot}$ is determined from a set of regions $\boldsymbol{R} = (R_i)$ as follow. First, the patch to patch distance $d(s_{ij} \cong \psi_{ij}, s_{ik} \cong \psi_{ik})$ [13] is employed to compute the distance between a region R_i to the roadway S_{rw} through fitting planes s_{ij} of $\psi_{ij} \in S_{rw}$, s_{ik} of $\psi_{ik} \in R_i$, where both ψ_{ij} and ψ_{ik} are occupied by c_i. However, if the patch $\psi_{ik} \in c_i \in R_i$ does not have a corresponding patch $\psi_{ij} \in c_i \in S_{rw}$, a fitting plane s_{kl} of the patch$\psi_{kl} \in c_k \in S_{rw}$ is used to replace s_{ij} in computing $d(s_{ij}, s_{ik})$, in which c_k is the closest cell of the cell c_i determined based on the Euclidean distance between centres of the cells. Subsequently, *KDE* generated from $d(s_{ij}, s_{ik})$ is used to extract the peak containing the patches of $S_{beam,bot}$, where the bandwidth given in Eq. (18) is used (Fig. 14b).

$$bw = 0.5S_{rw}.W^* g_{cr} \quad (18)$$

where $0.5S_{rw}$. W is half of the roadway width because the highest elevation of S_{rw} is at the centre, while g_{cr} is the transverse slope of S_{rw}, which often equals to 2% for drains (Table 1) [50,53].

Additionally, as the area of $S_{beam.bot}$ is larger than those of other surfaces, the patches of $S_{beam.bot}$ are dominated by the largest peak of KDE (Eq. (2)). A region R_i is classified as $S_{beam.bot}$, if more than 90% of the patches of the region R_i have the distances $d(s_{ij}, s_{ik})$ dropped the largest peak (Fig. 14b). The value of 90% is empirically selected to compensate for data quality or any defects of the structures. Finally, CSC is used to group surfaces of $S_{supstr.bot}$ as shown in Fig. 14c.

4.2.4. Intermediate surfaces

As $S_{supstr,inter}$ is located between $S_{supstr,top}$ and $S_{supstr,bot}$, point-patch distance is used to extract the point cloud $p_{supstr.inter} \in S_{supstr,inter}$. The process starts with cells c_i containing patches ψ_{ij} belonging to either $S_{supstr.top}$ or $S_{supstr.bot}$. Candidate points of $S_{supstr.inter}$ within c_i are extracted using Eq. (14), similarly to the procedure for extracting candidate points of road curbs in *Section 4.2.1.2*. In addition, VRG method is employed to extract regions $\boldsymbol{R} = (R_i) \mid i = [1, N_{inter}]$ from the candidate points. Finally, a region R_i is considered as part of $S_{supstr.inter}$, if its length $R_i.L$ in the a bridge longitudinal direction is at least than $0.75L_{sp}.L_{min}$, in which $R_i.L$ is determined based on a *2D mBB* generated from projected points of $p_i \in R_i$ onto the fitting plane of R_i.

4.3. Part 3: Substructure extraction

The substructure consists of abutments and piers, and point clouds of these structures are distributed in vertical and transverse directions but have different geometric shapes (Fig. 1). Substructure extraction consists of two main steps: (1) coarse extraction of candidate points of the abutments and piers through cell-based filtering, and (2) fine filtering of final points of surfaces using VRG and CSC (Fig. 3).

4.3.1. Extracting candidate points of abutments and piers

As point clouds ($\boldsymbol{P}_{substr}$) of the substructure are continuously distributed in a vertical direction, 1D segmentation is implemented to determine cells occupying $\boldsymbol{P}_{substr}$. For each c_i, 1D segmentation (Algorithm 1) is applied to cluster points, in which the distance between two consecutive points in the cluster is no larger than a predefined threshold (Z_0). Notably, only active points, which are not classified as bridge components, are considered. In this study, Z_0 is emperically selected to be equal to $0.5H_{substr,min}$, which is the minimum height of the substructure (e.g., the breast wall or a pier cap) (Table 1).

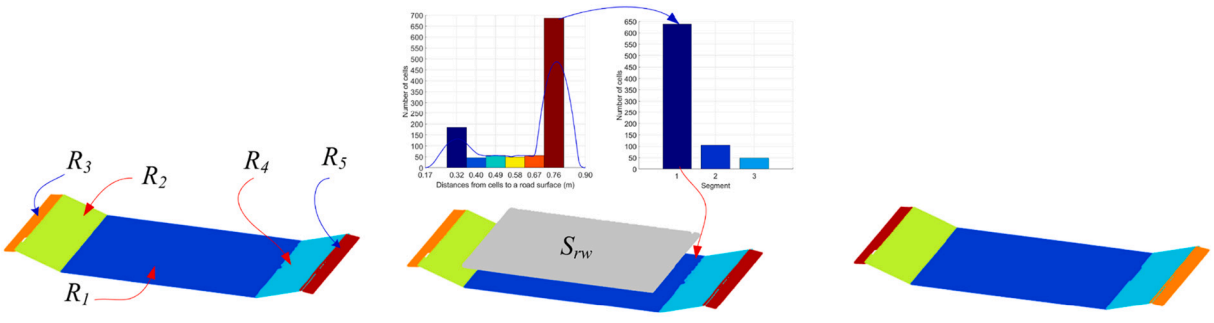


Fig. 14. Filtering surfaces of the bottom surfaces of the superstructure $S_{supstr,bot}$ from CRG.

Algorithm 1: Extracting the point block

Input: data points $(p_i) = (x_i, y_i, z_i) \in c_i$ and a predefined threshold Z_0
Output: $c_i = (cb_{ij}$, where $|cb_{ij}| = N_{cb}$ and $cb_{ij} = (p_{ij})$
Begin
1. Sort p_i according to z coordinates, where $p_i.z < p_{i+1}.z$
2. $j = 1$;
3. $k = 1$ and block $cb_{ij} = (p_{ij} \in p_i)$;
4. while $k < |p_i|$
   if $p_{k+1}.z - p_k.z \le Z_0$ then $p_{k+1} \rightarrow cb_{ij}$;
   else j++ and $cb_{ij} = (p_{k+1})$;
   k++;
return $\boldsymbol{cb} = (cb_{ij})$

A cell c_i is considered to contain points of the substructure if the cluster cb_{ij} satisfies Eq. (19).

$$c_i = (p_i) \rightarrow c_{substr,i} \; if max\left(cb_{ij}.H\right) \ge 0.9H_{substr,min} \tag{19}$$

where $cb_{ij}.H = \max(p_i.z) - \min(p_i.z)$ $\mathrm{cb_{ij}.H} = \max(\mathrm{p_i.z}) - \min(\mathrm{p_i.z})$ is a height of the cluster $\mathrm{cb_{ij}}$, and the factor of *0.9* is used to compensate for either missing data or edge loss in data acquisition [61].

However, in 3D space, cells containing points of irrelevant objects (e.g., road barries) may also satisfy Eq. (19), but such points are sparsely distributed. Cell connectivity clustering, which is similar to connected component labelling [62], is used to obtain cell clusters $\boldsymbol{R} = (R_i, i = [1, N_c])$ where $R_i = (c_{substr,i})$. Subsequently, for each R_i, width – $R_i.W$ and length - $R_i.L$ are respectively lengths of edges of a *2D mBB* estimated from x- and y- coordinates of all points $p_i \in R_i$. Notably, the length of the edge along a transverse direction of a bridge is considered as $R_i.L$. Based on *Feature 4*, the region R_i is considered to represent abutments and piers if it satisfies Eq. (20).

$$\mathbf{R} = (R_i) \rightarrow R_{substr,i} \; if \; R_i.L \ge 0.5B_{br}.W \tag{20}$$

where B_{br}. $W = S_{rw}$. $W + \sum S_{sw}$. W is the bridge width.

Notably, using the condition expressed in Eq. (19), some candidate cells of the substructure may be missed. For example, cells contained points of top and/or bottom surfaces of the pier cap may not satisfy Eq. (19). To overcome this problem, cells located inside the 2DmBB of $R_{substr,i}$ are re-added to the region $R_{substr,i}$. Finally, centroids of the region $R_{substr,i}$ are projected on the bridge central line (L_{bCL}) to determine the abutment and pier clusters, where the outmost clusters are the abutments ($R_{substr,i} \rightarrow \boldsymbol{P}_{abut,i}$) while the others are the piers ($\boldsymbol{P}_{pier,i}$).

4.3.2. Abutment extraction

From a set of candidate points $\boldsymbol{P}_{abut,i} = (p_i, i = [1, N_A])$ of an abutment, VRG is employed to segment $\boldsymbol{P}_{abut,i}$ into a set of regions $\boldsymbol{R}_{abut,i} = (R_i, i = [1,N_{Ai}])$. However, as $\boldsymbol{P}_{abut,i}$ may contain points of other objects (e.g., ground and plants), the CSC method (Section 4.1.5) is subsequently employed to determine whether extracted regions are parts of the abutment. In CSC, a breast wall (A_{brw}) is considered as the initial reference surface (Fig. 2), which is a region R_i satisfying three sequential conditions (Eq. (21)) reflecting three main features of A_{brw}: (i) a length approximattely equal to the bridge width, (ii) a vertical surface, and (iii) connects to the ground level.

$$\begin{cases} 0.9B_{br}.W \le R_i.L \\ \angle n_{Ri}, n_{oz} \approx \pi/2 \\ R_i.z \longrightarrow min \end{cases} \tag{21}$$

where R_i. L is the length of the region in a transverse bridge direction, n_{Ri} and $\mathrm{n_{oz}}$ are respectively the normal vectors of R_i and oz, while $R_i.z$ is the z coordinate of the centre of R_i.

4.3.3. Pier extraction

A pier often consists of a pile cap (P_{plcap}), pier columns (P_{pcol}) and pier cap (P_{prcap}), which are completely different in term of shape, dimension, location and orientation (Fig. 2). P_{prcap} is often located either under ground or water level, and a point cloud of P_{plcap} is not available. P_{prcap} may not explicitly appear in some types of the piers, for example a solid wall pier [63], and if the pier cap appears, it is distributed along a transverse bridge direction. P_{pcol} locates between P_{plcap} and P_{prcap}, and distributes in a vertical direction. Moreover, P_{prcap} mostly consists of planes while P_{pcol} involves planes or curve surfaces or both [63]. Therefore, to reduce the complexity in extracting mixed surface types, the proposed method roughly extracts the points of the pier components, and subsequently VRG and CSC are applied on each pier component to obtain the final points of surfaces.

First, candidate points $\boldsymbol{P}_{pier,i}$ of the pier (Fig. 15) are uniformly sliced in a vertical direction with a predefined slice thickness (ΔZ_t), as expressed in Eq. (22) (Fig. 15b).

$$P_{pier,i} = \left(p_j \in s_j\right), j = [1, N_s] \; where \; s_j = \left(p_j \middle| \; z_j \le p_j.z \le z_{j+1}\right) \tag{22}$$

where z_j and z_{j+1} are the bounds of the layer j.

Next, for each slice s_j, width ($s_j.W$) and length ($s_j.L$) are estimated based on a *2D mBB* generated from the points $p_j \in s_j$ projected onto the xy plane. Subsequently, slice-based region growing segmentation (SRG) is implemeted to iteratively group slices s_j and s_k into the same cluster if deviations of width and length are within the threshold values (Eq. (23)). In this implementatiom, s_j ($j \cong N_s/2$) at the middle of the pier is an intial seeding slice, and a searching window is used to extract adjacent slices s_k ($k = [j - w, j + w]$ and $k \ne j$, $w = 3$ – a window size) of s_j. SRG groups all slices $s_j \in P_{pier,i}$ into clusters $\boldsymbol{SC} = (sc_i = (s_j \mid j = [1,K])$ and $i = [1, N_{sc}]$ where N_{sc} is the number of clusters.

$$\begin{cases} \left|s_j.L - s_k.L\right| \le \Delta D \\ \left|s_j.W - s_k.W\right| \le \Delta D \end{cases} \tag{23}$$

where $\Delta D = 1.5\Delta Z_t$ is the dimensional threshold, which shows an englarge of a cross-section in a vertical direction regarding to an inclined angle (α_{struct}) of the pier column. In bridge design, to optimize load tranfer from the superstructure to foundation, α_{struct} is designed no larger than 45 degrees. As such, dimensions between two consecutive slices differs no larger than ΔZ_t. Moreover, a factor of 1.5 is empirically selected to overcome any problem relating to data qualify (e.g., outlier), which may affect to estimate dimensions of the columns.

Next, the slice cluster sc_i is sequentially classified to pier components ($\boldsymbol{P}_{plcap}$, $\boldsymbol{P}_{prcap}$ and $\boldsymbol{P}_{pcol}$) according to Eqs. (24)–(26) (Fig. 15). However, due to the slice owns a thickness, outmost slices ($s_j \in sc_i$, where $j = 0$ or K) may include data points of adjacent pier components. For example, the outmost slice of $\boldsymbol{P}_{prcap}$ may possess the data points of $\boldsymbol{P}_{pcol}$, and vice versa. To overcome this issue, $p_k \in s_j$ ($j = 0$ *or* $j = K$) of sc_i, are added to the cluster sc_j as input data points for VRG (Fig. 16a and d)

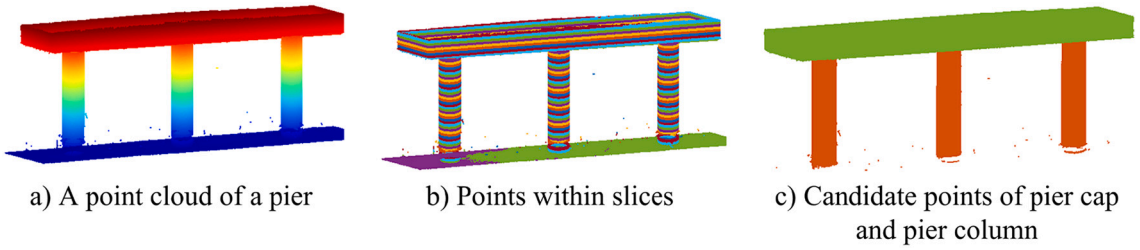


Fig. 15. Roughly extraction of pier components.

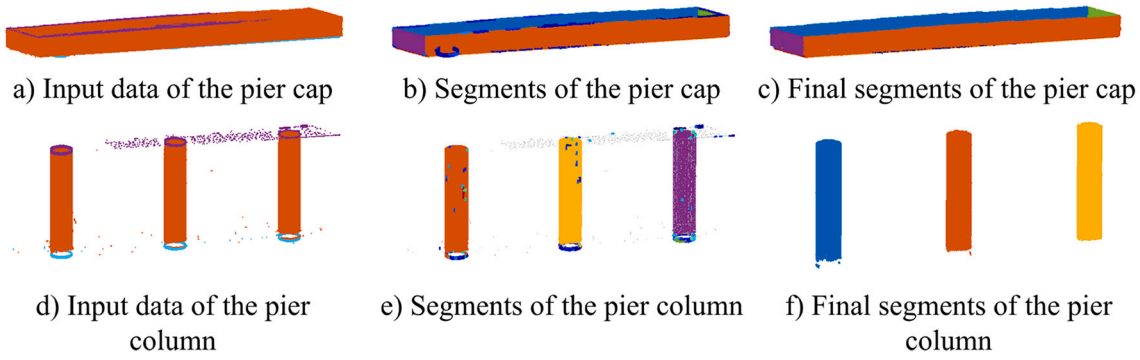


Fig. 16. Extracting point clouds of surfaces of the pier.

$$\boldsymbol{P}_{plcap,i} = (p_i) \in sc_j \; if \begin{cases} sc_j.H \geq 0.9H_{substr,min} \\ sc_j.z_0 < P_{pier,i}.z_0 \\ sc_j.z_0 - P_{pier,i}.z_{min} \rightarrow min \end{cases} \quad (24)$$

$$\boldsymbol{P}_{prcap,i} = (p_i) \in sc_j \; if \begin{cases} sc_j.H \geq 0.9H_{substr,min} \\ sc_j.z_0 > P_{pier,i}.z_0 \\ P_{pier,i}.z_{max} - sc_j.z_0 \rightarrow min \end{cases} \quad (25)$$

$$\boldsymbol{P}_{pcol,i} = \boldsymbol{P}_{pier,i} \backslash \left(\boldsymbol{P}'_{plcap,i} \cup \boldsymbol{P}'_{prcap,i} \right) \quad (26)$$

where $sc_j.H = p_j.z_{max} - p_j.z_{\min}$ and $sc_j.z_0 = 0.5(p_j.z_{max} + p_j.z_{min})$ where $p_j \in sc_j$, $P_{pier,i}.z_{min}$, $P_{pier,i}.z_0$ and $P_{pier,i}.z_{max}$ are respectively the minimum, middle and maximum z coordinates of $\boldsymbol{P}_{pier,i}$, and $\boldsymbol{P}'_{plcap,i}$ and $\boldsymbol{P}'_{prcap,i}$ are the points labeling as the pile cap and pier cap based on results of VRG.

Next, VRG is employed to extract data points of pier components' surfaces in a sequential order $\boldsymbol{P}_{prcap}$, $\boldsymbol{P}_{plcap}$, and $\boldsymbol{P}_{pcol}$. For example, Fig. 16 shows results of VRG for $\boldsymbol{P}_{prcap,i}$ and $\boldsymbol{P}_{pcol,i}$ based on data points derived from Eqs. (25) and (26). VRG segments P_{plcap} and P_{pcol} to a set of regions $\boldsymbol{R} = (R_i,\ i = [1, N_s])$ (Fig. 16b and e). Subsequently, the CSC method is applied to **R** to identify final surfaces for each pier components (Fig. 16c and f). In this work, for P_{prcap} and P_{plcap}, the largest region in terms of area is considered as the initial reference surface, while for P_{pcol}, the initial reference surface is selected from the regions satisfying two consecutive conditions: (i) the deviation angle between the long edge vector of the *3D mBB* of the region (R_i) and n_{oz} no larger than the $\alpha_{struct} =$ 45 degrees, and (ii) the largest region in a term of area.

5. Experiental tests, results and discussions

The goal of this section is to validate the proposed method in extracting point clouds of surfaces of bridge components. To this end, two concrete box and slab-beam bridges were scanned using a terrestrial laser scanner to obtain point clouds for the experimental test. In additional, ground truth derived by manual extraction is used to validate the performance of the proposed method through different validation strategies including a level of location and shape similarity. Finally, based on the experimental tests, impact of quantity and quality of point clouds on surface extraction performance is drawn.

5.1. Bridge description and data acquisition

To demonstrate the performance of the proposed method, two bridges are considered: Bridge 1: a concrete box bridge on the CO16 road, Seßlach, Germany, and Bridge 2: a slab-beam bridge in the UK. The first bridge is a two span bridge with a span length about 10.5 m, and its cross-section consists of 2 traffic lanes 7.0 m wide and 2 sidewalks of about 2.0 m wide per each (Fig. 17a). The bridge was scanned by a Leica ScanStation P20 with a maximum scanning range at 120 m and an angular accuracy of 8 arcseconds in both vertical and horizontal direction [64]. During data acquisition, a sampling step of 6.3 mm at a range of 10 m was used, which translates to an average sampling step of 10 mm on the surface for an average scan distance of 15 m from the ground level. A total of 5 different scan stations (1 station on the sidewalk) was established to maximize data coverage (Fig. 17a). The point clouds of scanning stations were registered by the Leica Cyclone software [65]

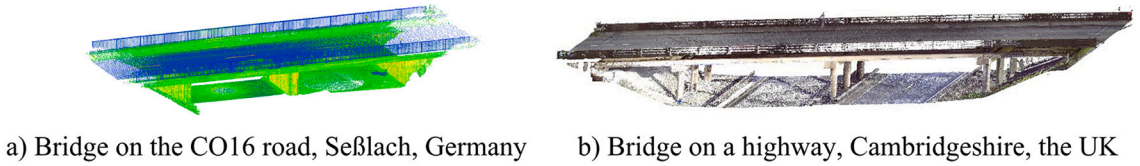


Fig. 17. Point clouds of two bridges used in experimental tests.

with a registration error about 5 mm.

The second bridge is a concrete slab-beam bridge on a highway around Cambridgeshire, UK, which is bridge 1 in a set of 10 highway bridges collected by Lu et al. [66]. This is a 4-span slab-beam bridge with a total bridge length of 59.5 m and a cross-section consisting of a traffic lane of 6.0 m and 2 × 1.45 m wide footpaths (Fig. 17b). The bridge was scanned by a FARO Focus 3D X330 scanner [67] with a sampling step of about 7.67 mm at a measurement range of 10 m, with a total 18 scan stations. The point clouds from different scan stations were registered using FARO Sense software [68].

Irrelevant points corresponding to ground surface and vegetation outside the bounding box of a bridge, and points on vehicles (or moving objects) underneath the bridge were manually removed by using CloudCompareV10 [69]. This process took about 2 min for each bridge. The goal of this removal is to avoid to process unnecessary data points. Notably, the availability of these points does not affect the robustness of the method because (1) points on ground surface, vegetation, and road facilities (e.g., road safety barriers) are still available in the input data, while (2) points on vehicles (or moving objects) are sparse and have low point density, and do not represent any object similar to surfaces of bridge structures. Finally, the point clouds with x-, y- and z- coordinates were exported as input data for proposed method, in which the data sets of Bridge 1 and 2 respectively consist of 28.505 million points (an average point density of 57,130 points/m^2) and 12.670 million points (an average point density of 3183 points/m^2)

5.2. Parameter selection

As the proposed method deploys both spatial point clouds and contextual knowledge of a bridge structure derived from bridge design specifications and practice, input parameters include two sets relating to: (1) minimum dimensions of structural components of the bridge (Table 1) and (2) a point cloud supporting to extract and filter data points of bridge components' surfaces (Table 2). Each input parameter used to extract point clouds of surfaces includes two parts depending on \, first, bridge geometry and, second, data quality. First, minimal dimensions and generic geometric information on components of common bridges were used to determine the first part of the input parameters, that depend on bridge geometry. Second, as data quality varies significantly, depending on object surface, laser sensor, data acquisition, environment, and registration, the second part was empirically selected as a part of the registration error.

In this study, KDE is used to roughly extract data points of surfaces/objects based on a valley-peak-valley pattern. The bandwidth *(bw)* is selected equal to 1/4 distance between two consecutive surfaces/objects, which allows at least one valley of PDS between the surfaces/objects. As such, for extracting local surfaces in cells in Section 4.1.2, bw = 0.05 m is chosen based on the smallest thickness of the structural element, in which $S_{cs}.T_0 = 0.2$ *m* (Table 1) is used. Moreover, for railing extraction, bw = 0.375 m corresponding to $S_{sw}.W_{min}$ = 1.5 m is used to separate pedestrian and vehicle railings.

In principle, CRG and VRG requires at least one initial seeding plane representing a local surface to ensure that a growing process sucessully extract a surface of a structural component. This implies ce_0/ve_0 must be less than 1/3 of the smallest size of the surface. As such, ce_0 = 0.5 m and ve_0 = 0.15 m are selected corresponding to the $S_{sw}.W_{min}$ = 1.5 m and $S_{cs}.D_0$ = 0.4 m (Table 1) [50,53]. However, in Bridge 1, as a shape of the box girder is more complicated, which include various size of surfaces, ce_0 = 0.3 m equals to 1/5 of $S_{sw}.W_{min}$. Additionally, the thresholds c_{min_ptc} and v_{min_ptc} are respectively used to classify 2D cells and 3D voxels as either "*full*" or "*empty*", depending on the number of points within a cell or voxel, respectively. Only full cells or voxels are subsequently used to extract points of surfaces of bridge components by using CRG or VRG. In these segmentation methods, the full cells and voxels are assumed to contain points representing local surfaces of the bridge components. This implies that such cell or voxel must contain at least three non-colinear points. However, to add some robustness to the extraction of local planes, both c_{min_ptc} and v_{min_ptc} are set equal to 10 points.

Table 2
Input parameters for extracting and filtering the point cloud.

Name	Notation	Values
Cell size (m)	ce_0	0.25/0.5
Voxel size (m)	ve_0	0.15
Bandwidth (m)	bw	Vary
Minimum number of points (points)	c_{min_ptc}/v_{min_ptc}	10
Angle threshold (degrees)	$\alpha_{c0} = \alpha_{v0}$	5
Distance threshold (mm)	$d_{c0} = d_{v0}$	10
Residual threshold (mm)	$cr_{c0} = cr_{v0}$	10

CRG is used to extract surfaces of a superstructure, which requires the angle (α_{c0}), distance (d_{c0}) and residual (cr_{c0}) threshold to segment patches within cells. These thresholds are determined as follow. As a cross-section of a roadway is often a bi-linear with a transverse slope (g_{rw}) of 2.0%, and the sidewalk slope (g_{sw}) is often 1.0% (Table 1) [53], three critical cases are used to determine the deviation angle between normal vectors of adjoined cells, in which the largest deviation angle (α_{max}) is 2.3^0 (Fig. 18a). The angle threshold α_{c0} is chosen equal to 5 degrees to cover uncertainty. Moreover, this angle threshold also satisfies a vertical curve bridge, where the maximum tangent grade of 4% used for a highway [53] (Fig. 18b).

The distance threshold (d_{c0}) in CRG is introduced to prevent to group patches belonging to two parallel surfaces, for example, the patches on boundaries of a roadway and sidewalk. In this study, d_{c0} = 10 mm is selected to cover unknow geometries of real surfaces of bridge components, locations of the cells, and quality of a point cloud. For example, for Position 2 of a road surface (Fig. 18a), the distance between two adjacent patches equals to $ce_0 sin\alpha_2/2$ = 10 mm with ce_0 = 0.5 m. Additionally, a residual in a fitting plane (Eq. (5)) depends on not only data quality (e.g., noisy data and a registration error) but also a surface roughness. However, as surfaces of the bridge components are mostly smooth, the residual threshold res_{c0} = 10 mm is selected based a sensor error plus a registration error about 5 mm for a data set of Bridge 1, for example.

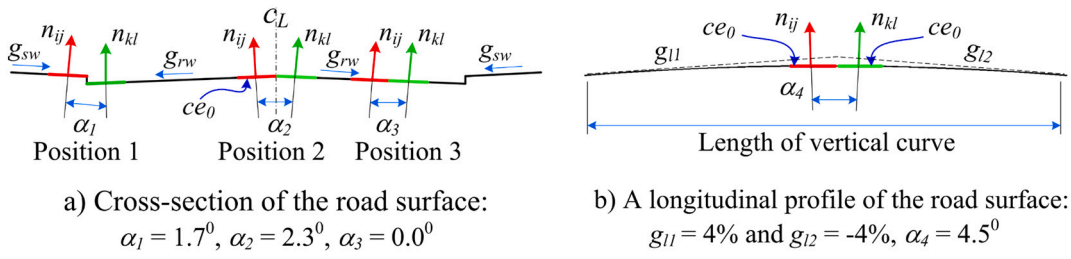
VRG is used to extract points of immediate surfaces ($S_{supstr.int}$) of a superstructure and a substructure, in which those surfaces can be planar and/or curve or both. For the case of planes, e.g., surfaces of $S_{supstr.int}$ and S_{abut}, thresholds (α_{v0}, d_{v0} and vr_{v0}) are set equal to these (α_{c0}, d_{c0} and cr_{c0}) in CRG because these surfaces have similar characteristics. However, as the pier may consist of planar and curve surfaces [6], for example a pier column can involve a semi-circular or circular, ve_0 associated with α_{v0}, d_{v0} and vr_{v0} must be adjusted to accommodate for a general case. Assuming a local surface of each voxel containing points of a circular shown in Fig. 19, a deviation angle - θ and the distance between two adjacent voxels - $d(p_{kl,0}, s_{ij})$ are expressed in Eqs. (27) and (28), while the residual threshold - vr_v is assumed equal to a half of AB$_s$ (Fig. 20) (Eq. (29)). For example, when R = 0.6 m and ve_0 = 0.05 m are used, θ, $d(p_{kl,0}, s_{ij})$ and AB$_s$ are respectively 6.75^0, 4.16 mm and 1.04 mm. Therefore, ve_0 = 0.05 m, $\alpha_{v0} = \theta$ = 6.75^0, d_{v0} = 10 mm and vr_{v0} = 10 mm are used for extracting surfaces of the pier components. Notably, a selection of ve_0 must be adopted to a sampling step, which ensure the voxel possessed sufficient points to estimate a fitting plane.

$$\theta = arcsin\left(\frac{AB_{ch}}{2R}\right) + arcsin\left(\frac{BC_{ch}}{2R}\right) \tag{27}$$

$$d\left(p_{kl,0}, s_{ij}\right) = \frac{BC_{ch}}{2} sin\theta \tag{28}$$

$$AB_s = R - \sqrt{R^2 - \left(\frac{AB_s}{2}\right)^2} \tag{29}$$

where AB_{ch} and BC_{ch} are lengths of chords, and R is a radius of the semi-



a) Cross-section of the road surface: $\alpha_1 = 1.7^0$, $\alpha_2 = 2.3^0$, $\alpha_3 = 0.0^0$

b) A longitudinal profile of the road surface: $g_{l1} = 4\%$ and $g_{l2} = -4\%$, $\alpha_4 = 4.5^0$

Fig. 18. Identifying the angle threshold α_{c0}.

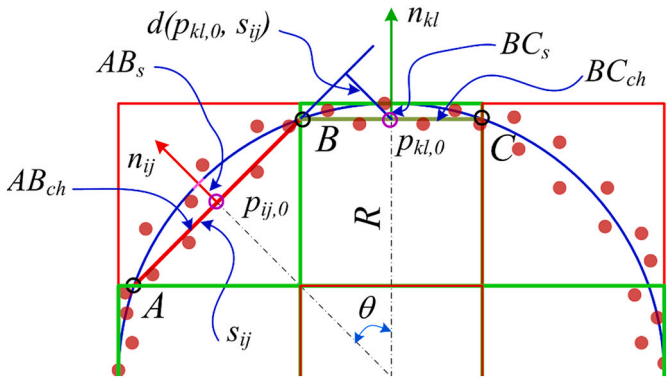


Fig. 19. Illustrate voxels occupied the data points of the cross-section as a circle(*).

Note: This illustration inFig. 19is shown a critical scenario.

circular or circular. The upper bound of AB_{ch} and BC equal to $\sqrt{2}\,ve_0$.

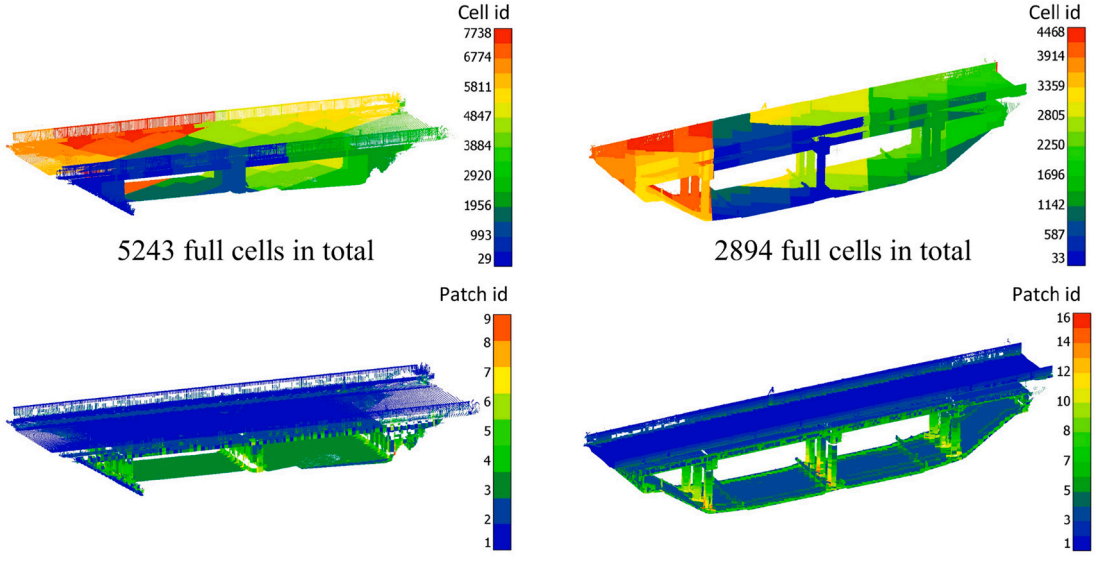
5.3. Results

Using input parameters in Table 2, a quadtree was used to decompose the point clouds of Bridge 1 and 2 into 2D cells in the xy plane Fig. 20). Subsequently, points belonging to patches within the 2D cells were extracted using a valley-peak-valley pattern of KDE (Fig. 20). For each patch, PCA was used to fit a plane through points of the patch and a residual was computed to express a deviation between the points and a fitting plane (Eqs. (3)–(6)), which are then used as input data for CRG.

To segment surfaces of $S_{supstr.top}$, the first patches ($\psi_{ij(j=1)} \in c_i$) were input data for *Step 1* of CRG, while other patches $\psi_{ij(j\neq1)} \in c_i$ were used in *Steps 2–5* to obtain the final points of the regions, where each region represents a surface of $S_{supstr.top}$. Moreover, contextual knowledge about the roadway and sidewalks including size, elevation and orientation was employed to classify the regions as S_{rw} and S_{sw} shown in Fig. 23. Subsequently, from the cell-patch occupied the points of S_{rw} and S_{sw}, data points of the road curbs (S_{cb}) and railings were extracted according to Sections 4.2.1.2 and 4.2.2. The railings were then classified as traffic railings ($\boldsymbol{P}_{tr}$) and pedestrian railings ($\boldsymbol{P}_{pr}$) based on distances from point-cell railings to the bridge central line (L_{bCL}) (Fig. 21).

Similar to extract the $S_{supstr.top}$, second patches $\psi_{ij(j=2)} \in c_i$ were initial input data for *Step 1* of CRG to extract the points of regions, and other patches $\psi_{ij(j>2)} \in c_i$ were used in other steps to obtain the regions $\mathbf{R} = (R_i)$ describing $S_{supstr.bot}$. Moreover, the CSC method was implemented to identify regions of the $S_{supstr.bot}$, where the angle threshold $\alpha_o = 15$ degrees was adopted to determine if two regions assuming as planar surfaces were intersecting while the length threshold (L_0) was chosen equal to $0.9L_{sp}.L_{min} = 8.1$ m ($L_{sp}.L_{min} = 9.0$ m – Table 1) in identifying a connectivity between two adjacent regions. The resulting extraction of $S_{supstr.bot}$ is illustrated in Fig. 22. Additionally, the point-patch distance is used to extract candidate of $S_{supstr.int}$, and final points of surfaces of $S_{supstr.int}$ were obtained using VRG applied to the candidate points, in which the input parameters for VRG in Table 2 were used. However, due to missing data, surfaces of $S_{supstr.int}$ in Bridge 1 appear in two regions (Fig. 22).

After points of the superstructure were extracted and deactivated, cell-based filtering was applied to extract candidate points of abutments and piers (Eq. (19)), in which cells containing candidate points of a



a) Point clouds within cells/patches of Bridge 1 b) Point clouds within cells/patches of Bridge 2

Fig. 20. Using KDE to extract patches for Bridges 1 and 2.

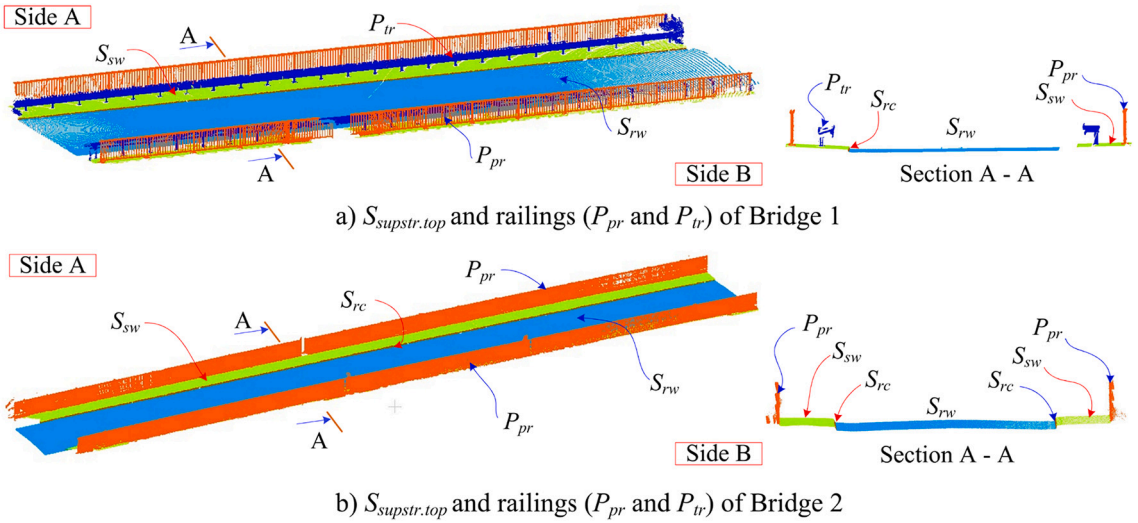


Fig. 21. Extract data points of $S_{supstr,top}$ (a roadway, sidewalks, and curbs) and railings.

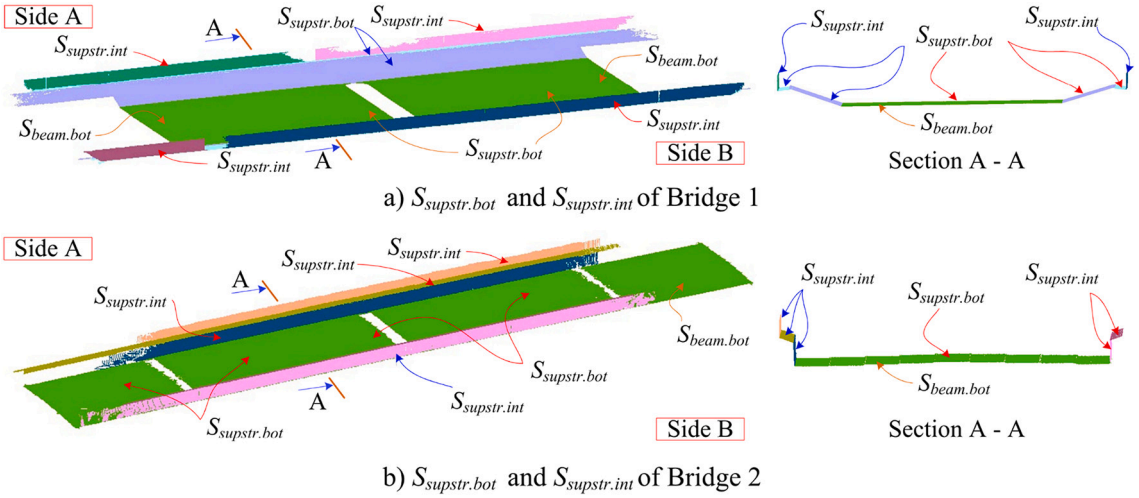


Fig. 22. Extract the points of $S_{supstr.bot}$ and $S_{supstr.int}$ of Bridge 1 and 2.

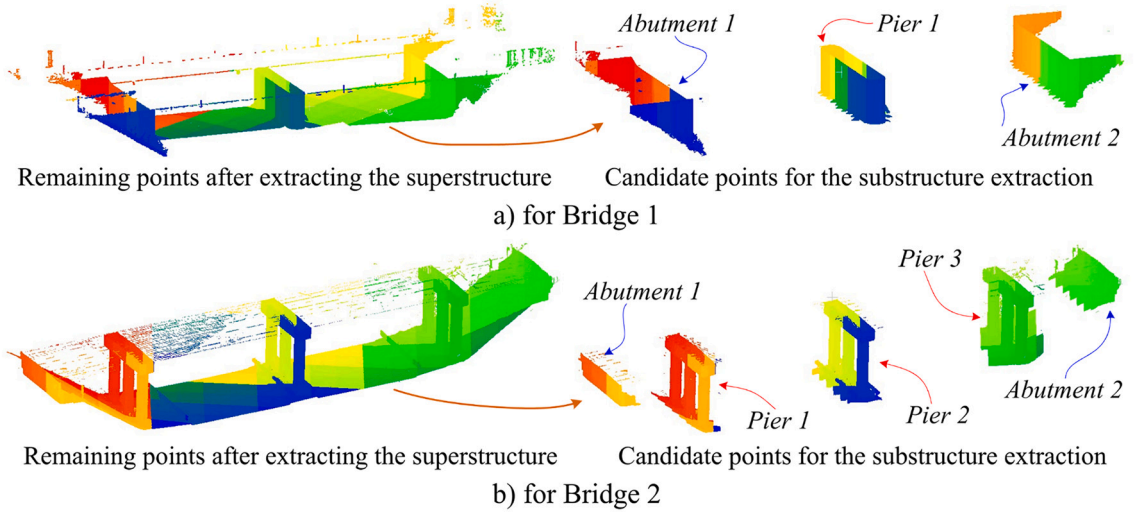


Fig. 23. Extracting candidate points of the substructure of Bridge 1 and 2.

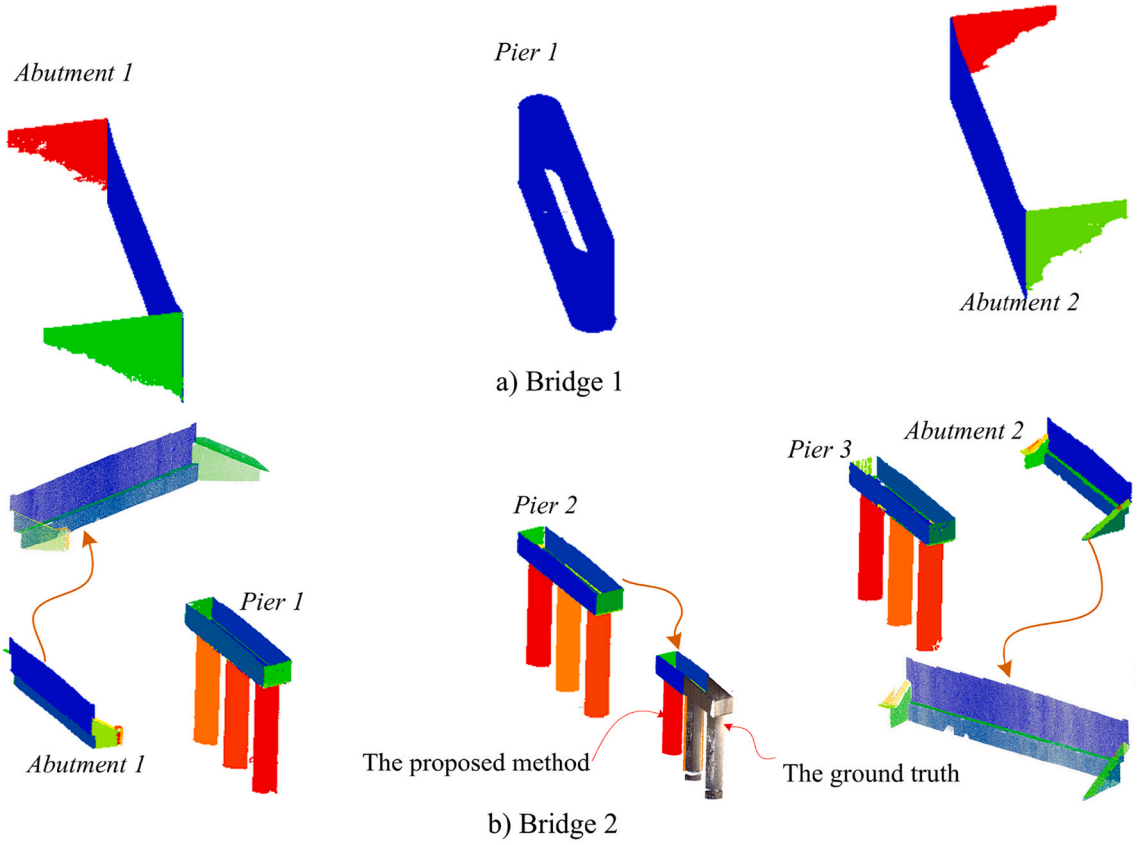


Fig. 24. Resulting surfaces of the abutments and piers for Bridge 1 and 2.

substructure must satisfy Eq. (19). The candidate points of the substructure as the results shown in Fig. 23.

The abutment and the pier have different shape. Surfaces of the abutment are planes, while the pier may involve planes and both planes and curves (as parts of a solid model of a pier appears a semi-cylinder or cylinder). To extract these surfaces by VRG, the following parameters were used $ve_0 = 0.15$ m, $\alpha_{v0} = 5$ degrees, $d_{v0} = 10$ mm, and $vr_{v0} = 10$ mm for the abutment, while $ve_0 = 0.05$ m, and $\alpha_{v0} = 6.75$ degrees, $d_{v0} = 10$ mm, and $vr_{v0} = 10$ mm for the piers, which is based on an assumption that the pier has a curve surface with a radius of 0.6 m. However, if the pier consists of planar surfaces, the set of parameters for the abutment can be used. Moreover, $\alpha_0 = 15$ degrees and a length threshold $L_0 = S_{cs}$. $D_0 = 0.4$ m were also used for the CSC method. Results of the extracted surfaces of the abutments and piers are shown in Fig. 24. However, because there is a gap due to missing data points at bottom parts of the pier columns in Bridge 2, points of these regions were not recognized

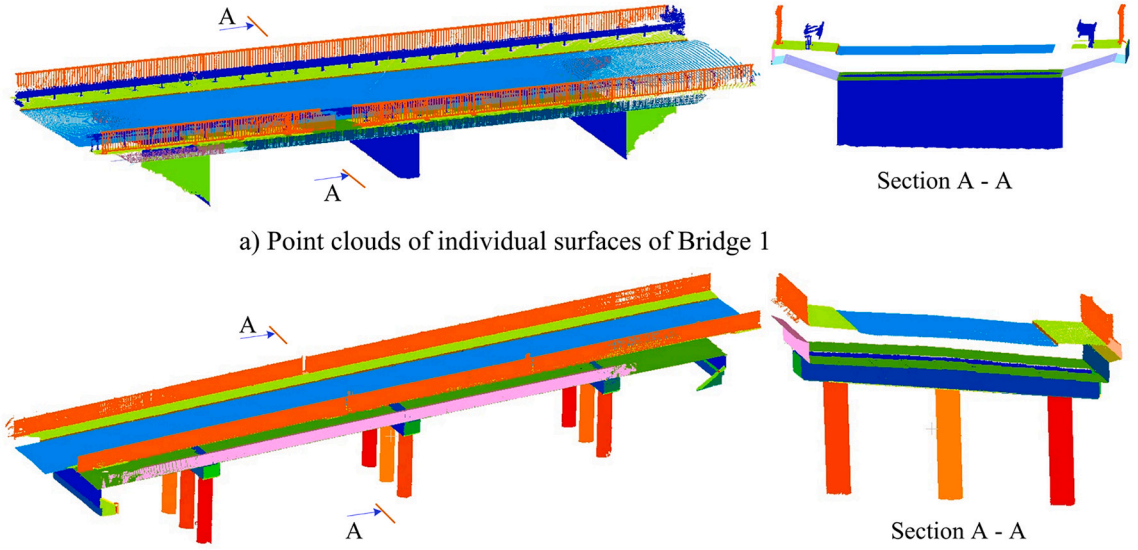


Fig. 25. Resulting decomposition of the bridge components

(Fig. 24b). Finally, Fig. 25 shows results of the extracted surfaces of all bridge components.

5.4. Evaluation

Goal of this study is to extract point clouds of surfaces of bridge components. Resulting data points can be used in various applications, for example, 3D bridge model reconstruction and surface-based damage detection. A complete evaluation must be done based on those applications because this can give full insight performance of the proposed method. However, as these applications are out of a scope of this study, resulting extraction of all surfaces of bridge components (Fig. 25) is evaluated by assessing location deviations and shape similarity with respect to ground truth [70]. In this evaluation, the ground truth was manually extracted from the bridge point clouds.

5.4.1. Location deviation

For local deviations, object and point levels are used to evaluate the component extraction results. At the object level, a bridge component is considered to be extracted successfully if the overlap between the extracted component and a corresponding one in the ground truth was larger than a predefined threshold. However, as the bridge components appear either as *2D* objects (e.g., surfaces of a superstructure) or *3D* objects (e.g., piers and abutments), *2D* and *3D mBB.s* derived from the point clouds of the bridge components were used to determine the overlap. A threshold-free [71] is used to investigate a stable performance of the proposed method (Table 3). Although all components are successfully extracted, the success rate was reduced when requiring an overlap at least than 0.85. Extraction for road curbs and intermediate surfaces are sometimes not successful because they are either small objects in terms of size or data coverage. Particularly, overlap ratios of 0.769 and 0.778 occur for intermediate surfaces A and B of Bridge 2, which indicated that the objects were failed to extract for the overlap threshold above 0.85. Additionally, road curbs A for both bridges have an overlap ratio below 0.90. Interestingly, 3D objects like railings, piers, and abutments are successfully extracted respect with an overlap threshold of 0.90.

For point-based evaluation, the following quantitative indicators including a true positive (*TP)*, false positive (*FP)* and false negative (*FN)* were used to identify a difference between data points of bridge components extracted from the proposed method and those of the ground truth. Evaluation quantities is interpreted through completeness (*Comp.*), correctness (*Corr.*), and *F1-score* Definition of these quantities and the methodology to compute them can refer to Laefer and Truong-Hong [70] and Vu et al. [23]. Based on labels of point clouds, a pair of surfaces and/or structural components, which are the same label, were retrieved, and *TP*, *FP* and *FN* were subsequently determined (Table 4). The evaluation showed the proposed method extracted the point clouds of the bridge components with high accuracy. Indeed, *Comp.*, *Corr.,* and *F1-score* were respectively no smaller than 91.3%, 93.7% and 0.954 for Bridge 1 and 88.6%, 87.2% and 0.932 for Bridge 2, except for the road curbs where the lowest *F1-score* was 0.878 and 0.662 for Bridge 1 and 2. The road curbs had got lower success scores than other components because low quantity and quality of the point cloud were acquired at the road curb. However, an opposite trend was found that the roadway and bottom surfaces of the superstructure were always got the highest *F1-score*. For example, for Bridge 1, *F1-score* was respectively 0.998 and 0.986, while it was 0.995 and 0.987 for Bridge 2. Moreover, there was a slightly different *F1-score* between extracting surfaces of the superstructure and ones of substructure, in which the minimum *F1-score* was respectively 0.954 and 0.993 for Bridge 1 and 0.932 and 0.978 for Bridge 2. Interestingly, there was highly consistency of performance of the proposed method when applied to Bridge 1 and 2, where the difference in *F1-score* for each component is no more than 0.053, except for that of the road curb where the difference is 0.078 (Table 4).

5.4.2. Shape similarity

In addition, a shape similarity was used to measure difference between surfaces of bridge components derived from the proposed method and the ground truth. This was interpreted through difference of area, area overlap rate and deviation of surface normal between a pair of surfaces derived from the proposed method and ground truth [70]. This evaluation only applied for components represented by planes. As such, surfaces of the roadway, road curbs, sidewalks, intermediate and bottom surfaces, and abutments were selected to evaluate, while surfaces of other components (e.g., railings and piers) excluded because they consist of different surface shapes requiring additional modelling techniques to estimate geometric parameters of the surfaces, which is out of the scope of this study.

From point cloud label, the data points describing the same type of bridge component were retrieved from both the proposed method (PM) and the ground truth (GT). From a pair of surfaces (S_{PM} and S_{GT}) of PM and GT, a fitting plane (s_{GT}) of the surface S_{GT} was estimated based on Eqs. (3) and (4), and the point clouds $p_{GT} \in S_{GT}$ and $p_{PM} \in S_{PM}$ were projected onto the fitting plane s_{GT}. Next, the alpha shape algorithm [72] was employed to extract boundary points ($p_{PM.ext} \subseteq p_{PM}$ and $p_{GT.ext} \subseteq p_{GT}$) from the projected points, in which the radius threshold of 0.1 m was empirically selected. For example, Fig. 26 illustrates a procedure to

Table 3
Object-based evaluation with various overlap thresholds[a].

Component	Bridge 1: Overlap threshold				Bridge 2: Overlap threshold			
	≥ 0.75	≥ 0.80	≥ 0.85	≥ 0.90	≥ 0.75	≥ 0.80	≥ 0.85	≥ 0.90
Roadway	1	1	1	1	1	1	1	1
Road curb A	1	1	1	0	1	1	1	0
Road curb B	N/A	N/A	N/A	N/A	1	1	1	1
Sidewalk A	1	1	1	1	1	1	1	1
Sidewalk B	1	1	1	1	1	1	1	1
Intermediate surface A	1	1	1	1	1	0	0	0
Intermediate surface B	1	1	1	1	1	0	0	0
Bottom surface	1	1	1	1	1	1	1	1
Vehicle railing A	1	1	1	1	1	1	1	1
Vehicle railing B	1	1	1	1	1	1	1	1
Pedestrian railing A	1	1	1	1	1	1	1	1
Pedestrian railing B	1	1	1	1	1	1	1	1
Pier 1	1	1	1	1	1	1	1	1
Pier 2	N/A	N/A	N/A	N/A	1	1	1	1
Pier 3	N/A	N/A	N/A	N/A	1	1	1	1
Abutment 1	1	1	1	1	1	1	1	1
Abutment 2	1	1	1	1	1	1	1	1

[a] Note: A or B indicates components on the Side A or B respect with the longitudinal central line of each bridge in Figs. 21 and 22.

Table 4
Point-based evaluation[a].

Bridge component		TP	FP	FN	Comp.	Corr.	F1-score
		Bridge 1					
Superstructure	Roadway	2,140,455	31	6747	99.7%	100.0%	0.998
	Road curb A	30,615	2630	5853	84.0%	92.1%	0.878
	Road curb B	N/A	N/A	N/A	N/A	N/A	N/A
	Sidewalk A	180,024	12,195	3345	98.2%	93.7%	0.959
	Sidewalk B	580,888	16,742	1666	99.7%	97.2%	0.984
	Intermediate surface A	678,604	301	24,060	96.6%	100.0%	0.982
	Intermediate surface B	547,973	217	24,900	95.7%	100.0%	0.978
	Bottom surface	11,394,247	146,543	187,123	98.4%	98.7%	0.986
	Vehicle railing A	239,287	486	8775	96.5%	99.8%	0.981
	Vehicle railing B	328,350	200	31,227	91.3%	99.9%	0.954
	Pedestrian railing A	754,279	3307	4675	99.4%	99.6%	0.995
	Pedestrian railing B	774,733	1024	22,213	97.2%	99.9%	0.985
Substructure	Pier 1	2,380,675	7618	15,098	99.4%	99.7%	0.995
	Abutment 1	1666,717	3485	20,918	98.8%	99.8%	0.993
	Abutment 2	1,395,300	5540	13,594	99.0%	99.6%	0.993
		Bridge 2					
Superstructure	Roadway	1,603,530	4757	11,292	99.3%	99.7%	0.995
	Road curb A	25,979	1384	11,582	69.2%	94.9%	0.800
	Road curb B	12,686	772	11,553	52.3%	94.3%	0.673
	Sidewalk A	285,834	6974	24,626	92.1%	97.6%	0.948
	Sidewalk B	362,412	7700	45,324	88.9%	97.9%	0.932
	Intermediate surface A	348,474	7379	44,877	88.6%	100.0%	0.939
	Intermediate surface B	195,513	5638	20,503	90.5%	100.0%	0.950
	Bottom surface	3,046,501	6135	74,069	97.6%	99.8%	0.987
	Pedestrian railing A	450,142	5413	23,930	95.0%	98.8%	0.959
	Pedestrian railing B	456,929	4992	34,323	93.0%	98.9%	0.964
Substructure	Pier 1	412,864	7789	10,566	97.5%	98.1%	0.978
	Pier 2	308,334	5869	19,873	93.9%	98.1%	0.960
	Pier 3	399,372	16,002	11,917	97.1%	96.1%	0.966
	Abutment 1	192,000	628	21,838	89.8%	99.7%	0.945
	Abutment 2	197,848	1623	17,119	92.0%	99.2%	0.955

[a] Note: A or B indicates components on the Side A or B respect with the longitudinal central line of each bridge in Figs. 21 and 22.

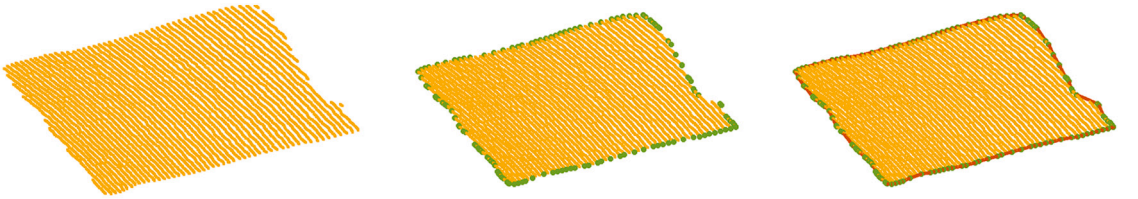


Fig. 26. Creating a polygon describing a boundary of the surface.

Table 5
Similarity evaluation[a].

Component	Bridge 1					Bridge 2				
	Surface area (m^2)			Overlap ratio	Dev. angle (degrees)	Surface area (m^2)			Overlap ratio	Dev. angle (degrees)
	Ground truth	Proposed method	Overlap			Ground truth	Proposed method	Overlap		
Roadway	225.1	224.2	224.2	0.996	0.007	398.1	398.3	397.1	0.997	0.003
Road curb A	2.1	1.9	1.7	0.847	0.237	6.0	5.0	4.8	0.802	0.863
Road curb B	N/A	N/A	N/A	N/A	N/A	5.2	3.5	3.3	0.639	2.193
Sidewalk A	60.5	60.7	58.9	0.973	0.039	96.9	94.6	93.2	0.962	0.169
Sidewalk B	29.3	31.9	29.1	0.993	0.015	97.5	93.9	92.0	0.944	0.051
Int. surface A	14.0	13.5	13.5	0.969	0.044	52.3	45.2	43.3	0.827	0.201
Int. surface B	12.4	11.6	11.6	0.938	0.094	54.0	48.5	46.9	0.870	0.373
Bot. surface	242.1	246.1	239.9	0.991	0.190	514.7	514.5	513.2	0.997	0.002
Abutment 1	27.9	27.3	27.2	0.973	0.018	25.1	25.8	24.0	0.957	0.593
Abutment 2	24.9	24.6	24.4	0.978	0.015	25.4	24.0	23.4	0.921	0.517

[a] Note: A or B indicates components on the Side A or B respect with the longitudinal central line of a bridge in Figs. 21 and 22.

create a boundary of the surfaces by using the alpha shape algorithm. The polygon bounded the surfaces S_{PM} and S_{GT} were respectively created from the boundary points $p_{PM.ext}$ and $p_{GT.ext}$. Finally, areas of the surfaces S_{PM} and S_{GT} were computed from these polygons, and the overlap area between the surfaces S_{PM} and S_{GT} was determined as the area of intersection between two polygons. Moreover, the discrepancy of orientations of the surfaces S_{PM} and S_{GT} was determined as the angle between normal vectors of the fitting planes of S_{PM} and S_{GT} (Table 5).

A shape similarity evaluation showed there were low discrepancy between surfaces derived from the proposed method and ground truth. The minimum overlap ratio and the maximum angle deviation were respectively 0.938 and 0.190 degree for Bridge 1 and 0.827 and 0.593 degree for Bridge 2, when surfaces of the road curbs were excluded. The results showed that the proposed method extracted the road curbs and intermediate surfaces with a lower overlap ratio and larger angle deviation compared to surfaces of other components. For example, the minimum overlap ratio and maximum angle deviation were 0.973 (for Abutment 1) and 0.044 degrees (for intermediate surface A) for Bridge 1, and 0.921 (for Abutment 2) and 0.593 degrees (for Abutment 1) for Bridge 2. Interestingly, the maximum overlap ratio for Bridge 1 and 2 are 0.996 and 0.997.

5.5. Computational time

Results showed that the proposed method extracts surfaces of bridge components with high accuracy, but also is able process large data sets efficiently with a processing time about 481.85 s for 28.505 million points of Bridge 1 and 679.34 s for 12.670 million points of Bridge 2. This efficiency was achieved because the proposed method only processed sub-sets containing candidate points of a structural component of interest. The superstructure extraction takes a large portion of the total processing time, which was respectively 407.87 s and 485.16 s for Bridge 1 and 2. That running time was based on processing 5308 cells with 12,338 patches for Bridge 1 and 2918 cells with 11,477 patches in Bridge 2. Moreover, the running time of Bridge 2 is larger than that of Bridge 1 because a larger number of patches or voxels on region boundaries needed to be analysed in *Step 3* and *4* for CRG and VRG. Moreover, as Bridge 2 has three piers while it was one pier in Bridge 1, executing time for the substructure of Bridge 2 was also larger than that of Bridge 1 (194.18 s for Bridge 2 vs. 73.98 s for Bridge 1). Thus, it is arguably remarked that the processing time mainly depends not only the number of cells (and patches) and voxels on the region boundaries but also the number of structural components. The proposed method is over-performance comparing to the work of Lu et al. [19] where the bridge components for 10 bridges were extracted with an average running time of 481.2 ± 181.2 s with less than one million points. Notably, the work of Lu et al. [19] was implemented in C# and only extracted data points of the bridge components including box-beams and piers instead of surfaces of the components as this study. The performance reported herein based on an implementation of the proposed method in MATLAB 2019b [73] and was processed on Dell Precision Workstation with a main system configuration as follows: Intel(R) Xeon(R) W-2123 CPU @ 3.6GHz with 32GB RAM.

5.6. Discussion

The proposed method extracts point clouds corresponding to bridge components' surfaces through local surfaces with 2D cells in the xy plane and 3D voxels. For each structural element, the process consists of two steps: coarse extraction and fine filtering, in which CRG and VRG are developed to segment point clouds of the surfaces from candidate points derived from the coarse extraction. Although experimental tests showed that the proposed method extracts surfaces of the bridge components with high accuracy, a success of the proposed method depends on suitable input parameters (Table 2), and quantity and quality of the point clouds.

Notably, as cells and voxel sizes were determined, based on the minimum dimensions of surfaces of bridge components, as shown in Table 1, the proposed method may fail to extract a surface if a small cell or voxel was generated in combination with a data set with a large sampling step. This could lead to a cell or voxel that classified as "*empty*", which would consecutively eliminate this cell or voxel from further processing, as the points within the cell or voxel would not satisfy the thresholds c_{min_ptc} and v_{min_ptc}, respectively.

Both CRG and VRG are often sensitive to input parameters consisting of angle, distance, and residual thresholds, which depend on both surface geometries and data quality. Although geometries of popular bridges are used to determine a portion of each threshold relating to the bridge geometry, the portion depending on the data quality is still empirically selected. Moreover, surfaces of the bridge components are mostly smooth planes, and the data quality is assumed as homogenous. The proposed method with fixed values of these thresholds (Table 2) can extract all surfaces with high accuracy. However, for an entire bridge, the components' surfaces can be made from different materials, have different roughness and/or are subjected to damage. The fixed thresholds may cause under- or over-segmentation for some cases. For example, parts of the roadway of Bridge 2 are missing because of surface damage (Fig. 27). As such, adaptive thresholds could be obtained based on a set of candidate points at a component level because this subset contains large portion of the data points of the component surface, which can provide rough features of the surface. For example, a possible solution is that a subset of the patches or voxels associated with their neighbourhood can be used to estimate the distance and residual thresholds through a statistical framework. This work will be parts of future work of this study.

One of the advantages of the proposed method is that a subset occupied candidate points of a component of interest is used as input data for CRG and VRG, which implies that fitting planes for patches and voxels would mostly have similar characteristics. With this strategy, a complexity is reduced significantly, which would support to select thresholds for CRG and VRG easier than the case of using entire data points. For example, Fig. 28 showed resulting surfaces based on the proposed method and RANdom Sample Consensus (RANSAC) proposed by Schnabel et al. [29] implemented in CloudCompare V2.7.2 [69]. To extract surfaces using RANSAC, a normal vector of each data point is computed, which is the normal of the fitting plane computed from its neighbouring points determined by using a range search with a searching radius of 0.1 m (in this study). Additionally, input parameters for RANSAC consisting of (i) the maximum distance from the points to the model ($\varepsilon_{max} = 10$ mm), (ii) the maximum deviation between points' normal vector and one of the model's ($\alpha_{RANSAC} = 5$ degrees), and (iii) the minimum number of points ($model_{min\_ptc} = 1000$ points) of the model are used to get the best results. The resulting segmentation showed that RANSAC extracted surfaces of pier components similar to those from the proposed method. Moreover, RANSAC-based surfaces miss points around edges (Fig. 28b), which do not occur in the proposed method (Fig. 28c). RANSAC can cause over-segmentation, for example, the bottom surface of the pier cap appears as 4 segments. RANSAC results require an additional post-processing step to eliminate unreal surfaces.

Although the complexity of data is significantly reduced in this proposed method, a high similarity between adjoining surfaces also prevents to separate them. For example, in Bridge 2, the superstructure consists of multiple slab-beams in a cross-section of the bridge, but bottom surfaces of these slab-beams are recognized as the same region. That is because the patches representing these surfaces have very small deviations of features including a normal vector, distance, and residual (Fig. 29). As such, to obtain the bottom surface of each slab-beam, cell size and thresholds for CRG must be adjusted. In addition, results of pier extraction can also be used to separate the slab-beams of each bridge span.

The point cloud of the bridge is decomposed into 2D cells and 3D voxels and only full cells and voxels possessing the number of points

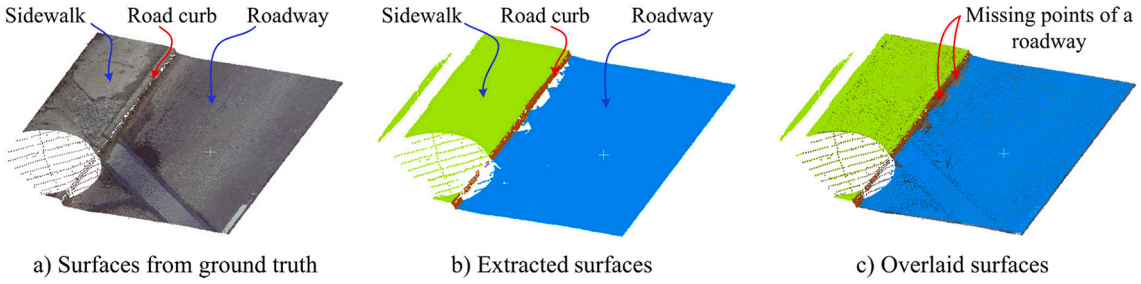


Fig. 27. Missing points due to surface damage.

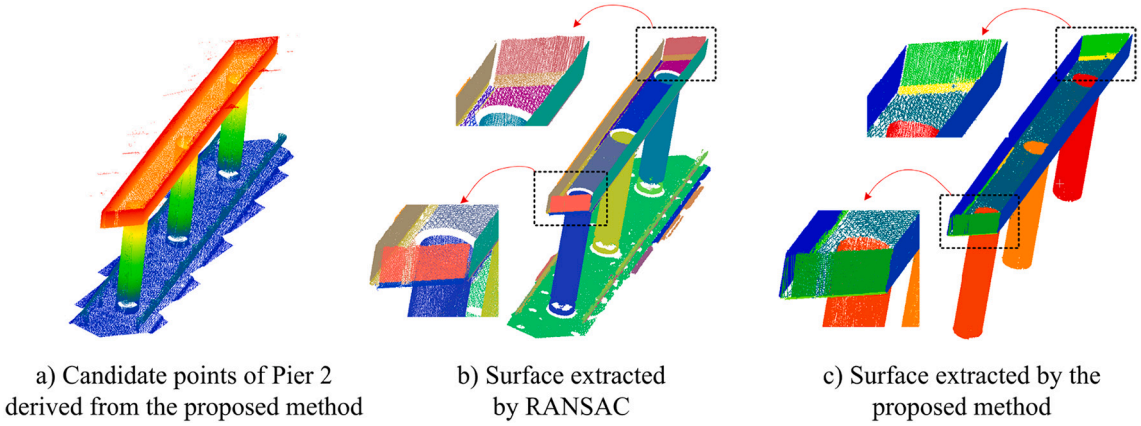


Fig. 28. Comparison between surface extraction results from the proposed method and RANSAC.

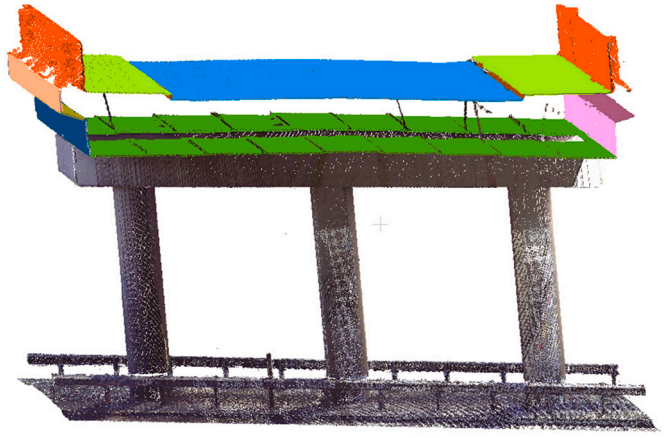


Fig. 29. Overlaid extracted data points of a superstructure to the point clouds of Bridge 2.

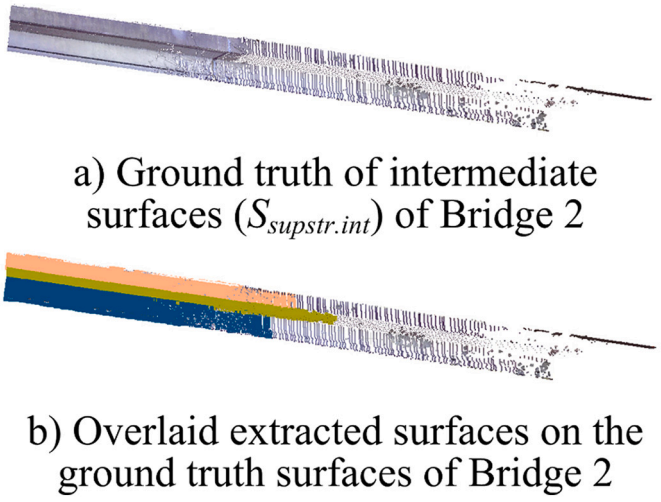


Fig. 30. Missing points due to a low point density.

larger than predefined thresholds (c_{min_ptc} and v_{min_ptc}) are considered in further steps. However, due to a locally sparse, some areas of the surface were represented by empty cells or voxels, and CRG and VRG failed to recognize surfaces of bridge components. For example, intermediate surfaces of Bridge 2 have areas with low point density, and VRG segmentation fails to extract surfaces for these areas (Fig. 30). Thus, selecting the cell size and voxel size should be considered not only dimensions of extracted surfaces or structures' dimensions but also the point density of the input data set.

As mentioned above, input parameters, particularly angle, distance, and residual thresholds in CRG and VRG are dominated by bridge geometry and data quality. When input data is of insufficient quality, angle and/or distance between two incident patches for CRG and voxels for VRG may be dominated by data quality. To extract surfaces of bridge components successfully, either the effect of data quality must be minimized, or the thresholds must be relaxed, or both. For the first option, RANSAC can be integrated into the proposed method to estimate parameters of a plane fitting to the points in the patch or voxel, which implies that the percentage of outlying points can be up to nearly 50%. Moreover, observing general bridge geometry shows that angles between incident surfaces in a bridge are theoretically not smaller than 45^0 (most of them are not smaller than 90^0) because of optimal load transfer and aerodynamic shapes, and the distance between two adjacent parallel surfaces is no smaller than 0.157 m (Table 1). Once a patch (or voxel) possesses mixed points of both adjoined surfaces, the angle and distance between this patch (or voxel) and the patch (or voxel)

containing purse points of one surface can theoretically be about 22.5^0 (a half of the minimum angle, 45^0), and 0.075 m (a half of the minimum distance, 0.157 m), respectively. As such, in the second option, to prevent over- or under segmentation due to low-quality data, these values for angle and distance can be used as upper bounds in selecting these thresholds. Finally, the use of a residual threshold equal to two times the registration error can often determine sufficient seeding patches or voxels for CRG and VRG, respectively.

CRG not only applies to extract planar surfaces of a roadway, side walls and bottom surfaces of a superstructure, but can also be extended to the extraction of planes of arbitrary orientation. In this case, the data set is decomposed into 2D cells in either the xy, yz or xz plane. Next, orientation KDE is implemented to roughly extract data points of patches within the 2D cell, in which the data points rotating around an axis (e.g., ox, oy, or oz) are used to generate a PDS of KDE to find local maximum peaks describing a distribution of the points of the patch. As such, the proposed method should be easily extendable for other type of bridges, but also for other structures.

6. Conclusions

Climate change and increasing freight have cause negative impact on bridges. That requires that bridges must be managed efficiently, and inspected and monitored periodically. However, as digitized bridge models are mostly not available, those tasks are not done efficiently. This paper presents a new method to automatically extract component surfaces of box and slab-beam bridges from a terrestrial laser scanning point cloud, which is a key step in generating a 3D geometric model and surface damage identification. The proposed method deploys both spatial point clouds and basic contextual knowledge of bridge components to extract bridge components in order from a superstructure to substructure. The proposed method starts to extract point clouds of surfaces from sub data sets, and subsequently assigns the surfaces as parts of superstructure and substructure. Surfaces of each bridge component are obtained through coarse extraction to get candidate points and fine filtering to obtain final data points of surfaces using CRV, VRG and CSC methods. In the proposed method, contextual knowledge relating to location, orientation and minimum dimensions of the bridge components are integrated to support coarse extraction and unreal structural surface elimination.

The proposed method was tested on data sets of two box- and slab-beam bridges, and results showed that all bridge components were successfully extracted at an object level using an overlap threshold of 0.85. Moreover, a point-based evaluation showed that the proposed method extracted the bridge components with the lowest *F1-score* of 0.954 and 0.932 for Bridge 1 and 2, except for the road curb. Interestingly, a shape similarity evaluation proved that extracted surfaces highly match to those of the ground truth with discrepancies in term of a minimum overlap ratio and a maximum angle deviation respectively 0.938 and 0.190 degree for Bridge 1 and 0.827 and 0.593 degree for Bridge 2. Finally, as only subsets need to be processed to extract the component's surfaces, the proposed method is able to handle large bridge data sets. This is shown by executing time about 481.85 s for 28.505 million points of Bridge 1 and 679.34 s for 12.670 million points of Bridge 2.

Data points of surfaces of bridge elements extracted from the proposed method provide fundamental elements for further processing to obtain results for bridge engineering applications. One of which is to generate a 3D model of the bridge, which can be potentially used for bridge management, bridge assessment and inspection, and bridge information modelling and digital twins. Another application is to identify damage-based surface geometry, for example deformation, scaling or spalling. Finally, although the proposed method aims to apply for slab-beam and box-girder bridges, the method can extent to other common types of the bridges. Those works would be parts of future works.

Declaration of Competing Interest

None.

Acknowledgement

This work was funded by the European Commission through H2020 MSCA-IF, "*BridgeScan: Laser Scanning for Automatic Bridge Assessment*", Grant 799149. The first author is grateful for this support.

References

[1] A. Žnidarič, V. Pakrashi, E. O'Brien, A. O'Connor, A review of road structure data in six European countries, Proceed. Inst. Civil Eng. Urban Des. Plann. 164 (4) (2011) 225–232, https://doi.org/10.1680/udap.900054.
[2] ASCE, Report Card for American's Infrastructure: Bridges, Available at: https://infrastructurereportcard.org/cat-item/bridges/, 2021 (Accessed by: Jan. 15, 2021).
[3] K.D. Flaig, A.R.J. Lark, The development of UK bridge management systems, Proceed. Inst. Civil Eng. Transp. 141 (2) (2000) 99–106, https://doi.org/10.1680/tran.2000.141.2.99.
[4] Y. Fujino, D.M. Siringoringo, Structural health monitoring of bridges in Japan: an overview of the current trend, in: Proceedings of the Fourth International Conference on FRP Composites in Civil Engineering (CICE), Zurich, Switzerland, 2008, p. 7. Jul. 20–25.
[5] L. Truong-Hong, D.F. Laefer, Laser scanning for bridge inspection, in: B. Riveiro, R. Lindenbergh (Eds.), Laser Scanning: An Emerging Technology in Structural Engineering 14, CRC Press, London, UK, 2019, p. 26 (135101885X).
[6] D. Moon, S. Chung, S. Kwon, J. Seo, J. Shin, Comparison and utilization of point cloud generated from photogrammetry and laser scanning: 3D world model for smart heavy equipment planning, Autom. Constr. 98 (2019) 322–331, https://doi.org/10.1016/j.autcon.2018.07.020.
[7] S. Chen, L. Truong-Hong, E. O'Keeffe, D.F. Laefer, E. Mangina, Outlier detection of point clouds generating from low-cost UAVs for bridge inspection, in: D. M. Frangopol, R. Caspeele, L. Taerwe (Eds.), 6th International Symposium on Life-Cycle Civil Engineering, IALCCE 2018, CRC Press/Balkema, Ghent, Belgium, 2018, p. 7. Oct. 28-31.
[8] B. Riveiro, M.J. DeJong, B. Conde, Automated processing of large point clouds for structural health monitoring of masonry arch bridges, Autom. Constr. 72 (2016) 258–268, https://doi.org/10.1016/j.autcon.2016.02.009.
[9] ClearEdge3D, EdgeWise, Available at, https://www.clearedge3d.com/edgewise/, 2021 (Accessed by: Jul. 10, 2021).
[10] R. Lu, I.K. Brilakis, Recursive segmentation for As-Is bridge information modelling, in: F. Bosche, I. Brilakis, R. Sacks (Eds.), Joint Conference on Computing in Construction (JC3) 1, Heriot-Watt University, Heraklion, Crete, Greece, 2017, pp. 209–217. Jul. 4–7.
[11] L. Barazzetti, Parametric As-built model generation of complex shapes from point clouds, Adv. Eng. Inform. 30 (3) (2016) 298–311, https://doi.org/10.1016/j.aei.2016.03.005.
[12] B. Riveiro, H. González-Jorge, M. Varela, D.V. Jauregui, Validation of terrestrial laser scanning and photogrammetry techniques for the measurement of vertical underclearance and beam geometry in structural inspection of bridges, Measurement 46 (1) (2013) 784–794, https://doi.org/10.1016/j.measurement.2012.09.018.
[13] L. Truong-Hong, R. Lindenbergh, Measuring Deformation of Bridge Structures using Laser Scanning Data. 4th Joint International Symposium on Deformation Monitoring (JISDM), Athens, Greece,, FIG, 15-17 May, 2019, p. 7, in: https://www.fig.net/resources/proceedings/2019/04_JISDM2019/43.pdf.
[14] L. Truong-Hong, S. Chen, V.L. Cao, D.F. Laefer, Automatic Bridge Deck Damage Using Low Cost UAV-based Images, Quality Specifications for Roadway Bridges, Standardization at A European Level, Barcelonan, Spain, Sept. 26-27, TU1406, 2018, p. 6, https://eurostruct.org/wp-content/uploads/2020/12/TU1406_BARCELONA_Ebook.pdf.
[15] G. Teza, A. Galgaro, F. Moro, Contactless recognition of concrete surface damage from laser scanning and curvature computation, NDT E Int. 42 (4) (2009) 240–249, https://doi.org/10.1016/j.ndteint.2008.10.009.
[16] L. Truong-Hong, D.F. Laefer, Documentation of Bridges by Terrestrial Laser Scanner, IABSE Conference: Structural Engineering: Providing Solutions to Global Challenges, IABSE, Geneva, Switzerland, 2015, pp. 1384–1391, https://doi.org/10.2749/geneva.2015, 23-25 September.
[17] M.-K. Kim, H. Sohn, C.-C. Chang, Localization and quantification of concrete spalling defects using terrestrial laser scanning, J. Comput. Civ. Eng. 29 (6) (2015) 04014086, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000415.
[18] J. Balado Frias, A. Sánchez-Rodríguez, Point cloud approach for modelling the lost volume of the Fillaboa Bridge cutwater, Surv. Geospatial Eng. J. 2 (01) (2021) 13–20, https://doi.org/10.38094/sgej2018.
[19] R. Lu, I. Brilakis, C.R. Middleton, Detection of structural components in point clouds of existing RC bridges, Comput. Aided Civil Infrastruct. Eng. 34 (3) (2019) 191–212, https://doi.org/10.1111/mice.12407.
[20] L. Truong-Hong, R. Lindenbergh, Extracting bridge components from a laser scanning point cloud, in: E. Toledo Santos, S. Scheer (Eds.), Proceedings of the 18th International Conference on Computing in Civil and Building Engineering, Springer

International Publishing, Cham, 2021, pp. 721–739, https://doi.org/10.1007/978-3-030-51295-8_50.
[21] Y. Yan, J.F. Hajjar, Automated extraction of structural elements in steel girder bridges from laser point clouds, Autom. Constr. 125 (2021), 103582, https://doi.org/10.1016/j.autcon.2021.103582.
[22] T. Rabbani, F.V.D. Heuvel, G. Vosselmann, Segmentation of point clouds using smoothness constraint, international archives of photogrammetry, remote sensing and spatial, Inf. Sci. 36 (5) (2006) 248–253, in: http://www.isprs.org/proceedings/XXXVI/part5/paper/RABB_639.pdf.
[23] A.-V. Vo, L. Truong-Hong, D.F. Laefer, M. Bertolotto, Octree-based region growing for point cloud segmentation, ISPRS J. Photogramm. Remote Sens. 104 (2015) 88–100, https://doi.org/10.1016/j.isprsjprs.2015.01.011.
[24] P.V.C. Hough, Analysis of Bubble Chamber Pictures, 2nd International Conference on High-Energy Accelerators and Instrumentation (HEACC), CERN, Geneva, Switzerland, 1959, pp. 554–558. Sept. 14-19, https://inspirehep.net/files/53d80b0393096ba4afe34f5b65152090.
[25] R. Schnabel, R. Wahl, R. Klein, Efficient RANSAC for point-cloud shape detection, Comput. Graph. Forum 26 (2007) 214–226, https://doi.org/10.1111/j.1467-8659.2007.01016.x.
[26] A. Dimitrov, M. Golparvar-Fard, Segmentation of building point cloud models including detailed architectural/structural features and MEP systems, Autom. Constr. 51 (2015) 32–45, https://doi.org/10.1016/j.autcon.2014.12.015.
[27] S. Ochmann, R. Vock, R. Wessel, R. Klein, Automatic reconstruction of parametric building models from indoor point clouds, Comput. Graph. 54 (2016) 94–103, https://doi.org/10.1016/j.cag.2015.07.008.
[28] C. Thomson, J. Boehm, Automatic geometry generation from point clouds for BIM, Remote Sens. 7 (9) (2015) 11753–11775, https://doi.org/10.3390/rs70911753.
[29] R. Schnabel, R. Wahl, R. Klein, Efficient RANSAC for point-cloud shape detection, Comput. Graph. Forum 26 (2) (2007) 214–226, https://doi.org/10.1111/j.1467-8659.2007.01016.x.
[30] A. Costin, A. Adibfar, H. Hu, S.S. Chen, Building Information Modeling (BIM) for transportation infrastructure – literature review, applications, challenges, and recommendations, Autom. Constr. 94 (2018) 257–281, https://doi.org/10.1016/j.autcon.2018.07.001.
[31] H. Son, F. Bosché, C. Kim, As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey, Adv. Eng. Inform. 29 (2) (2015) 172–183, https://doi.org/10.1016/j.aei.2015.01.009.
[32] L. Truong-Hong, D.F. Laefer, Application of terrestrial laser scanner in bridge inspection: review and an opportunity, in: 37th IABSE Symposium: Engineering for Progress, Nature and People, Madrid, Spain, Sept. 3–5, 2014, International Association for Bridge and Structural Engineering (IABSE), 2014, https://doi.org/10.2749/222137814814070190.
[33] C.J. Stull, C.J. Earls, A rapid assessment methodology for bridges damaged by truck strikes, Steel Compos. Struct. 9 (3) (2009) 223–238, https://doi.org/10.12989/scs.2009.9.3.223.
[34] G. Nora, L. Truong-Hong, D.F. Laefer, Laser scan-based structural assessment of wrought iron bridges: Guinness Bridge, Ireland, Proceed. Inst. Civil Eng. 171 (2) (2018) 76–89, https://doi.org/10.1680/jenhh.17.00018.
[35] Leica Geosystems. Leica CloudWorx for AutoCAD, Leica, 2018. https://leica-geosystems.com/products/laser-scanners/software/leica-cloudworx/leica-cloudworx-autocad.
[36] Y. Yan, B. Guldur, F. Hajjar Jerome, Automated Structural Modelling of Bridges from Laser Scanning, Structures Congress 2017, ASCE, Denver, Colorado, USA, 2017, pp. 457–468, https://doi.org/10.1061/9780784480403.039. Apr. 6–8.
[37] R. Schneiders, A grid-based algorithm for the generation of hexahedral element meshes, Eng. Comput. 12 (3–4) (1996) 168–177, https://doi.org/10.1007/BF01198732.
[38] Y. Pan, Y. Dong, D. Wang, A. Chen, Z. Ye, Three-dimensional reconstruction of structural surface model of heritage bridges using UAV-based photogrammetric point clouds, Remote Sens. 11 (10) (2019) 1204, https://doi.org/10.3390/rs11101204.
[39] S.B. Walsh, D.J. Borello, B. Guldur, J.F. Hajjar, Data processing of point clouds for object detection for structural engineering applications, Comput. Aided Civil Infrastruct. Eng. 28 (7) (2013) 495–508, https://doi.org/10.1111/mice.12016.
[40] G. Zhang, P.A. Vela, P. Karasev, I. Brilakis, A sparsity-inducing optimization-based algorithm for planar patches extraction from noisy point-cloud data, Comput. Aided Civil Infrastruct. Eng. 30 (2) (2015) 85–102, https://doi.org/10.1111/mice.12063.
[41] Y. Narazaki, V. Hoskere, T.A. Hoang, Y. Fujino, A. Sakurai, B.F. Spencer Jr., Vision-based automated bridge component recognition with high-level scene consistency, Comput. Aided Civil Infrastruct. Eng. 35 (5) (2020) 465–482, https://doi.org/10.1111/mice.12505.
[42] F. Hu, J. Zhao, Y. Huang, H. Li, Structure-aware 3D reconstruction for cable-stayed bridges: a learning-based method, Comput. Aided Civil Infrastruct. Eng. 36 (2020) 89–108, https://doi.org/10.1111/mice.12568.
[43] B.J. Perry, Y. Guo, R. Atadero, J.W. van de Lindt, Streamlined bridge inspection system utilizing Unmanned Aerial Vehicles (UAVs) and machine learning, Measurement 164 (2020), 108048, https://doi.org/10.1016/j.measurement.2020.108048.
[44] H. Kim, J. Yoon, S.-H. Sim, Automated bridge component recognition from point clouds using deep learning, Struct. Control. Health Monit. 27 (e2591) (2020) 13, https://doi.org/10.1002/stc.2591.
[45] J.S. Lee, J. Park, Y.-M. Ryu, Semantic segmentation of bridge components based on hierarchical point cloud model, Autom. Constr. 130 (2021), 103847, https://doi.org/10.1016/j.autcon.2021.103847.
[46] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph CNN for learning on point clouds, ACM Trans. Graph. 38 (5) (2019), https://doi.org/10.1145/3326362. Article 146.
[47] N. Saovana, N. Yabuki, T. Fukuda, Automated point cloud classification using an image-based instance segmentation for structure from motion, Autom. Constr. 129 (2021), 103804, https://doi.org/10.1016/j.autcon.2021.103804.
[48] D.N. Farhey, Structural performances of bridge types in the U.S. National Bridge Inventory, Infrastructures 3 (1) (2018) 6, https://doi.org/10.3390/infrastructures3010006.
[49] M.-K. Kim, S. McGovern, M. Belsky, C. Middleton, I. Brilakis, A suitability analysis of precast components for standardized bridge construction in the United Kingdom, Procedia Eng. 164 (2016) 188–195, https://doi.org/10.1016/j.proeng.2016.11.609.
[50] AASHTO, AASHTO LRFD Bridge Design Specifications, 8th edition, American Association of State Highway and Transportation Officials (AASHTO), AASHTO, 2017 (978-1-56051-654-5).
[51] New York State Department of Transportation (Ed.), NYS Bridge Manual, 2019 ed., New York State Department of Transportation, New York, USA, 2019. https://www.dot.ny.gov/divisions/engineering/structures/repository/manuals/brman-usc/NYSDOT_bridge_manual_US_5-2019.pdf.
[52] W. Lin, T. Yoda, Chapter Five - Bridge deck systems, in: W. Lin, T. Yoda (Eds.), Bridge Engineering, Butterworth-Heinemann, 2017 (pp. 85-96, 978-0-12-804432-2).
[53] AASHTO, A Policy on Geometric Design of Higways and Streets, AASHTO, Washington, DC, 2018 (p. 1048, 978-1-56051-676-7).
[54] W.-F. Chen, L. Duan, Bridge Engineering Handbook 1st Edition ed, 1, CRC Press, 2019, p. 2045 (0429277040).
[55] I. Armeni, O. Sener, A.R. Zamir, H. Jiang, I. Brilakis, M. Fischer, S. Savarese, 3D Semantic Parsing of Large-Scale Indoor Spaces, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 27-30, 2016, IEEE, 2016, pp. 1534–1543, https://doi.org/10.1109/CVPR.2016.170.
[56] D.F. Laefer, L. Truong-Hong, Toward automatic generation of 3D steel structures for building information modelling, Autom. Constr. 74 (2017) 66–77, https://doi.org/10.1016/j.autcon.2016.11.011.
[57] V.A. Epanechnikov, Non-parametric estimation of a multivariate probability density, Theory Probab. Appl. 14 (1) (1969) 153–158, https://doi.org/10.1137/1114019.
[58] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Surface Reconstruction from Unorganized Points, ACM SIGGRAPH 1992, Chicago, USA, Jul 26-31, 1992, pp. 71–78, https://doi.org/10.1145/142920.134011.
[59] D. Meagher, Geometric modeling using octree encoding, Comp. Graph. Image Process. 19 (2) (1982) 129–147, https://doi.org/10.1016/0146-664X(82)90104-6.
[60] M. Soilán, L. Truong-Hong, B. Riveiro, D. Laefer, Automatic extraction of road features in urban environments using dense ALS data, Int. J. Appl. Earth Obs. Geoinf. 64 (2018) 226–236, https://doi.org/10.1016/j.jag.2017.09.010.
[61] P. Tang, B. Akinci, D. Huber, Quantification of edge loss of laser scanned data at spatial discontinuities, Autom. Constr. 18 (8) (2009) 1070–1083, https://doi.org/10.1016/j.autcon.2009.07.001.
[62] R.M. Haralick, L.G. Shapiro, Computer and Robot Vision, Addison-wesley Reading 0201108771, 1992, p. 97802.
[63] W.-F. Chen, L. Duan, Bridge Engineering Handbook: Substructure Design, 2nd ed, CRC Press, Boca Raton, 2014, p. 386 (9781439852194).
[64] Leica Geosystems, Leica ScanStation P20, Available at, https://w3.leica-geosystems.com/downloads123/hds/hds/scanstation_p20/brochures-datasheet/leica_scanstation_p20_dat_en.pdf, 2020 (Accessed by: Jan. 15, 2020).
[65] Leica Geosystems, Leica Cyclone, Available at, https://leica-geosystems.com/products/laser-scanners/software/leica-cyclone, 2020 (Accessed by: 15 Dec, 2020).
[66] R. Lu, I. Brilakis, C.R. Middleton, Detection of Structural Components in Point Clouds of Existing RC Bridges, Available at, 2020, https://doi.org/10.5281/zenodo.1240534 (Accessed by: Jan. 15, 2020).
[67] FARO, Faro Focuss 350/350, Available at, https://www.faro.com/en-gb/products/construction-bim-cim/faro-focus/features/ (Accessed by: Dec. 15, 2020).
[68] FARO, PointSense Pro and Basic, Available at, https://knowledge.faro.com/Software/Legacy-Software/Legacy-PointSense_and_CAD_Plugins/PointSense/PointSense_Pro_And_Basic/Overview_PointSense_Pro_and_Basic (Accessed by: Dec. 15, 2020).
[69] CloudCompare, CloudCompare, 2020.
[70] L. Truong-Hong, D.F. Laefer, Quantitative evaluation strategies for urban 3D model generation from remote sensing data, Comput. Graph. 49 (2015) 82–91, https://doi.org/10.1016/j.cag.2015.03.001.
[71] M. Awrangjeb, G. Lu, Automatic Building Footprint Extraction and Regularisation from LIDAR Point Cloud Data, 2014 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Nov. 25-27, IEEE, 2014, pp. 1–8, https://doi.org/10.1109/DICTA.2014.7008096.
[72] H. Edelsbrunner, D. Kirkpatrick, R. Seidel, On the shape of a set of points in the plane, IEEE Trans. Inf. Theory 29 (4) (1983) 551–559, https://doi.org/10.1109/TIT.1983.1056714.
[73] MathWorks, MATLAB Function Reference, MatLab, 2019. https://nl.mathworks.com/help/pdf_doc/matlab/index.html.